

ISSN 2415-7740 (Print)
ISSN 2415-7074 (Online)

Belarusian State University of Informatics and Radioelectronics

Open Semantic Technologies for Intelligent Systems

Research Papers Collection

Founded in 2017

Issue 7

Minsk
2023

UDC 004.822+004.89-027.31

The collection includes peer-reviewed articles approved by the Editorial board. Designed for university professors, researchers, students, graduate students, undergraduates, as well as for specialists of enterprises in the field of intelligent systems design.

Editorial board:

V. V. Golenkov – *Editor-in-chief*,

*I. S. Azarov, V. A. Golovko, A. N. Gordey, N. A. Guliakina,
V. V. Krasnoproshin, A. N. Kurbatsky, B. M. Lobanov, B. N. Panshin,
V. B. Taranchuk, D. V. Shunkevich*

Editorial address:

220005, Minsk, Platonova str., 39, rm. 606 b

Phone: +375 (17) 293-80-92

E-mail: ostisconf@gmail.com

Web-site: <http://proc.ostis.net>

The collection is included in the List of scientific publications of the Republic of Belarus for publishing the results of dissertation research in the technical field of science (computer science, computer technology and management)

The collection is included in the Russian Science Citation Index

ISSN 2415-7740 (Print)
ISSN 2415-7074 (Online)

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Открытые семантические технологии проектирования интеллектуальных систем

Сборник научных трудов

Основан в 2017 году

Выпуск 7

Минск
2023

УДК 004.822+004.89-027.31

Сборник включает прошедшие рецензирование и утвержденные Редакционной коллегией статьи.

Предназначен для преподавателей высших учебных заведений, научных сотрудников, студентов, аспирантов, магистрантов, а также для специалистов предприятий в сфере проектирования интеллектуальных систем.

Редакционная коллегия:

В. В. Голенков – главный редактор,

*И. С. Азаров, В. А. Головки, А. Н. Гордей, Н. А. Гулякина,
В. В. Краснопрошин, А. Н. Курбацкий, Б. М. Лобанов, Б. Н. Панышин,
В. Б. Таранчук, Д. В. Шункевич*

Адрес редакции:

220005, г. Минск, ул. Платонова 39, каб. 606 б

Телефон: +375(17)293-80-92

Электронный адрес: ostisconf@gmail.com

Сайт: <http://proc.ostis.net>

Сборник включён в Перечень научных изданий Республики Беларусь для опубликования результатов диссертационных исследований по технической отрасли науки (информатика, вычислительная техника и управление)

Сборник включён в Российский индекс научного цитирования

© УО «Белорусский государственный университет информатики и радиоэлектроники», 2023

TABLE OF CONTENTS

FOREWORD	11
THE MAIN DIRECTIONS, PROBLEMS AND PROSPECTS OF THE DEVELOPMENT OF THE NEXT-GENERATION INTELLIGENT COMPUTER SYSTEMS AND THE CORRESPONDING TECHNOLOGY Vladimir Golenkov, Natalia Gulyakina	15
FACTOR OF DIGITAL CULTURE IN THE APPLICATION OF ARTIFICIAL INTELLIGENCE IN ECONOMICS AND EDUCATION Boris Panshin	27
CONCEPTUAL DESIGN OF COMPLEX INTEGRATED SYSTEMS Victar Kochyn, Alexander Kurbatsky	33
ASSOCIATIVE SEMANTIC COMPUTERS FOR INTELLIGENT COMPUTER SYSTEMS OF A NEW GENERATION Vladimir Golenkov, Daniil Shunkevich, Natalia Gulyakina, Valerian Ivashenko, Vadim Zahariev	39
OPEN SEMANTIC TECHNOLOGY AS THE FOUNDATION FOR NEW GENERATION INTELLIGENT SYSTEMS Mikhail Tatur, Anton Paramonov	61
DESIGN PRINCIPLES, STRUCTURE, AND DEVELOPMENT PROSPECTS OF THE SOFTWARE PLATFORM OF OSTIS-SYSTEMS Nikita Zotov	67
IMPLEMENTATION OF INFORMATION RETRIEVAL SUBSYSTEM IN THE SOFTWARE PLATFORM OF OSTIS-SYSTEMS Nikita Zotov	77
SEMANTIC SPACE INTEGRATION OF LOGICAL KNOWLEDGE REPRESENTATION AND KNOWLEDGE PROCESSING MODELS Valerian Ivashenko	95
PRINCIPLES OF PROBLEM SOLVING IN DISTRIBUTED TEAMS OF INTELLIGENT COMPUTER SYSTEMS OF A NEW GENERATION Daniil Shunkevich	115
VERSIONING MODEL OF NEURAL NETWORK PROBLEM-SOLVING METHODS IN INTELLIGENT SYSTEMS Mikhail Kovalev	121
REDUCTION OF NEURAL NETWORK MODELS IN INTELLIGENT COMPUTER SYSTEMS OF A NEW GENERATION Aliaksandr Kroshchanka	127
NEURAL NETWORK SOFTWARE TECHNOLOGY TRAINABLE ON THE RANDOM SEARCH PRINCIPLES Victor Krasnoproshin, Vadim Matskevich	133
IMPLEMENTATION OF INFORMATION NEEDS OF OSTIS-SYSTEMS USERS Sergei Samodumkin	141
CONSTRUCTION OF INTELLIGENT SYSTEMS BASED ON KNOWLEDGE DISCOVERY IN DATASETS Victor Krasnoproshin, Vadim Rodchenko, Anna Karkanitsa	147
USER INTERFACE OF THE OSTIS ECOSYSTEM Mikhail Sadoski	153

USER INTERACTION DESIGN IN NEXT-GENERATION IMMERSIVE SYSTEMS Aliaksandr Halavaty, Valeria Bahai, Danila Bernat, Katsiaryna Halavataya	159
DEVELOPING BIRDS SOUND RECOGNITION SYSTEM USING AN ONTOLOGICAL APPROACH Yauheniya Zianouka, Dzianis Bialiauski, Liesia Kajharodava, Aliaksandr Trafimau, Vitalij Chachlou, Juras Hetsevich, Vadim Zahariev, Kuanysh Zhaksylyk	165
ONTOLOGICAL APPROACH TO CHINESE LANGUAGE INTERFACE DEVELOPMENT IN INTELLIGENT SYSTEMS Longwei Qian	171
TOOLS FOR CREATING AND MAINTAINING A KNOWLEDGE BASE BY INTEGRATING WOLFRAM MATHEMATICA SYSTEM AND NEVOD PACKAGE Valery Taranchuk, Vladislav Savionok	181
CONTROL TOOLS FOR REUSABLE COMPONENTS OF INTELLIGENT COMPUTER SYSTEMS OF A NEW GENERATION Maksim Orlov	191
INTEGRATION OF THIRD-PARTY FUNCTIONAL SERVICES ON A UNIFIED SEMANTIC BASIS Alexandr Zagorskiy	207
INTEGRATION OF LARGE LANGUAGE MODELS WITH KNOWLEDGE BASES OF INTELLIGENT SYSTEMS Kseniya Bantsevich, Mikhail Kovalev, Vitaly Tsishchanka, Natallia Malinovskaya, Aliaksei Andrushevich	213
OSTIS TECHNOLOGY INTEGRATION WITH THIRD-PARTY NLP SERVICES Alena Cherkas, Aliaksandr Kupo	219
EXAMPLES OF INTEGRATING WOLFRAM MATHEMATICA TOOLS INTO OSTIS APPLICATIONS Valery Taranchuk, Vladislav Savionok	225
INTELLIGENT HEALTH MONITORING SYSTEMS Vladimir Rostovtsev	237
TECHNOLOGY OF NEUROLOGICAL DISEASE RECOGNITION USING GATED RECURRENT UNIT NEURAL NETWORK AND INTERNET OF THINGS Uladzimir Vishniakou, Yiwei Xia, Chuyue Yu	241
HAND GESTURE RECOGNITION BASED ON SKELETAL IMAGE PROPERTIES Jun Ma, Viktor Tsviatkou, Anatoliy Boriskevich	247
AUTOMATION OF EDUCATIONAL ACTIVITIES WITHIN THE OSTIS ECOSYSTEM, Alena Kazlova, Aliaksandr Halavaty, Natalya Grakova	257
BUILDING A COMPREHENSIVE DIGITAL ARCHIVE FOR THE POLOTSK CADET CORPS: A CASE STUDY INTELLIGENT INFORMATION AND REFERENCE SYSTEM DESIGN Arkadzi Oskin	263
AN AUTOMATED APPROACH TO CHECKING USER KNOWLEDGE LEVELS IN INTELLIGENT TUTORING SYSTEMS Wenzu Li	269
MODEL OF INTEROPERABILITY OF INFORMATION SYSTEMS OF INFORMATION AND COMMUNICATION ENVIRONMENT OF SECONDARY SPECIAL EDUCATION INSTITUTION Nikolai Listopad, Lizaveta Bushchik	279

ABOUT CREATION OF THE INTELLIGENT TRANSPORTATION CONTROL SYSTEM IN RAILWAY TRANSPORT Aleksandr Erofeev	285
ADAPTIVE CONTROL SYSTEM FOR TECHNOLOGICAL PROCESS WITHIN OSTIS ECOSYSTEM Valery Taberko, Dzmitry Ivaniuk, Viktor Smorodin, Vladislav Prokhorenko	291
SUPPORT OF THE LIFE CYCLE OF INTELLIGENT GEOINFORMATION SYSTEMS FOR VARIOUS PURPOSES Sergei Samodumkin	299
ENSURING INFORMATION SECURITY OF THE OSTIS ECOSYSTEM Valery Chertkov, Vladimir Zakharau	305
SEMANTIC APPROACH TO DESIGNING APPLICATIONS WITH PASSWORDLESS AUTHENTICATION ACCORDING TO THE FIDO2 SPECIFICATION Anton Zhidovich, Alexei Lubenko, Iosif Vojteshenko, Alexey Andrushevich	311
MATHEMATICAL METHODS FOR ASSESSING INFORMATION SECURITY RISKS Aleksander Sobol, Viktor Kochyn, Natalia Grakova	317
AUTHOR INDEX	323

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ	11
ОСНОВНЫЕ НАПРАВЛЕНИЯ, ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ РАЗВИТИЯ ИНТЕЛЛЕКТУАЛЬНЫХ КОМПЬЮТЕРНЫХ СИСТЕМ НОВОГО ПОКОЛЕНИЯ И СООТВЕТСТВУЮЩЕЙ ИМ ТЕХНОЛОГИИ Голенков В. В., Гулякина Н. А.	15
ФАКТОР ЦИФРОВОЙ КУЛЬТУРЫ В ПРИМЕНЕНИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ЭКОНОМИКЕ И ОБРАЗОВАНИИ Паньшин Б. Н.	27
КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ СЛОЖНЫХ ИНТЕГРИРОВАННЫХ СИСТЕМ Кочин В. П., Курбацкий А. Н.	33
АССОЦИАТИВНЫЕ СЕМАНТИЧЕСКИЕ КОМПЬЮТЕРЫ ДЛЯ ИНТЕЛЛЕКТУАЛЬНЫХ КОМПЬЮТЕРНЫХ СИСТЕМ НОВОГО ПОКОЛЕНИЯ Голенков В. В., Шункевич Д. В., Гулякина Н. А., Ивашенко В. П., Захарьев В. А.	39
ОТКРЫТАЯ СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ КАК ФУНДАМЕНТ НОВОГО ПОКОЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ Татур М. М., Парамонов А. И.	61
ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ, СТРУКТУРА И ПЕРСПЕКТИВЫ РАЗВИТИЯ ПРОГРАММНОЙ ПЛАТФОРМЫ OSTIS-СИСТЕМ Зотов Н. В.	67
РЕАЛИЗАЦИЯ ИНФОРМАЦИОННО-ПОИСКОВОЙ ПОДСИСТЕМЫ В ПРОГРАММНОЙ ПЛАТФОРМЕ OSTIS-СИСТЕМ Зотов Н. В.	77
ИНТЕГРАЦИЯ ЛОГИЧЕСКИХ МОДЕЛЕЙ ПРЕДСТАВЛЕНИЯ И ОБРАБОТКИ ЗНАНИЙ В СМЫСЛОВОМ ПРОСТРАНСТВЕ Ивашенко В. П.	95
ПРИНЦИПЫ РЕШЕНИЯ ЗАДАЧ В РАСПРЕДЕЛЕННЫХ КОЛЛЕКТИВАХ ИНТЕЛЛЕКТУАЛЬНЫХ КОМПЬЮТЕРНЫХ СИСТЕМ НОВОГО ПОКОЛЕНИЯ Шункевич Д. В.	115
МОДЕЛЬ ВЕРСИОНИРОВАНИЯ НЕЙРОСЕТЕВЫХ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ Ковалёв М. В.	121
РЕДУЦИРОВАНИЕ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ В ИНТЕЛЛЕКТУАЛЬНЫХ КОМПЬЮТЕРНЫХ СИСТЕМАХ НОВОГО ПОКОЛЕНИЯ Крощенко А. А.	127
НЕЙРОСЕТЕВАЯ ПРОГРАММНАЯ ТЕХНОЛОГИЯ, ОБУЧАЕМАЯ НА ПРИНЦИПАХ СЛУЧАЙНОГО ПОИСКА Краснопрошин В. В., Мацкевич В. В.	133
РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ ПОТРЕБНОСТИ ПОЛЬЗОВАТЕЛЕЙ OSTIS СИСТЕМ Самодумкин С. А.	141
ПОСТРОЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ НА ОСНОВЕ KNOWLEDGE DISCOVERY IN DATASETS Краснопрошин В. В., Родченко В. Г., Карканица А. В.	147

ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ЭКОСИСТЕМЫ OSTIS Садовский М. Е.	153
ПРОЕКТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ВЗАИМОДЕЙСТВИЯ В ИММЕРСИВНЫХ СИСТЕМАХ НОВОГО ПОКОЛЕНИЯ Головатый А. И., Багай В. Д., Бернат Д. А., Головатая Е. А.	159
РАЗРАБОТКА СИСТЕМЫ РАСПОЗНАВАНИЯ ЗВУКОВ ПТИЦ С ИСПОЛЬЗОВАНИЕМ ОНТОЛОГИЧЕСКОГО ПОДХОДА Зеновко Е., Белявский Д., Кайгородова Л., Трофимов А., Хохлов В., Гецевич Ю., Захарьев В., Жаксылык К.	165
ОНТОЛОГИЧЕСКИЙ ПОДХОД К РАЗРАБОТКЕ КИТАЙСКО-ЯЗЫКОВОГО ИНТЕРФЕЙСА В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ Цянь Л.	171
ИНСТРУМЕНТЫ СОЗДАНИЯ И СОПРОВОЖДЕНИЯ БАЗЫ ЗНАНИЙ ПУТЕМ ИНТЕГРАЦИИ СИСТЕМЫ WOLFRAM MATHEMATICA И ПАКЕТА NEVOD Таранчук В. Б., Савёнок В. А.	181
СРЕДСТВА УПРАВЛЕНИЯ МНОГОКРАТНО ИСПОЛЬЗУЕМЫМИ КОМПОНЕНТАМИ ИНТЕЛЛЕКТУАЛЬНЫХ КОМПЬЮТЕРНЫХ СИСТЕМ НОВОГО ПОКОЛЕНИЯ Орлов М. К.	191
ИНТЕГРАЦИЯ СТОРОННИХ ФУНКЦИОНАЛЬНЫХ СЕРВИСОВ НА УНИФИЦИРОВАННОЙ СЕМАНТИЧЕСКОЙ ОСНОВЕ Загорский А. Г.	207
ИНТЕГРАЦИЯ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ С БАЗАМИ ЗНАНИЙ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ Банцевич К. А., Ковалев М. В., Тищенко В. Н., Малиновская Н. В., Андрушевич А. А.	213
ИНТЕГРАЦИЯ ТЕХНОЛОГИИ OSTIS СО СТОРОННИМИ NLP СЕРВИСАМИ Черкас Е. О., Купо А. Н.	219
ПРИМЕРЫ ИНТЕГРАЦИИ ИНСТРУМЕНТОВ WOLFRAM MATHEMATICA В ПРИЛОЖЕНИЯ OSTIS Таранчук В. Б., Савёнок В. А.	225
ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ МОНИТОРИНГА ЗДОРОВЬЯ Ростовцев В. Н.	237
ТЕХНОЛОГИЯ РАСПОЗНАВАНИЯ НЕЙРОЛОГИЧЕСКИХ ЗАБОЛЕВАНИЙ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННОЙ СЕТИ ЗАКРЫТОГО РЕКУРРЕНТНОГО БЛОКА И ИНТЕРНЕТА ВЕЩЕЙ Вишняков В. А., Ивей С., Чуюэ Ю.	241
РАСПОЗНАВАНИЕ ЖЕСТОВ РУК НА ОСНОВЕ СВОЙСТВ СКЕЛЕТИЗИРОВАННЫХ ИЗОБРАЖЕНИЙ Ма Ц., Цветков В. Ю., Борискевич А. А.	247
АВТОМАТИЗАЦИЯ ОБРАЗОВАТЕЛЬНОЙ ДЕЯТЕЛЬНОСТИ В РАМКАХ ЭКОСИСТЕМЫ OSTIS Козлова Е. И., Головатый А. И., Гракова Н. В.	257
СОЗДАНИЕ КОМПЛЕКСНОГО ЦИФРОВОГО АРХИВА ПОЛОЦКОГО КАДЕТСКОГО КОРПУСА: ПРИМЕР ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНОЙ ИНФОРМАЦИОННО-СПРАВОЧНОЙ СИСТЕМЫ Оськин А. Ф.	263

АВТОМАТИЗИРОВАННЫЙ ПОДХОД К ПРОВЕРКЕ УРОВНЯ ЗНАНИЙ ПОЛЬЗОВАТЕЛЕЙ В ИНТЕЛЛЕКТУАЛЬНЫХ ОБУЧАЮЩИХ СИСТЕМАХ Ли В.	269
МОДЕЛЬ ИНТЕРОПЕРАБЕЛЬНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ ИНФОРМАЦИОННО-КОММУНИКАЦИОННОЙ СРЕДЫ УЧРЕЖДЕНИЯ СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ Листопад Н. И., Бущик Е. А.	279
О СОЗДАНИИ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ УПРАВЛЕНИЯ ПЕРЕВОЗКАМИ НА ЖЕЛЕЗНОДОРОЖНОМ ТРАНСПОРТЕ Ерофеев А. А.	285
СИСТЕМА АДАПТИВНОГО УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ЦИКЛОМ ПРОИЗВОДСТВА В РАМКАХ ЭКОСИСТЕМЫ OSTIS Таберко В. В., Иванюк Д. С., Смородин В. С., Прохоренко В. А.	291
ПОДДЕРЖКА ЖИЗНЕННОГО ЦИКЛА ИНТЕЛЛЕКТУАЛЬНЫХ ГЕОИНФОРМАЦИОННЫХ СИСТЕМ РАЗЛИЧНОГО НАЗНАЧЕНИЯ Самодумкин С. А.	299
ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ЭКОСИСТЕМЫ OSTIS Чертков В. М., Захаров В. В.	305
СЕМАНТИЧЕСКИЙ ПОДХОД К ПРОЕКТИРОВАНИЮ ПРИЛОЖЕНИЙ С БЕСПАРОЛЬНОЙ АУТЕНТИФИКАЦИЕЙ ПО СПЕЦИФИКАЦИИ FIDO2 Жидович А. А., Лубенько А. А., Войтешенко И. С., Андрушевич А. А.	311
МАТЕМАТИЧЕСКИЕ МЕТОДЫ АНАЛИЗА РИСКА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ Соболь А. М., Кочин В. П., Гракова Н. В.	317
АВТОРСКИЙ УКАЗАТЕЛЬ	323

FOREWORD

The **main practical result** of the current stage of works in the field of Artificial intelligence is not only the creation of the next-generation intelligent computer systems ensuring effective interaction in solving comprehensive problems, but also the creation of a technological complex that ensures the rapid and high-quality building of such systems. This collection of scientific papers "Open semantic technology for intelligent systems design" is dedicated to these issues.

The development of this technological complex requires solving the following problems:

- clear identification of the logical-semantic level of intelligent computer systems, which abstracts from all possible options for the technical implementation of these systems (including the use of fundamentally new computers focused on their hardware support);
- development of an ontology for the design of intelligent computer systems and unification of the description of their logical-semantic models;
- ensuring the platform-independent character of the logical design of intelligent computer systems, the result of which is a unified description of the logical-semantic models of the designed systems;
- use of the methodology for component design of intelligent computer systems, which is based on a constantly replenished library of reusable components of these systems (reusable subsystems, knowledge base components, knowledge processing agents, user interface components);
- ensuring semantic compatibility of reusable components of intelligent computer systems and semantic compatibility of these systems themselves, as well as technologies for their design and support of subsequent stages of their life cycle.

The **main topic** of the collection of scientific papers "Open semantic technology for intelligent systems design" is various aspects of convergence and integration that ensure the transition to intelligent computer systems of a new generation and the corresponding technology of integrated support of their life cycle:

- convergence and integration of various models for information representation and processing in intelligent computer systems of a new generation:
 - convergence and integration of various knowledge types in the knowledge bases of intelligent computer systems of a new generation;
 - convergence and integration of various problem-solving models in intelligent computer systems of a new generation;
 - convergence and integration of various types of interfaces for intelligent computer systems of a new generation;
- convergence and integration of various directions of Artificial intelligence in order to build a general formal theory of intelligent computer systems of a new generation;
- convergence and integration of design technologies for various components of intelligent computer systems of a new generation in order to build a comprehensive technology for designing intelligent computer systems of a new generation;
- convergence and integration of technologies to support various stages of the life cycle for intelligent computer systems of a new generation in order to build a technology for integrated support of all stages of the life cycle for intelligent computer systems of a new generation;
- convergence and integration of various types of human activities in the field of Artificial intelligence (research activities, development of technological complex, applied engineering, educational activities) to increase the level of coherence and coordination of these activities, as well as to increase the level of their complex automation with the help of semantically compatible intelligent computer systems of a new generation;
- convergence and integration of various types and fields of human activities, as well as means of complex automation of this activity with the help of intelligent computer systems of a new generation.

The **main directions** of the collection of scientific papers «Open semantic technology for intelligent systems design»:

- Requirements for intelligent computer systems of a new generation;
- Principles underlying intelligent computer systems of a new generation;

- Requirements for the semantic representation of information. The universal language of the semantic representation of knowledge;
- Ontological stratification of the semantic representation of knowledge bases in intelligent computer systems of a new generation;
- An agent-oriented hierarchical model of hybrid problem solvers based on the semantic representation of knowledge bases;
- An ontological model of multimodal interfaces for intelligent computer systems of a new generation;
- A model of understanding information coming from outside, based on the semantic representation of knowledge bases;
- Flexibility, stratification, reflexivity, and learnability of intelligent computer systems of a new generation;
- Unification, standardization, and semantic compatibility of intelligent computer systems of a new generation — consistency of concepts and terms;
- Interoperability (the ability to interact effectively) of intelligent computer systems of a new generation — the ability to understand each other, negotiate, and coordinate their actions in the collective solution of complex problems;
- An integrated technology for designing intelligent computer systems of a new generation;
- Platforms for the implementation of intelligent computer systems of a new generation. Software platforms and associative semantic computers focused on the implementation of intelligent computer systems of a new generation;
- Convergence of logical-semantic models of intelligent computer systems of a new generation and architectures of associative semantic computers providing interpretation of these models. Wave programming languages;
- Principles underlying the Technology of integrated support of the life cycle for intelligent computer systems of a new generation;
- A global Ecosystem of intelligent computer systems of a new generation, providing complex convergence and automation of all kinds of human activities.

The collection contains 37 articles. The editorial board of the collection thanks all the authors who submitted their articles. For publication, scientific experts selected the best of the submitted papers, many of them were revised in accordance with the comments of the reviewers.

We also thank the experts for their great work in reviewing articles in close cooperation with the authors, which allowed increasing the level of presentation of scientific results and also created a platform for further scientific discussions.

We hope that, as before, the collection will fulfill its main function — to promote active cooperation between business, science, and education in the field of Artificial intelligence.

Editor-in-chief
Golenkov Vladimir

ПРЕДИСЛОВИЕ

Основным **практическим результатом** текущего этапа работ в области Искусственного интеллекта является создание не только интеллектуальных компьютерных систем следующего поколения, обеспечивающих эффективное взаимодействие при решении комплексных задач, но и создание технологического комплекса, обеспечивающего быстрое и качественное построение таких систем. Этим вопросам посвящён, данный сборник научных трудов «Открытые семантические технологии проектирования интеллектуальных систем».

Разработка указанного технологического комплекса требует решения следующих задач:

- чёткого выделения логико-семантического уровня интеллектуальных компьютерных систем, который абстрагируется от всевозможных вариантов технической реализации этих систем (в том числе и от использования принципиально новых компьютеров, ориентированных на их аппаратную поддержку);
- разработки онтологии проектирования интеллектуальных компьютерных систем и унификации описания их логико-семантических моделей;
- обеспечения платформенно независимого характера логического проектирования интеллектуальных компьютерных систем, результатом которого является унифицированное описание логико-семантических моделей проектируемых систем;
- использования методики компонентного проектирования интеллектуальных компьютерных систем, в основе которой лежит постоянно пополняемая библиотека многократно используемых компонентов этих систем (многократно используемых подсистем, компонентов баз знаний, агентов обработки знаний, компонентов пользовательских интерфейсов);
- обеспечения семантической совместимости многократно используемых компонентов интеллектуальных компьютерных систем и семантической совместимости самих этих систем, а также технологий их проектирования и поддержки последующих этапов их жизненного цикла.

Основная тема сборника научных трудов «Открытые семантические технологии проектирования интеллектуальных систем» — различные аспекты конвергенции и интеграции, обеспечивающие переход к интеллектуальным компьютерным системам нового поколения и соответствующей технологии комплексной поддержки их жизненного цикла:

- конвергенция и интеграция различных моделей представления и обработки информации в интеллектуальных компьютерных системах нового поколения:
 - конвергенция и интеграция различных видов знаний в базах знаний интеллектуальных компьютерных систем нового поколения;
 - конвергенция и интеграция различных моделей решения задач в интеллектуальных компьютерных системах нового поколения;
 - конвергенция и интеграция различных видов интерфейсов интеллектуальных компьютерных систем нового поколения;
- конвергенция и интеграция различных направлений Искусственного интеллекта в целях построения общей формальной теории интеллектуальных компьютерных систем нового поколения;
- конвергенция и интеграция технологий проектирования различных компонентов интеллектуальных компьютерных систем нового поколения в целях построения комплексной технологии проектирования интеллектуальных компьютерных систем нового поколения;
- конвергенция и интеграция технологий поддержки различных этапов жизненного цикла интеллектуальных компьютерных систем нового поколения в целях построения технологии комплексной поддержки всех этапов жизненного цикла интеллектуальных компьютерных систем нового поколения;
- конвергенция и интеграция различных видов человеческой деятельности в области Искусственного интеллекта (научно-исследовательской деятельности, развития технологического комплекса, прикладной инженерии, образовательной деятельности) для повышения уровня согласованности и координации этих видов деятельности, а также для

повышения уровня их комплексной автоматизации с помощью семантически совместимых интеллектуальных компьютерных систем нового поколения

- конвергенция и интеграция самых различных видов и областей человеческой деятельности, а также средств комплексной автоматизации этой деятельности с помощью интеллектуальных компьютерных систем нового поколения.

Основные **направления** сборника научных трудов «Открытые семантические технологии проектирования интеллектуальных систем»:

- Требования, предъявляемые к интеллектуальным компьютерным системам нового поколения;
- Принципы, лежащие в основе интеллектуальных компьютерных систем нового поколения;
- Требования, предъявляемые к смысловому представлению информации. Универсальный язык смыслового представления знаний;
- Онтологическая стратификация смыслового представления баз знаний в интеллектуальных компьютерных системах нового поколения;
- Агентно-ориентированная иерархическая модель гибридных решателей задач, основанных на смысловом представлении баз знаний;
- Онтологическая модель мультимодальных интерфейсов интеллектуальных компьютерных систем нового поколения;
- Модель понимания информации, поступающей извне, основанная на смысловом представлении баз знаний;
- Гибкость, стратифицированность, рефлексивность и обучаемость интеллектуальных компьютерных систем нового поколения;
- Унификация, стандартизация и семантическая совместимость интеллектуальных компьютерных систем нового поколения — согласованность понятий и терминов;
- Интероперабельность (способность к эффективному взаимодействию) интеллектуальных компьютерных систем нового поколения — способность к взаимопониманию, договороспособность, способность к координации своих действий при коллективном решении комплексных задач;
- Комплексная технология проектирования интеллектуальных компьютерных систем нового поколения;
- Платформы реализации интеллектуальных компьютерных систем нового поколения. Программные платформы и ассоциативные семантические компьютеры, ориентированные на реализацию интеллектуальных компьютерных систем нового поколения;
- Конвергенция логико-семантических моделей интеллектуальных компьютерных систем нового поколения и архитектур ассоциативных семантических компьютеров, обеспечивающих интерпретацию указанных моделей. Волновые языки микропрограммирования;
- Принципы, лежащие в основе Технологии комплексной поддержки жизненного цикла интеллектуальных компьютерных систем нового поколения;
- Глобальная Экосистема интеллектуальных компьютерных систем нового поколения, обеспечивающая комплексную конвергенцию и автоматизацию всевозможных видов человеческой деятельности.

Сборник содержит 37 статей. Редакция сборника благодарит всех авторов, представивших свои статьи. Для публикации научными экспертами были отобраны лучшие из представленных работ, многие из них были переработаны в соответствии с замечаниями рецензентов.

Мы также благодарим экспертов за большой труд по рецензированию статей в тесном взаимодействии с авторами, который позволил повысить уровень изложения научных результатов, а также создал платформу для дальнейших научных дискуссий.

Надеемся, что, как и прежде, сборник будет выполнять свою основную функцию — способствовать активному сотрудничеству между бизнесом, наукой и образованием в области искусственного интеллекта.

Главный редактор

Голенков Владимир Васильевич

The Main Directions, Problems and Prospects of the Development of the Next-Generation Intelligent Computer Systems and the Corresponding Technology

Vladimir Golenkov, Natalia Gulyakina
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: golen@bsuir.by, guliakina@bsuir.by

Abstract—The paper considers the main directions, problems, and prospects for the development of next-generation intelligent computer systems and corresponding technologies, with a focus on the current state of work in the field of Artificial Intelligence. It highlights the problems and methodological challenges of the current stage of development, as well as the prerequisites for transitioning to intelligent computer systems of the new generation. The text also provides background and history of OSTIS Technology development, and outlines its features, advantages, and novelty. Additionally, it mentions current projects related to the development of OSTIS Technology in the current stage of work.

Keywords—Next-generation intelligent computer system, self-learning intelligent computer system, interoperable intelligent computer system, individual subject, collective of subjects, hierarchical subject, social responsibility[^], interoperability[^], individual activity, collective activity, intelligence of a collective of subjects[^], strategic problem of the subject^{*}, subproblem^{*}, Society, OSTIS Ecosystem .

I. THE FEATURE OF THE CURRENT STATE OF WORK IN THE FIELD OF ARTIFICIAL INTELLIGENCE — THE TRANSITION TO THE NEXT-GENERATION INTELLIGENT COMPUTER SYSTEMS

The epicenter of the modern stage of automation of human activity is a low level of automation and high overhead costs

- for the system integration of various computer systems, in other words, for the creation of complicated hierarchical computer complexes;
- for the modernization of computer systems during their operation [1].

Modern computer systems clearly lack the *intelligence* and *autonomy* to automate these aspects of human activity.

The necessity to move from modern *computer systems* (including modern *intelligent computer systems*) to *next-generation intelligent computer systems* is due to the necessity to move to automate more and more complex types and areas of *human activity* requiring the creation of whole complexes of *intelligent computer systems* that

can independently *evolve* and *interact* effectively with each other in the *collective solution of complex problems*.

Computer systems with these abilities are the *new generation computer systems*. Since these *computer systems* cannot but have a high level of *intelligence*, they should also be referred to as *next-generation intelligent computer systems*. A high level of *intelligence* is necessary for next-generation computer systems:

- to adequately assessment their own competence and the competence of their partners;
- to ensure mutual understanding, contractual capacity and coordination (consistency) of their actions with the actions of partners in the course of *collective solution of complex problems* in the conditions of possible occurrence of unpredictable (non-standard) circumstances.

Obviously, the creation and operation of the *next-generation intelligent computer systems* requires:

- to develop a general formal theory of such systems;
- to develop a comprehensive technology for designing and supporting the subsequent stages of the life cycle of these systems;;
- to develop a general formal theory of the whole variety of types and areas of human activity that it is advisable to automate.

next-generation intelligent computer system

= (self-learning intelligent computer system \cap interoperable intelligent computer system) [2]

self-learning intelligent computer system

:= [an intelligent computer system that has a high rate of self-realized evolution, which results in a significant reduction in the complexity (costs, overhead) of its modernization]

self-learning of an intelligent computer system

\Rightarrow suggests*:

- ability to monitor the state and dynamics of the environment and adjust their actions with appropriate environmental changes (adaptability);
- ability to analyze and improve the quality of your own knowledge base (structuring and analysis of contradictions, information holes, information garbage);
- ability to extract knowledge from external sources of information;
- ability to analyze and improve the quality of their own activities (including the ability to learn from their own mistakes);
- ability to analyze the quality of the activities of other subjects and benefit from it for yourself (learn from the mistakes of others).

high level of self-learning of an intelligent computer system

⇒ is provided by*:

- high level of flexibility of an intelligent computer system
- high level of stratification of an intelligent computer system
- high level of reflexivity of an intelligent computer system
- high level of cognitive activity

interoperable intelligent computer system

:= [computer system capable of independent effective interaction with other systems]

interoperability of an intelligent computer system

⇒ suggests*:

- ability to understand other systems and its users
⇒ suggests*:
semantic compatibility with interacting systems and users
- negotiability
- ability to coordinate their actions with the actions of partners

Our proposed approach to the construction of the next-generation intelligent computer systems is based on the following principles:

- semantic representation of knowledge stored in the memory of *next-generation intelligent computer systems*;
- ontological structuring and systematization of knowledge stored in memory;
- decentralized situational agent-oriented organization of *problem solving processes*;
- convergence and deep (diffuse) integration of various

problem solving models and, as a consequence, the hybrid nature of *problem solvers*;

- semantic integration of input information entering an individual intelligent computer system from the outside through different sensory channels and in different languages by translating input information into a common universal *language of internal semantic representation of knowledge*..

it should be distinguished*

- ⇒ {
- *individual next-generation intelligent computer system*
 - *collective next-generation intelligent computer system*
- }

collective next-generation intelligent computer system

⇒ subdividing*:

- {
- *collective of next-generation individual intelligent computer systems*
 - *hierarchical collective of next-generation intelligent computer systems*

:= [collective of next-generation intelligent computer systems, whose members can be both collective and *individual next-generation intelligent computer systems*]

}

individual next-generation intelligent computer system

⇒ features*:

- *individual next-generation computer system* cannot be decomposed into subsystems that can be developed absolutely independently of each other and coordinated only by inputs and outputs, implementing the "black box" principle.
- In an *individual next-generation intelligent computer system*, convergence, compatibility and "meaningful" interaction of various types of knowledge and models of problem solving are necessary. That is, an *individual next-generation intelligent computer system* should be a *hybrid system*.

OSTIS Technology

:= [Our proposed Technology for developing and maintaining *next-generation intelligent computer systems*]

:= [Open Semantic Technology for Intelligent Systems [3]]

⇒ requirements*:

- complexity — *OSTIS Technology* ensures the compatibility of all private *Artificial intelligence* technologies; compatibility, self-learning and interoperability of *intelligent computer systems* being developed, as well as support not only for the design of *intelligent computer systems*, but also for their entire life cycle
- universality — *OSTIS Technology* is focused on the development and maintenance of *next-generation intelligent computer systems* for any purpose
- self-learning — *OSTIS Technology* ensures the permanent evolution of the *OSTIS Technology* itself (itself) due to the fact that it is implemented in the form of the *next-generation intelligent computer system* that "knows" the *OSTIS Technology* and "knows" how to use it

ostis-system

:= [*intelligent computer system* based on *OSTIS Technology*]

OSTIS Ecosystem

:= [The main product of *OSTIS Technology*, which is a global network of *ostis-systems*]

∈ *hierarchical collective of intelligent computer systems of the new generation*

The main components of *OSTIS Technology* are:

- ***OSTIS Standard***
:= [The standard of next-generation intelligent computer systems as well as methods, methods and means of supporting their life cycle]
:= [*OSTIS Technology Standard*]
- ***OSTIS metasystem***
:= [The core of the *ostis-systems* lifecycle support automation system]
- ***OSTIS Library***
:= [Distributed library of typical (reusable) components of *ostis-systems*]

II. PROBLEMS OF THE CURRENT STAGE OF DEVELOPMENT OF THE THEORY AND TECHNOLOGY OF NEXT-GENERATION INTELLIGENT COMPUTER SYSTEMS

The creation of *next-generation intelligent computer systems* requires answers to the following questions:

- What are the requirements for *intelligent computer systems* that provide the above complex automation of *human activities*;
- Why modern intelligent computer systems do not meet these requirements and, accordingly, why the transition to a fundamentally *next-generation intelligent computer systems* is necessary;

- What fundamental principles should underlie the *next-generation intelligent computer systems*;
- What principles should underlie the most automated technology for designing and supporting the entire life cycle of *next-generation intelligent computer systems*;
- What principles should underlie the structure and organization of various types and areas of *human activity* to ensure its comprehensive and maximum possible automation with the help of *next-generation intelligent computer systems* (as you know, before automating any *human activity*, it is necessary to put it in order — automating disorder leads to even greater disorder).

The current fundamental problems of creating the theory and technology of *next-generation intelligent computer systems* include:

- Development of a theory of *hierarchical multi-agent systems* in which agents are individual or collective *intelligent computer systems that are interoperable*.
- Unification and standardization of various models of knowledge representation and processing. The effect of this unification will not be visible immediately. But if this does not happen, we will never come to an effective comprehensive automation of *human activity*. The eclectic variety of automation methods and tools leads not only to unjustified duplication of the systems being developed, but also to an increase in the complexity of their use and maintenance.
- Convergence and integration of different areas of *Artificial Intelligence*.
Currently, various areas of *Artificial Intelligence* have a fairly high level of development (signal processing, natural language processing, logical models, artificial neural networks, ontological models, multi-agent models, and many others). Integration of all these directions is a rather time-consuming problem, but it is quite a solvable one, which is based on the harmonization of related notions.
- Convergence of such activities in the field of *Artificial Intelligence* as:
 - training of specialists in the field of *Artificial Intelligence*;
 - engineering activities for the development of applied *next-generation intelligent computer systems*;
 - development of *technology* for designing and supporting the life cycle of *next-generation intelligent computer systems*;
 - research activities in the field of *Artificial intelligence*.
- In order to develop the Technology of *next-generation intelligent computer systems*, it is also necessary to converge this Technology with all

types and areas of *human activity* that are not part of the activities in the field of *Artificial Intelligence*. The development of the technology of *next-generation intelligent computer systems* has a pronounced interdisciplinary character. This means that all the knowledge accumulated by *human society* in various fields should be represented as part of the *Global Knowledge Base* of the Ecosystem of *next-generation intelligent computer systems* (using portals of scientific, technical, administrative and other knowledge), should be clearly stratified in the form of a hierarchical system of semantically compatible reusable *ontologies* and transformed into a hierarchical system semantically compatible *formal* components of knowledge bases of *intelligent computer systems* for various applications.

- Ensuring semantic compatibility of *next-generation intelligent computer systems* not only at the stage of their design, but also at all subsequent stages of their life cycle.
- Development of a model of collective behavior of *next-generation intelligent computer systems*, in other words, a model of decentralized collective problem solving at the level of:
 - a *multi-agent system* whose agents are internal agents of an *individual intelligent computer system* interacting through a common memory (through a common *knowledge base* stored in one memory);
 - a *multi-agent system* whose agents are *interoperable intelligent computer systems* interacting through a common *knowledge base* stored in the memory of a *corporate intelligent computer system* or in the memory of the coordinator of the activities of a temporary collective of *intelligent computer systems*.

Within the framework of the theory of *collective problem solving*, the following problem situations can be distinguished:

- a problem that can be solved by the *individual intelligent computer system* in which this task is initiated;
- a problem corresponding to the competence of the collective of *intelligent computer systems* within which this problem was initiated;
- a problem that goes beyond the competence of the *collective of intelligent computer systems* within which this problem was initiated. Such a problem requires the formation of a temporary team, the coordinator (but not the manager) of which becomes the *intelligent computer system* within which the specified problem was initiated. To do this, it is necessary to find those *intelligent computer systems* that together will provide the necessary competence. At the same time, note that each *interoperable*

intelligent computer system (both individual and collective) must know its competence in order to determine whether it can or cannot solve a given (arisen) task. This, in particular, is necessary for the formation of temporary collectives of *intelligent computer systems*.

- Development of the principles underlying a powerful Library of reusable and compatible components of *next-generation intelligent computer systems*, which provides full automation of the integration of these components in the assembly process of the designed systems.
- Development of methods and means of permanent expansion of the Library of reusable components of *next-generation intelligent computer systems* in various fields of *human activity*:
- Scientific and technical activity in any field should be reduced to the development of *knowledge bases* of various *intellectual portals of scientific and technical knowledge*. At the same time, the *knowledge base* of each such portal should be decomposed into fragments included in the Library of reusable components of *knowledge bases* of *next-generation intelligent computer systems*, which can hierarchically fit into each other. To do this, the specified components must be specified accordingly.
- Developers of any *intelligent computer system* should decompose the developed system into a set of components included in the Library of components of *next-generation intelligent computer systems* — so that the development of any similar system is reduced to the assembly of components from this Library.
- All(!) developers should take care of expanding the Library of reusable (typical) components of *next-generation intelligent computer systems*, which will significantly reduce the complexity of developing new *next-generation intelligent computer systems* within the Ecosystem of such systems. At the same time, the authorship of the components of the specified Library should be encouraged, which is the fundamental basis for the development of the knowledge market, the knowledge economy.

If we competently develop and use the Technology of *next-generation intelligent computer systems*, then the development of any new *intelligent computer system* will mainly be reduced to its automatic assembly from the components of this system specified by the developer. Some components of the *intelligent computer system* being developed may be included in the current state of the Library of Components of *next-generation intelligent computer system*, and some of them will require additional development. But at the same time, each such new component is most often the result of modification of existing components from the specified

Library and must be specified and included in this Library. Thus, the developer of an applied *intelligent computer system* should develop not only this system, but also contribute to the development of a Library of Components of *next-generation intelligent computer systems*, as a result of which the next *intelligent computer system* he is developing can be assembled without additional components being developed, but only from components of the Component Library. If all developers of applied systems act in this way, then the rate of increasing the level of automation of *human activity* will increase significantly.

III. METHODOLOGICAL PROBLEMS OF THE CURRENT STAGE OF WORK IN THE FIELD OF ARTIFICIAL INTELLIGENCE

A. *Social responsibility of specialists in the field of Artificial intelligence*

The current stage of development of the theory and practice of *Artificial Intelligence* exposes a whole range of problems that hinder this development [4], [5]. Further development of *Artificial Intelligence* technologies

- on the one hand, it can and quite quickly make the transition of modern society to a fundamentally new level of its evolution, providing comprehensive automation of all types and areas of *human activity* subject to automation, as well as providing the maximum possible comfort and the maximum possible disclosure of the creative potential of each person;
- on the other hand, it can quite long and quite convincingly imitate the specified progress of automation of *human activity* for an illiterate layman — any even very worthy goal can be ruined by the imitation of its achievement;
- on the third hand, can quickly enough lead *human society* to degradation and self-destruction.

As a result, at the present stage of development of *Artificial intelligence* technologies, the level of social responsibility of specialists in the field of *Artificial intelligence* is a determining factor in the development of *human society*. The danger to *human society* does not come from *intelligent computer systems*, but from the motivation of specialists who develop these systems. It is obvious that the creation of *intelligent computer systems* designed for the deliberate infliction of any damage to *human society*, and requiring the creation of appropriate intellectual means of ensuring security, is a shortcut to self-destruction.

The efforts of specialists in the field of *Artificial intelligence* should be aimed at significantly increasing the level of intelligence of *human society* as a whole, the basis of which is the complex automation of all those types and areas of *human activity* that fundamentally makes sense to automate.

B. *The global goal of Artificial Intelligence activities*

Why the current stage of activity in the field of *Artificial intelligence* requires the formulation of the global goal of this activity and its permanent clarification.

The current state of *Artificial Intelligence* can be characterized as a deep methodological crisis caused by:

- the fact that scientific results in this area came out of scientific laboratories and began to have a real practical impact;
- the lack of understanding that obtaining serious scientific results in a particular field and the creation of technologies that ensure the effective practical use of these results are tasks commensurate in importance and complexity. This is especially true for *Artificial Intelligence*.

The latter circumstance leads to unjustified euphoria, the illusion of well-being and to a burgeoning eclecticism that completely ignores even the seemingly obvious laws of the general theory of systems.

Unfortunately, local implementation of the results of scientific research in the field of *Artificial intelligence*, local automation of business processes of any organization without taking into account the system organization of the entire complex of methods and automation tools of various types and areas of human activity leads to unjustified duplication of results.

If in the near future there is no awareness of the global (strategic) goal of work in the field of *Artificial Intelligence*, then activities in this area as a whole will be carried out in the style of "swan, cancer and pike". Wasting effort will not lead to a holistic, practically meaningful result. The "vectors" of specific areas of this activity, the "vectors" of our efforts will not have the same focus, which will significantly reduce the overall productivity of all these activities and the quality of the overall (total) result.

What should be the *strategic problem* (super problem) that experts in the field of *Artificial intelligence* should solve. It is obvious that such a super task is the transition of the entire complex of *human activity* to a fundamentally new level of maximum possible automation, within which *creative* activity remains a fundamentally non-automated part of human activity, in particular, research activities, teaching and educational activities, permanent increase in the level of complex automation of *human activity*. The main goal of complex automation of *human activity* is not only to automate what can be effectively automated using *Artificial Intelligence* methods, but to automate all(!) "bottlenecks" of *human activity* that determine its overall performance in various fields.

As a result, at present, *Artificial Intelligence* technologies are on the threshold of transition to a fundamentally new level of development — on the threshold of transition from solving private (local) problems to solving the global problem of complex automation of all types and areas

of *human activity*, which requires automation of solving not only private urgent and important problems, but also automation of the solution more and more high-level problems, for which the problems that are being automated now become sub-problems. In other words, when automating the solution of complex problems (supervising problems), automation focuses on the development of methods and means of interaction between the means of solving local problems (particular problems).

Shifting the focus to automation of solving not just *intellectual problems*, but to automation of solving complex problems, the sub-problems of which are a variety of intellectual problems, not only takes *Artificial Intelligence* technologies to a fundamentally new level, but will also have a significant impact on all aspects of human activity:

- research and scientific and technical work should acquire a convergent mutually enriching character;
- the basis of education should be interdisciplinarity;
- the basis of global automation of *human activity* should be a general complex formal and permanently improved theory of *human activity*, which should be based on an interdisciplinary convergent methodology aimed at overcoming the eclectic approach.

Consequently, the main goal of complex automation of various types and areas of human activity with the help of *interoperable intelligent computer systems* is to significantly increase the *level of intelligence* of human society as a whole.

Modern *human society* — is a complex distributed multi-agent *cybernetic system*, the development of which is carried out, unfortunately, in violation of many laws of Cybernetics and, in particular, in violation of the criteria that determine the level of *intelligence* of hierarchical multi-agent systems. The level of *intelligence* of such systems is determined by a number of seemingly obvious factors:

- what is the volume and *quality of knowledge* accumulated by a *multi-agent system* and available to all agents (subjects) included in this system
 - how much of this *knowledge* is sufficient to organize the management of the activities of this system;
 - to what extent is this knowledge correct (consistent) and adequate;
 - how great is the convergence, compactness and purity of this *knowledge* (the presence of information garbage, information duplication is taken into account here);
 - how well structured (systematized) is the accumulated knowledge;
- how each agent of a *multi-agent system* has access to the knowledge stored in the shared memory of the entire *multi-agent system*;

- how this *knowledge* accumulates and evolves, how a multi-agent system learns itself
 - how a *multi-agent system* learns from its own mistakes,
 - how a *multi-agent system* improves the quality of its knowledge;
- how the *multi-agent system* as a whole and each agent in particular use the *knowledge* accumulated in the shared memory to solve various *problems*.

As a result, if we consider modern *human society* from the standpoint of the theory of *multi-agent systems*, which are communities of *intelligent systems* (not only artificial, but also natural intelligent systems), it is obvious that the next stage of its evolution requires:

- automation of accumulation, analysis and permanent improvement of the ***quality of knowledge*** accumulated by mankind;
- automation of the effective use of knowledge accumulated by mankind in solving problems of various levels that require the formation of various short-term or long-term ***communities of people and intelligent computer systems***. Each such community is intended either to solve any one specific problem, or to solve some set of problems in some area;
- increasing the level of ***convergence*** of knowledge, methods, actions, as well as new technical systems being created;
- improving the level of ***interoperability*** for both intelligent computer systems and people.

C. *General requirements for specialists in the field of Artificial intelligence*

⇒ *epigraph**:

- The requirements for specialists in the field of *Artificial intelligence* at the new stage of development of this field are a reflection of the requirements for next-generation intelligent computer systems and related technologies
- The level of *intelligence* (including collective intelligence) of *intelligent computer system* developers cannot be lower than the level of *intelligence* of *intelligent computer systems* being created
- The level of *intelligence* of a team of agents is not always higher than the level of *intelligence* of its agents
- The development of *interoperable intelligent computer systems* can only be collective
- A collective of non-interoperable developers cannot create *interoperable intelligent computer systems*

The high level of *social responsibility* required of *specialists in the field of Artificial Intelligence* imposes on

them a number of obvious, but, unfortunately, often overlooked general requirements necessary for high-quality participation in complex collective socially significant projects. Such general requirements include:

- a high level of **motivation** to participate in the permanent evolution of an integral technological complex that provides the development of effective *interoperable intelligent computer systems*. A comprehensive and high-quality technology for the development and maintenance of *interoperable intelligent computer systems* should be considered as a key product of collective activity in the field of *Artificial Intelligence*. This motivation implies appropriate purposefulness, lack of selfishness, arrogance, individualism, isolationism, parasitism;
- high level of **creative activity**, passionarity, courage;
- a high level of **reflection** — is the ability to analyze one's own goals and actions and correct one's own mistakes, as well as analyze the goals, actions and mistakes made by the team of which the specialist is a member. It is one thing to sincerely recognize the logic and expediency of observing certain rules (principles, requirements), and it is quite another thing to be able to see and correct your own violations of these rules. Without such reflection, the progress of collective creativity is impossible. Knowing how to do it and actually following — it is not the same thing.
- high level of **own interoperability**:
 - the ability to *understand each other* and ensure *semantic compatibility*, which requires permanent monitoring of the current state and evolution of the technological complex;
 - *contractual capacity* — the ability to quickly coordinate their goals and plans, denotational semantics of concepts and terms, as well as decentralize the distribution of sub-problems of a collectively solved problem;
 - *ability to coordinate* and synchronize their actions with colleagues in conditions of possible occurrence of unpredictable circumstances.

Without a high level of developer interoperability, it is impossible to ensure:

- **convergence**, unification, standardization of *interoperable intelligent computer systems*;
- formation of a powerful **Library of standard components of next-generation intelligent computer systems**;
- a significant reduction in labor intensity and an increase in the level of automation of the development and maintenance of *next-generation intelligent computer systems*;

- building a general theory of the ***Ecosystem of the next-generation intelligent computer systems*** and, accordingly, a general theory of *human activity*.

As a result, in order to create *interoperable intelligent computer systems*, it is necessary that their creators themselves have a high level of *interoperability*. The problem of ensuring this is the main challenge that is addressed to specialists in the field of *Artificial intelligence* at the current stage of development of this field.

The main reason that prevents the formation of the necessary level of *interoperability* among specialists in the field of *Artificial intelligence* is the competitive style of relationships between specialists. This style of relationship is a widespread way to stimulate employee activity. But this is not the only way to stimulate creative activity in solving strategically important problems, which, in particular, is the problem of effective complex automation of all types and areas of *human activity* with the help of *interoperable intelligent computer systems*. Moreover, competition provokes selfishness and ignoring the interests of other subjects (including the interests of the collective of which the subject is a member). Thus, competition clearly contradicts the principles of *interoperability* and, accordingly, the principles of the organization of *intellectual communities*, intellectual creative teams and organizations.

It is necessary to move from a competitive style of relationships to mutually beneficial interaction between subjects of all levels of the hierarchy. This is the main essence of *interoperability* and the transition to *intellectual collectives* and an intellectual society.

It should be noted that the listed general requirements for specialists in the field of Artificial intelligence at the present stage of the development of *Artificial intelligence technologies* should be imposed not only on them, but also on all people who are ready to contribute to technological progress. It's just that at this stage, the main responsibility for this lies precisely with specialists in the field of *Artificial intelligence*.

D. Requirements for the fundamental training of specialists in the field of Artificial intelligence

The need to significantly increase the level of practical significance and efficiency of work in the field of *Artificial intelligence*, which requires a transition to the *next-generation intelligent computer systems* and to a fundamentally new technological complex, imposes on specialists in the field of *Artificial intelligence* not only general requirements necessary for effective participation in complex collective socially significant projects, but also high requirements for their fundamental professional training:

- a high level of system culture that allows you to "see" the hierarchy of complex systems, the

connections between different levels and hierarchies, the difference between tactical and strategic tasks;

- a high level of mathematical culture, culture of formalization;
- a high level of technological culture and technological discipline;
- a high level of self-learning in a rapidly changing technological infrastructure

E. Problems of the current stage of development of the theory and technology of next-generation intelligent computer systems

Let's list the main methodological problems of the current stage of work in the field of Artificial Intelligence, which hinder the solution of the fundamental tasks discussed above:

- An insufficiently high level of awareness by specialists in the field of *Artificial intelligence* of their social responsibility.
- The lack of a coordinated awareness of the global goal of work in the field of *Artificial intelligence*, which is to gradually increase the **level of intelligence** of *human society* by complex automation of all aspects of its activities using a network of interacting *intelligent computer systems*.
- The insufficiently high level of *interoperability* of specialists in the field of *Artificial intelligence* and the predominance of a competitive style of relationships. The consequence of this is an insufficient number of motivated specialists in the field of *Artificial Intelligence*, capable of effective creative interaction. In order for them to appear in sufficient numbers, a good system of their professional training is not enough. It should also be noted that good human relations, psychological atmosphere and Team Building in the development team, which many companies take seriously, is a necessary, but far from sufficient condition for the effectiveness of collective development of complex computer systems (especially for *next-generation intelligent computer systems*).
- Insufficiently high level of comprehensive fundamental training of specialists in the field of *Artificial intelligence*.
- The pronounced interdisciplinary nature of *Artificial Intelligence* as a field of human activity, requiring specialists to be able to work at the junctions of sciences.
- Lack of awareness of the need for deep *convergence* between various areas of *Artificial Intelligence* and the formalization of the entire complex of knowledge in the field of *Artificial intelligence* for their use in the *knowledge bases* of intelligent computer

systems (first of all, instrumental *intelligent computer systems* that are part of the technological complex for the development and maintenance of *intelligent computer systems* for various purposes).

- The high level of complexity of the complex formalization of all knowledge accumulated by mankind (primarily in the field of mathematics and general systems theory) and their convergence with the complex of knowledge accumulated and formalized in the field of *Artificial intelligence*. This is necessary for the direct use of the knowledge accumulated by mankind in *intelligent computer systems* for various purposes.
- Lack of awareness of the need for deep *convergence* and consistency between
 - research activities in the field of *Artificial intelligence*;
 - activities aimed at the development of private *artificial intelligence* technologies, as well as integrated technology for designing and supporting the life cycle of *intelligent computer systems*;
 - engineering activities aimed at the development of specific *intelligent computer systems* for various purposes;
 - educational activities aimed at training specialists in the field of *Artificial intelligence*.
- The problem of ensuring **semantic compatibility** of *next-generation intelligent computer systems* not only at the stage of their design, but also throughout their life cycle in the conditions of permanent evolution of *intelligent computer systems* themselves during their operation, as well as the permanent evolution of complex technology of their development [6], [7].

The main part of these problems is the need to move to a fundamentally new style and organization of interaction between specialists in the field of *Artificial Intelligence*, without which it is impossible to move from private theories of *Artificial Intelligence* to a **General theory of intelligent computer systems** that ensures compatibility of all private theories of *Artificial Intelligence*, as well as the transition from private *Artificial intelligence* technologies to **Complex Artificial Intelligence technology**, ensuring compatibility of all private *artificial intelligence* technologies. The transition to a new style of interaction of specialists in the field of *Artificial intelligence* is based on the transition from competition to synergetic mutually beneficial interaction aimed at convergence and deep integration of private (local) results, which will lead to the transformation of the modern community of specialists in the field of *Artificial intelligence* into an **intellectual community**.(see [8]).

IV. PREREQUISITES FOR THE TRANSITION TO INTELLIGENT COMPUTER SYSTEMS OF THE NEW GENERATION

- An actively expanding variety of information resources and services, the efficiency of which has a low level due to the lack of their systematization and compatibility [9], [10]
- The emergence of formal ontologies as a means of ensuring semantic compatibility of information resources accumulated by mankind, Semantic Web
- Active development of the theory of multi-agent systems, their self-organization, emergence, synergy, theory of intellectual communities and organizations
- Development of the theory of decentralized situational management ("the orchestra plays without a conductor")
- Smart home, smart hospital, smart city
- Industry 4.0, University 4.0
- The appearance of works aimed at clarifying the cybernetic principles underlying Society 5.0

V. BACKGROUND AND HISTORY OF OSTIS TECHNOLOGY DEVELOPMENT

- 1981 – Japanese and American fifth-generation computer projects
- 1984 – V.V. Golenkov defended his PhD thesis "Structural organization and processing of information in electronic mathematical machines controlled by the flow of complex structured data"
- D.A. Pospelov's advice: «Before designing computers focused on the implementation of intelligent computer systems, it is necessary:
 - to develop basic mathematical and software for such computers;
 - to develop the basics of technology for designing intelligent computer systems implemented on the basis of these computers;
 - to develop a software model (emulator) of a new generation computer being created on modern computers;
 - implement several specific *intelligent computer systems* based on the above technology and the specified software model of the future computer.

If all this is not done, then the developed computer of the new generation will be a brilliantly made "hardware", which is unclear how to use and which, therefore, will quickly become obsolete. That is why all fifth-generation computer projects were doomed.»

- 1992 – Prototype of a semantic computer on transputers
- 1995 – Opening of an educational specialty in BSUIR "Artificial intelligence" and the creation of an appropriate graduate department
- 1996 – V.V. Golenkov defended his doctoral dissertation "Graphodynamic models and methods of

parallel asynchronous processing of information in intelligent systems"

- 2010 – Creation of an open *OSTIS Project* aimed at creating an open integrated technology for designing *intelligent computer systems*, the implementation of which focuses on the use of *next-generation computers*
- 2011 – The beginning of the annual OSTIS conferences aimed at the development of the open *OSTIS Project*
- 2019 – On the basis of the educational institution "Belarusian State University of Informatics and Radioelectronics", an educational and scientific association in the direction of "Artificial Intelligence";
- 2021 – Publication of a prototype of the *OSTIS Technology Standard*, presented in the form of a formalized text, which is the source text of the knowledge base of the Meta-system for Supporting the Design of Intelligent computer systems developed using *OSTIS Technology* [11];
- 2023 – Publication of a collective monograph on *OSTIS Technology*, which is considered as the basis for further development and official recognition of the formalized *Standard of OSTIS Technology* and a significant expansion of the corresponding collective of authors

Summarizing our experience in the field of Artificial Intelligence, we can say the following:

- The requirements for *intelligent computer systems* of the next generation (a high level of *self-learning, interoperability, independence, universality*), involves the creation of a *fundamentally new integrated* technology that integrates, ensures compatibility of the entire variety of existing *private Artificial intelligence technologies* and which supports all stages of the life cycle of *intelligent computer systems* being developed;
- The complexity of the implementation of *next-generation intelligent computer systems* (due to the discrepancy between the basic principles of information processing in such systems and the principles of the von Neumann machine underlying modern computers) requires the creation of computers specifically focused on the implementation of *next-generation intelligent computer systems*. But it is necessary to create these new-generation computers on the basis of (or, more precisely, within) the above-mentioned integrated technology for designing and supporting the subsequent stages of the life cycle of *next-generation intelligent computer systems*.
- The epicenter of the creation and subsequent evolution of this complex technology for *next-generation intelligent computer systems* is:
 - training of a new generation of specialists in the field of *Artificial Intelligence*, who are initially

focused on convergence, on ensuring compatibility of their results with the results of their colleagues and on the specification of their results within the Library of standard (reusable) components;

- permanent development of the *OSTIS Technology Standard*, presented in the form of a formalized text of the *knowledge base* of the Meta-system for supporting the design of *intelligent computer systems* developed using *OSTIS Technology*.

VI. FEATURES, ADVANTAGES AND NOVELTY OF OSTIS TECHNOLOGY

The novelty of OSTIS Technology primarily consists of:

- the requirements for systems created and maintained with the help of this Technology (for *next-generation intelligent computer systems*) – hybridity, interoperability, self-learning.
- the requirements for the *OSTIS Technology* itself (for the methods used by it, automated methods and tools) – the complexity of the technology, its versatility and self-learning

Additional factors of the novelty of *OSTIS Technology* are:

- the fact that the intensive evolution of the *OSTIS Technology* itself (the transition to its new versions) does not lead to the moral aging of already operated *intelligent computer systems (ostis-systems)*, since during the operation of these systems, their automatic modification (modernization) is possible in the direction of bringing them into line with the current version of the *OSTIS Technology*;
- the fact that permanent support of semantic compatibility of operated *intelligent computer systems (ostis-systems)* is provided during their own evolution, as well as during the evolution of the *OSTIS Technology* itself;
- the fact that the basis of the activity (functioning) of hierarchical collectives of *ostis-systems* is decentralized planning, initiation and situational management of collectively performed actions (processes) carried out within the framework of both long-term and temporarily existing *collectives of ostis-systems*;
- in a significant increase in the efficiency of expanding and using the Library of standard (reusable) components of *ostis-systems (OSTIS Libraries)* due to:
 - exclusion of semantic equivalence of components;
 - a significant reduction in the variety of logically and functionally equivalent components;
 - the presence of a simple and fairly easily automated procedure for integrating the components of the specified library and, accordingly, the

procedure for assembling *ostis-systems* from ready-made components of the *OSTIS Library*;

- in the orientation to create an integrated model that ensures the coordination of the entire variety of types and areas of *human activity* and to develop the architecture of a global complex of *ostis-systems* that provides automation of this diversity (*OSTIS Ecosystem*).

VII. CURRENT PROJECTS OF THE CURRENT STAGE OF WORK ON THE DEVELOPMENT OF OSTIS TECHNOLOGY

Let's list some projects of applied *next-generation intelligent computer systems* that are relevant at this stage and the means of their development:

- Development of a formalized ***Standard for next-generation intelligent computer systems***, presented as part of the *knowledge base* of the intellectual portal of scientific and technical knowledge on the theory of *next-generation intelligent computer systems* and providing *semantic compatibility of computer systems* of this class [11].
- Development of a formalized ***Standard of methods and tools for supporting the life cycle of next-generation intelligent computer systems***, presented as part of the *knowledge base* of the intelligent *Metasystem for automating the life cycle of next-generation computer systems (OSTIS Metasystem)*.
- Development of a comprehensive ***Library of standard components of next-generation intelligent computer systems (OSTIS Library)***, providing compatibility of standard (reusable) components and full automation of their integration (connection) in the process of assembly (component) design of semantically compatible *next-generation intelligent computer systems*.
- Within the framework of the ***OSTIS Meta-system***, providing wide access to the current state of the *OSTIS Standard* and developing appropriate semantic visualization and navigation tools.
- Within the framework of the ***OSTIS Metasystem***, the development of automation and management tools for the process of collective improvement (modernization, reengineering) of the *OSTIS Standard*.
- Development of a ***software platform*** for the implementation of *next-generation intelligent computer systems*.
- Development of an ***associative semantic computer*** for the implementation of *next-generation intelligent computer systems*. This is a universal computer in which the hardware implementation of associative

reconfigurable (structurally tunable) memory is carried out, in which information processing is reduced to reconfiguration of connections between memory elements.

- Development of the architecture of the *next-generation intelligent computer system*, which is a **personal intellectual assistant** (secretary, referent) for each user, providing the maximum possible automation of the user interaction process with the entire Global ecosystem of *next-generation intelligent computer systems (OSTIS Ecosystem)*. The *knowledge base* of each such *personal intellectual assistant* includes:
 - personal information of the corresponding user, access to which is provided to other *intelligent computer systems* by the personal intelligent assistant of this user, but necessarily with the permission of this user and with the notification of the relevant risk factors to the user. The user's personal information is his medical data, biographical data, personal photographs, unpublished intellectual property, generated or sent messages addressed to other users or various communities.
 - information about various communities of the *Global Ecosystem of next-generation intelligent computer systems*, of which the corresponding (assisted) user is a member, indicating the role (position, duties) that the specified user performs within each such community. There can be many of these communities — professional communities, friends, relatives, consumer-producer communities, administrative and civil communities, banks, medical service communities, etc.
 - information about your own plans and intentions (both strategic and immediate, including meetings, negotiations, meetings)

Personal intelligent assistant *problem solver*

- provides the maximum possible automation of various types of professional individual activities of the corresponding (serviced) user;
- provides intellectual mediation (representation of interests) of the served user within all communities of which he is a member.

User interface of a personal intelligent assistant

- provides the user with the means to manage his individual activities carried out collectively with the corresponding personal intellectual assistant;
- provides a unified nature of user interaction within the various communities to which it belongs. The simplest type of community is a one-time dialogue between two users.
- Development of a unified **set of automation tools for individual design of fragments of knowledge bases**, which is part of each user's *personal intel-*

lectual assistant and provides support for individual contributions to the development of both their own (personal) *knowledge base* and the *knowledge base* of other systems that are part of the *Ecosystem of intelligent computer systems*. The specified complex of automation tools includes

- editor of the internal representation of knowledge (editor of *sc-texts*);
- editors of various external forms of knowledge representation (*sc.g-texts*, *sc.n-texts*);
- translators from the internal representation of knowledge to various external forms of representation;
- translators from each form of external representation of knowledge to their internal representation;
- means of syntactic and semantic analysis of the projected fragment of the *knowledge base*;
- a translator that provides the transformation of the internal representation of knowledge (in *SC-code*) into a natural language representation in the format of the LaTeX markup language that meets the requirements for the design of articles in collections of scientific and technical materials. This translator will allow to concentrate the efforts of developers of various *intelligent computer systems* on the formalization of scientific and technical knowledge used in intelligent computer systems, and significantly reduce the complexity of the preparation and registration of publications of relevant scientific and technical results. In the future, various scientific and technical journals should be transformed into intellectual portals of collectively developed scientific and technical knowledge in various fields.
- Development of a set of *textbf tools for individual comprehensive permanent medical control and monitoring* of the corresponding (serviced) user within the framework of a personal intelligent assistant
- Development for each community of *next-generation intelligent computer systems* of a unified set of **tools for the collective development of a common knowledge base** of this community (*knowledge base* of the corporate system of the specified community), which includes:
 - means of assembly (integration) of the developed *knowledge base* from its individually developed fragments;
 - means of coordination of individually developed fragments (personal points of view, the epicenter of which is the coordination of the *concepts* used);
 - means of mutual reviewing;
 - means of coordinated adjustment of the *knowledge base*;
 - means of forming and agreeing on a plan for

- improving a collectively developed *knowledge base*;
- means of monitoring and managing the process of improving the collectively developed *knowledge base*.
- Expansion of the set of **design automation tools** for various types of components of *next-generation intelligent computer systems (ostis-systems)* and various classes of such systems.
- Development of the formal structure of the global complex of automated human activity and the corresponding architecture of the **OSTIS Ecosystem**. A significant expansion of the areas of application of *OSTIS Technology* (medicine, industry, construction, law, and so on).
- Development within the framework of the *OSTIS Ecosystem* of a **set of tools and techniques for training specialists in the field of Artificial intelligence** (at the level of students, undergraduates and postgraduates).
- Development within the framework of the *OSTIS Ecosystem* of a **complex of means of informatization of secondary education** with the help of semantically compatible *next-generation intelligent computer systems*.
- Development within the framework of the *OSTIS Ecosystem* of a **complex of informatization tools for higher technical education** using semantically compatible *next-generation intelligent computer systems*.

ACKNOWLEDGMENT

The authors would like to thank the scientific collectives of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help in the work and valuable comments.

REFERENCES

- [1] A. V. Palagin, "Problemy transdisciplinarnosti i rol' informatiki[problems of transdisciplinarity and the role of informatics]," *Cybernetics and System Analysis*, no. 5, pp. 3–13, 2013.
- [2] A. Kleshchev, V. Gribova, D. Krylov, F. Moskalenko, S. Smagin, V. Timchenko, M. Tyutyunnik, and E. Shalfeeva, "Oblachnaya platforma dlya razrabotki i upravleniya intellektual'nimi sistemami[cloud platform for development and management of intelligent systems]," in *Proceedings of the 1 International Scientific and Technical Conference, 10-12 February, Minsk, 2011*, pp. 5–14.
- [3] V. Golenkov, N. Gulyakina, I. Davydenko, and D. Shunkevich, "Semanticheskie tehnologii proektirovaniya intellektual'nykh sistem i semanticheskie associativnye komputery[semantic technologies of intelligent systems design and semantic associative computers]," *Reports of Belarusian State University of Informatics and Radioelectronics*. 2019, no. 3, pp. 42–50, 2019.
- [4] G. Shchedrovitsy, "Ishodnie predstavleniya i categorizacionnie sredstva teorii deyatelnosti[initial representations and categorization means of activity theory]," *Selected works*, pp. 232–298, 1995.

- [5] I. Lukashevich, R. Machinskaya, and M. Fishman, "Struktural'naya organizatsiya medicinskoj informatsii "eeg-expert"[structural organization of medical information "eeg-expert]," *Mathematical methods in engineering and technology" MMTT-2000*, 2000.
- [6] N. Borgest, "Startegii intellekta i ego ontologii: popitka razobrat'sya[strategies of intellect and its ontology: an attempt to understand]," *Design Ontology*, no. 9, pp. 407–425, 2019.
- [7] S. Bhatt, C. Zhao, A. Seth, and V. Shalin, "Graph znaniy kak sredstvo ulucheniya ii[knowledge graphs as a means to improve ai]," *Open Systems*, 2020.
- [8] V. Tarasov, *Ot mnogoagentnykh sistem k intellektual'nykh organizatsiyam [From multi-agent systems to intelligent organizations]*. M.: Editorial URSS, 2002, (in Russian).
- [9] R. Gataullin, A. Gatiatullin, R. Gilmullin, O. Nevzorova, D. Mukhamedshin, D. Suleimanov, B. Khakimov, and A. Khusainov, *Formal'nye modeli i programnye instrumenty komp'yuternoy obrabotki tatarskogo yazika: Nauchnoe izdanie [Formal models and software tools for computer processing of Tatar language: Scientific publication]*. Kazan: Publishing house of the Academy of Sciences of the Republic of Tatarstan: Academy of Sciences of the Republic of Tatarstan, Institute of Applied Semiotics., 2019.
- [10] D. Muromtsev, D. Volchek, and A. Romanov, "Industrial'nye graphy znaniy - intellektual'noe yadro cifrovoj ekonomiki[industrial knowledge graphs - the intellectual core of the digital economy]," *Control Engineering Russia*, 2019.
- [11] V. V. Golenkov, N. A. Gulyakina, and D. V. Shunkevich, *Otkrytaya tehnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheskoi sovmestimix gibridnix intellektual'nykh komp'yuternykh sistem[Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*. Minsk: Bestprint, 2021.

Основные направления, проблемы и перспективы развития интеллектуальных компьютерных систем нового поколения и соответствующей им технологии

Голенков В. В., Гулякина Н. А.

В статье рассмотрены основные направления, проблемы и перспективы развития интеллектуальных компьютерных систем нового поколения и соответствующих технологий с акцентом на современное состояние работ в области Искусственного интеллекта. Освещены проблемы и методологические вызовы современного этапа развития, а также предпосылки перехода к интеллектуальным компьютерным системам нового поколения. В тексте также представлены предыстория и история развития Технологии OSTIS, а также описаны ее особенности, преимущества и новизна. Дополнительно упоминаются текущие проекты, связанные с развитием Технологии OSTIS на текущем этапе работы.

Received 13.03.2023

Factor of Digital Culture in the Application of Artificial Intelligence in Economics and Education

Boris Panshin

*Department of Digital economic of Economic faculty
Belarusian State University*

Minsk, Belarus

Email: panshin@bsu.by

Abstract—The purpose of the article is to attract the attention of young researchers of research on the problems of digital culture as one of the most important factors in the successful development and implementation of artificial intelligence technologies. It is shown that the relevance of the development and application of AI technologies is due to the growing complexity of the objectively necessary tasks of economic management and the ever-increasing pace and scale of digitalization. The importance of the digital culture factor in the formation of a digital environment comfortable for life and interaction is noted, which determines the effectiveness of synergistic processes of self-assembly and self-organization of complex dynamic systems, which are modern society and economy in the context of global digitalization. The role of digital culture is analyzed as a science about the relationship of people to each other in the digital environment and the environment itself with the outside world and as an institution for achieving excellence in the creation and application of digital technologies and artificial intelligence technologies.

Keywords—artificial intelligence, digital culture, synergistic effects in the economy and society, digital culture in the economy and education, digital culture of the enterprise.

I. INTRODUCTION

Currently, artificial intelligence (AI) is entering a new phase of development and is increasingly becoming one of the main catalysts for change in the economy and education.

At the same time, decisions on how to use this technology, to balance risks and opportunities, are made primarily by large corporations, relegating to the background research on assessing the risks of developing and implementing applications with AI technologies. As a result, the development of AI is very contradictory and zigzag.

If in 2022 32 significant industrial machine learning models were created, then only 3 by scientific centers. There is a tendency to reduce the departments dealing with ethics and security issues in corporations like Microsoft or Google. This is mainly due to the fact that AI technology has begun to require more and more resources: personnel, information (databases) and

computing power necessary to create such applications. At the same time, there is a growing interest in the regulation of AI from the public administration: an analysis of the situation in 127 countries showed that the number of laws adopted in different countries containing the phrase “artificial intelligence” has grown from 2nd in 2016 to 37th in 2022 [1].

It is obvious that with the growing concerns about the impact of AI on the labor market, the regulation of the use of AI will be improved in various directions, which requires the development of an appropriate methodological framework for assessing the risks of AI and choosing directions for the development and application of applications with AI technologies. In terms of areas, in our opinion, these are economics and education. In terms of methodology, this is the application of the principles of synergetics to the construction of qualitative models (phase portraits) of economic development under the influence of the growth in the use of AI in various fields of activity with an assessment of the impact of the level of human development on the consistency and success of society development.

II. THE OBJECTIVE NECESSITY OF AI

According to the conclusions made by academician V.M. Glushkov [2] in the early 70s of the last century, the complexity of objectively necessary management tasks is growing faster than the square of the number of people employed in the economy of people. Since, as a result of the constant development of technologies and organizations, the complication and deepening of specialization and cooperation between economic entities, new connections arise, and hence new management tasks. At the same time, the number and complexity of emerging tasks and the number of people employed in the economy have a non-linear relationship, which makes it difficult to build integrated planning and economic management systems in the context of digitalization: building a digital state plan and a digital strategic management system.

Appropriate training and retraining of personnel is of key importance. If in recent centuries mental abilities were more important than emotional skills and the ability to work with hands, then now, there is a reversal of the trend — emotional (social) skills, such as empathy, the ability to build relationships and persuade, come to the fore. At the same time, the balance between cognitive and social skills will change significantly even in traditionally intellectual professions, which necessitates appropriate changes in the education system related to the development of emotional intelligence among schoolchildren and students and its adaptation to activities in the digital environment. Emotional intelligence is considered as important as mental ability (IQ) because it helps to establish teamwork and achieve synergies in production and management while reducing the number of employees. With regard to AI, it is assumed that intelligent digital technologies will replace people in routine tasks, and people will be successful in activities that require good social skills (soft skills) and interdisciplinary experience.

III. SYNERGETICS IS A "GATHERING POINT" OF OPINIONS ON THE DIRECTIONS OF DIGITALIZATION AND THE APPLICATION OF ARTIFICIAL INTELLIGENCE TECHNOLOGIES

The main goal of numerous theories of digital transformation and AI is an attempt to build certain models to predict the development directions of this process in order to increase the efficiency of the economy and the sustainable development of society [3].

In the last two decades, in the analysis of various problems, the principles of synergetics are increasingly used — the science of the processes of development and self-organization of complex systems, which, undoubtedly, is any modern enterprise. The main thing in synergetics is the self-organization of the components of complex systems when a certain variety of elements and relationships between them are reached, and the presence of a certain degree of culture of production participants.

Self-assembly refers to the process of combining system components into horizontal structures. And self-organization is the emergence of qualitatively new structures (bifurcation) as a result of multiple interactions of components of lower hierarchical levels in order to form a production environment that is comfortable for interaction — convenient, fast, with minimal barriers.

In practice, various irrational phenomena in the course of digital transformation are encouraged to turn to synergetics, which are not always amenable to clear definitions and explanations, but are definitely determined by the level of organizational and digital culture of the employees of the enterprise.

In a more simplified cybernetic representation, an enterprise is a program that ideally works according to self-assembly and self-organization algorithms through

the dynamic formation of a cultural environment and cultural code (skills, abilities, traditions, values, ethics, aesthetics). This program "works" in the direction of ever faster, more diverse and simplified interaction of the elements of the enterprise to create higher forms of organization in accordance with the laws of natural harmony.

With this approach to considering the digital transformation of an enterprise, it becomes possible to analyze production and get an explanation of the reasons and forecasts for the development of modern digitized enterprises.

It is important to note the interdisciplinary nature of the synergetic method, which requires the joint efforts of scientists and specialists from various fields — from philosophers, artists, mathematicians, engineers to managers and system analysts.

It can be assumed that both the industrial revolution in the past and the modern information and digital revolution are the result of the adaptation of people and industries to new technologies. In turn, the problem of creating an information and digital culture is to form the environment and people's skills (digital competencies) as a condition for self-assembly and self-organization. The understanding of technologies should turn into a desire to use them effectively.

At the same time, as practice shows, the main engine of self-assembly and self-organization is artificial intelligence (AI), on the level of development and application of which economic growth and quality of life depend.

IV. PRINCIPLES OF SYNERGETICS IN THE DEVELOPMENT OF THE ECONOMY

The structure of the economic system and society as a whole is determined by the nature of the interaction between its elements. From the point of view of synergetics, the goal of the economy as a subsystem of society is the self-organization of producers and consumers of goods and services, which works according to the pricing model "goods — money — goods". The current market economy is focused on the concentration of ownership, which leads to the division of players into active and passive ones. In the time of Adam Smith, the economy was a market economy as long as producers and consumers strictly followed the postulates of the Protestant ethic [4]. Over time, the ethical culture of self-organization of the economy under the influence of the imperative of profit was increasingly eroded, leading to an increase in inequality in society, which was offset by an increase in consumption and a decrease in the birth rate in the most civilized countries.

In a planned economy, self-organization was hampered by excessive administrative regulation of the forms of interaction between economic agents and prices for products and services. Therefore, it is necessary to move to

a new paradigm — a digital sharing economy, which will give a new impetus to self-organization due to the greater information content of consumers and producers and the growth of opportunities for their cooperation in the production of goods and services. That is, cultural self-organization is required first, and then economic self-organization takes place.

The digital economy of the future is an inclusive, solidarity and sharing economy, which means the maximum involvement of the population in production, the distribution of income in accordance with collective interests, the maximum efficiency in the use of the country's resources through cooperation between enterprises and individuals based on trust and transferring responsibility for work to lower levels of decision making. As a result, there is more space for the emergence and development of new ways of entrepreneurship and cooperation between producers of goods and services and taking advantage of the digital environment.

V. FEATURES OF THE DIGITAL ENVIRONMENT

The digital environment is an integral part of the natural and virtual worlds surrounding a person and becomes as significant as the natural world. The speed and extent of change is critical. The forecast of unregulated development of the digital environment is not optimistic. It is not those who provide quality content that win, but those who quickly gain a critical mass of consumers. The intelligence level of data processing systems (“digital footprints”) is growing, but trust in the system is decreasing. Deepfakes and chatbots increase the entropy of the environment, which goes into the turbulence stage. The environment is no longer conducive to the production and perception of new information, ceases to be useful for development and increases cognitive degradation. The pattern of thinking and the pattern of behavior are changing. New synergies are emerging, and the challenge is to predict bifurcation points and define digital development trajectories. The maximum effort is to determine what measures should be taken. To do this, it is necessary to define some research framework and methodology. For a qualitative assessment of development, synergetics is most applicable, among the priority measures is raising the level of digital culture [5].

The four new laws of robotics formulated in Frank Pasquale's book also confirm the growing importance of culture in the creation and application of AI systems [6]:

- 1) Robotic systems and AI should complement professionals, not replace them.
- 2) Robotic systems and AI should not pretend to be people.
- 3) Robotic systems and AI should not fuel a zero-sum arms race.
- 4) Robotic systems and AI must always contain an indication of the identity of its creator (or creators), operator (or operators) and owner (or owners).

VI. DIGITAL CULTURE

Digital culture is the science of the relationship of people to each other in the digital environment and the environment itself with the outside world. In the most general sense, digital culture can be viewed, on the one hand, as an institution for achieving excellence in the creation and application of digital technologies, on the other hand, as a set of practices for regulating the behavior of people and communities in the digital environment. The methodology for creating an environment with such characteristics is based on the synergistic principles of self-assembly and self-organization of complex dynamic systems, such as modern society and the economy in the context of global digitalization. Synergetics makes it possible to connect the humanities and natural sciences and gives an understanding that we live in a highly non-linear world, that social systems are historical and depend on their “trajectory” in the past [7].

The phenomenon of digital culture, due to its complexity, should be considered at three levels: a person, an enterprise (community) and society as a whole:

- 1) With regard to an individual, the essence of culture is the development of imaginative thinking, which allows you to create an ethical coordinate system for life in a digital environment. Culture creates appreciation and self-esteem of the individual in digital interactions. Digital culture is the ability to understand the patterns of development of digital systems, which gives a person additional vitality to solve complex problems and determine their role in shaping the digital environment. It is conscious activity in the digital environment that gives rise to digital culture.
- 2) In relation to the enterprise, digital culture is what employees do, what they believe in and how they behave over time, that is, it is the attitudes, behaviors and habits associated with digital technologies that employees repeat over time.

For an enterprise, digital culture is to some extent a task, after completing which one can begin to solve the technical and organizational problems of introducing new technologies into production and management. That is, understanding that the digital environment predetermines both the appropriate organizational structure of the business and the behavior of the employee in terms of his competencies and values.

Cultural costs have a strong impact on the development of the traditional economy and have even greater consequences in the digital economy. The more complex the technology, the higher the requirements for qualification and quality of interactions. Studies show that over 30% of the obstacles to successful digital transformation of enterprises are due to the cultural and behavioral problems of employees and the unwillingness of managers to communicate effectively in the digital environment.

Therefore, the meaning of a developed digital culture for a modern enterprise is that everyone in the team listens to everyone and everyone, and everyone listens to everyone, which leads to the preparation of conditions for the development and implementation of AI technologies, as the most advanced digital technologies.

The effects that digitalization and the use of digital technologies give depend on thousands of small deeds of managers and personnel of enterprises that must be correctly and consistently performed, which requires appropriate information and cultural self-organization of enterprise personnel. This requires the integration of professional and cultural knowledge and skills, which requires the study and understanding of digital culture — a new direction in the theory and practice of digital transformation and the use of artificial intelligence systems.

For this reason, the global giants of the digital industry Google, Alibaba and others have long been using methods and algorithms to determine whether employees meet the specific requirements of the digital culture of modern "smart" companies (risk management, cooperation, the ability to act independently in flexible organizational structures based on horizontal connections and network coordination, decision-making based on intelligent data processing, personal participation in decision-making, etc.).

From the point of view of the development of society, digital culture can be considered as a humanitarian resource of digitalization, as an institution for achieving excellence in the creation and application of digital technologies in order to create a digital environment as comfortable as possible for interaction.

Digital culture is the understanding that digital transformation is primarily a social phenomenon that comes from new forms of communication and interaction between people through social networks, digital platforms and technologies. And, of course, for this, the digital environment must be "aesthetically charged", that is, it must be formed according to the laws of aesthetics and rationalization, minimizing the risks and threats of network communications. The main place of digital culture, as well as traditional culture, is the inner world of a person and the world of the community, for the maintenance of which constant self-improvement and the expenditure of human energy are needed, work on oneself and the formation of appropriate institutions to regulate behavior and interactions in the digital environment.

VII. DIGITAL CULTURE IS THE MOST IMPORTANT FACTOR IN THE SUCCESS OF DIGITAL TRANSFORMATION

A 2016 McKinsey study showed that over 30% of the barriers to successful digital transformation are due to

cultural and behavioral issues in enterprise employees. The digital economy is an ultra-low cost economy. But subject to the necessary level of digital culture. Manufacturing has one of the highest human error rates of any industry. At the same time, up to 70% of these errors are the result of imperfect organizational culture [8].

The specificity of digital culture is determined by the digital economy, which involves network coordination of the interaction of business processes and intelligent data processing, and therefore requires certain knowledge, skills and value orientations. First of all, the readiness to work in conditions of weakly hierarchical flexible systems of independent decision-making and understanding the increasing complexity of digital ecosystems as they function and develop.

From the point of view of systems theory, complex historically developing organic wholes (systems, which are modern large enterprises) must contain special information and organizational structures within themselves that ensure the management of the system and its self-regulation. These structures are represented by links and codes, in accordance with which the organization of the system as a whole is reproduced and the features of its main reactions to the external environment are recreated.

In biological organisms, this role is played by genetic codes (DNA, RNA). In society, as an integral social organism, culture acts as an analogue of genetic codes. In digitized enterprises, it is organizational and digital culture.

Therefore, the success of digital transformation is, to a greater extent, the result of a control action that comes "from within" the system (from the cultural predisposition to changes in employees, and not just from the outside), by creating conditions for maximizing the use of the skills and enthusiasm of the personnel of enterprises (the energy of human culture), which makes it possible to achieve synergistic effects from the interaction of employees (self-assembly and self-organization of individuals and communities) in the production of demanded goods and services.

For this reason, digital culture can be seen as an institution for achieving excellence in the creation and application of digital technologies in order to create a digital environment as comfortable as possible for interaction, self-assembly and self-organization of enterprises and individuals.

VIII. DIGITAL CULTURE AND AI TECHNOLOGIES IN EDUCATION

From the point of view of teaching, it is important, in our opinion, to note that digital culture is still science X, that is, not fully defined and with a pronounced interdisciplinary character. This is a kind of philosophy for the development of the digital world, and for various specializations, training in digital culture should be

carried out approximately sequentially, starting from the culture of dialogue and academic writing and ending with the cultural aspects of creating and applying artificial intelligence systems. At the same time, this approach makes it possible to combine the efforts of all faculties in research and the preparation of practical methods for digital culture. And it will enrich students with an understanding of the complexity, diversity and inconsistency of the development of the digital world. There is a place for everyone, both physicists and lyricists.

The theoretical model of culture is a kind of coordinate system, a system of key concepts that reveal the essence of culture. For the theory of digital culture, it is important to understand the differences between material culture and digital culture. Digital culture is formed through terms, operations, standards.

Education is a system of production of human capital, which is one of the most complex subsystems of society. Various studies show that humans can learn new things from AI systems and communicate them to other humans in a way that could potentially impact broader human culture. Algorithms using AI technologies are having an increasing impact on human culture, which requires understanding how they interact with us or with each other. Since, algorithms and AI are not simple means of cultural transmission (such as books or the Internet) and can also play an active role in shaping the processes of cultural evolution on the Internet, where people and algorithms regularly interact.

The highest priority is to improve the quality of content, the culture of dialogue and academic writing. This is a fundamental problem, literally and figuratively. Everything else depends on the quality of the primary data. This is the raw material for neural networks and decision-making systems (what we “feed” the neural networks will be the output). What is needed: an ordered object world and an ethically charged digital environment and a definition of who and how the value base will be created.

The problem with the spread of AI on a broader basis is trust - trust in the data that enters the systems and the decisions made by those systems. In modern societies, the most important condition for the development of artificial intelligence technologies is digital culture. A significant and growing proportion of AI algorithms operate online, both as bots that interact with the user and as sorting and guiding algorithms that mediate network interactions.

According to research, in 2020, up to 20% of companies' profits were generated thanks to artificial intelligence. At the same time, there were a number of obstacles to the successful use of AI, due to the influence of digital culture:

- 1) The quality of the preparation of primary data for systems for developing and making decisions,

including with the use of AI technologies.

- 2) Insufficient consideration of the peculiarities of thinking and values of AI developers, which affects the ethical attitudes of artificial intelligence (understanding the humanitarian context of applications with AI).
- 3) The lack of a strategy in the organization of labor for the collection and filtering of primary data and their processing on various platforms.

The main reason for the new possibilities of artificial intelligence is the integration of its applications with modern digital technology, Internet technologies and big data.

The emergence of text generators based on AI technologies has opened a new era in global digitalization, creating both additional opportunities in data processing and analysis, and producing new risks and threats, primarily in the education system. The use of AI has led to the fact that human thinking has actually become combined: both the brain and the computer are involved in thinking, and it is required to optimize their interaction. The main risks are that there is a psychological danger of increasing dependence on a computer and a decrease in the level of mastery of knowledge due to a change in the technology of transferring knowledge: previously it was a teacher or an expert community who are in dialogue with the student. The use of AI reduces the learning format to a monologue.

It is important to note that learning is a process that results in the ability to recognize signs of new information according to a certain algorithm and then develop decision rules for developing managerial decisions and actions. Thinking models have a different format and require a different level of attention (the ability to identify signs of new information and give priority to one or another sign [9]).

The inevitable emergence of ChatGPT is an objectively necessary response to the critically increased volumes and dynamics of data updates in the digital environment. At the same time, this is the next stage in the development of systems for systematizing and concentrating data on the way of their transformation into information and knowledge (books, libraries, catalogs, websites, search engines, etc.), that is, the formalization of knowledge to turn it into a resource for analysis and management decisions. . As one of the first Soviet cybernetics N. G. Zaitsev — data is not yet a resource, but a sign of a resource, and in order to turn data into a resource, an appropriate level of researcher training is required to recognize in these data what is necessary for development and interaction.

It can be assumed that the subsequent development of intelligent systems such as GPT will be carried out in the direction of greater specialization of content generators (texts, images, sounds) in areas of knowledge and an

increase in the level of their functionality in relation to specific types of activities. That is, it will be a process similar to the cataloging of various sources in specific thematic areas. This is an objective process, since information is a quantitative measure of order in a certain system. If there is no order, then entropy increases, which leads to degradation (aging) of the system.

At the same time, with the growing use of GPT-type generators for preparing information blocks, the problem of trust in information generated by AI arises. Especially in those cases when the content and configuration of the information block corresponds to traditional ideas and therefore is not subject to subsequent verification. As a result, the level of trust in information products obtained with the help of AI technologies is reduced. Moreover, the growth in the volume of products generated by chatbots will lead, in the shortest possible time, to the formation of a second-hand data layer in the global network, that is, containing only the data on which the chatbot was trained.

With regard to the economy and education, a decrease in the credibility of data from the Internet leads to the degradation of information and digital systems. Efforts to form a digital culture of users are aimed at preventing the negative trend of reducing trust in data on the network. At the same time, it is not, first of all, about changing the way people think, but starting with changing the way people behave and what they do.

The scope and integration of digital systems into the economy and society as a whole is so large-scale that it requires special attention and an integrated approach to raising the level of digital culture at all levels [10].

Digital culture can be viewed as an institution for achieving excellence in the creation and application of digital technologies in order to create a digital environment as comfortable as possible for interaction, self-assembly and self-organization of enterprises and individuals within a holistic digital ecosystem [11].

IX. CONCLUSIONS

Shaping a digital culture in the economy and society is a full-time job. Culture is a resource that cannot be acquired for any amount of money. Culture can only be developed through interaction. This is a blessing transmitted in the form of knowledge, traditions, norms, customs, rules. The sensitivity of employees of enterprises and the population to digital culture should be developed. The challenge of spreading AI on a broader basis is trust in the data that enters the systems and the decisions made by those systems. In modern societies, the most important condition for the development of artificial intelligence technologies is digital culture.

REFERENCES

- [1] The state of AI in 2022—and a half decade in review. Available at: <http://www.mckinsey.com> (accessed 09.09.2022).

- [2] V. M. Glushkov Introduction to ACS. Ed. 2nd, corrected and supplemented. Tekhnzha, 1974, P. 320
- [3] V. V. Ivanov, G. G. Malinetsky Digital Economy: Myths, Reality, Perspective. *Russian Academy of Sciences*, 2017, pp. 36–37.
- [4] V. S. Dashkevich Great cultural savagery. Russian Chess House, 2018, P. 720.
- [5] D. Moshella A guide to the digital future. *Alpina Digital*, 2018
- [6] F. Pasquale Novye zakony robototekhniki. Apologiya chelovecheskikh znaniy v epokhu iskusstvennogo intellekta [New laws of robotics. Apology of human knowledge in the age of artificial intelligence], Moscow, Delo, 2022, P. 448.
- [7] Digital Culture: Fears, Hopes And Uncertainty About Artificial Intelligence In Sci-Fi Movies. Available at: <https://www.researchgate.net> (accessed 15.02.2023).
- [8] Digital Culture & Capabilities — McKinsey. Available at: <https://www.mckinsey.com> (accessed 17.02.2023).
- [9] D. S. Chernavsky Sinergetika i informatsiya: Dinamicheskaya teoriya informatsii [Synergetics and Information: Dynamic Information Theory], 2021, Stereotip [stereotype], No. 13, P. 304.
- [10] N. Forsgren, J. Humble, G. Kim Speed up! DevOps Science: How to Build and Scale High-Performance Digital Organizations, Alpina PRO, 2022, P. 216.
- [11] Man of work in the digital economy: new realities and social challenges. V. N. Minina, R. V. Karapetyan, O. V. Veredyuk (eds.). St. Petersburg: Mzd-vo St. Petersburg University, 2021, P. 284.

Фактор цифровой культуры в применении искусственного интеллекта в экономике и образовании

Паньшин Б. Н.

Цель статьи состоит в привлечении внимания молодых исследователей исследований по проблемам цифровой культуры как одного из наиболее важных факторов успешного освоения и внедрения технологий искусственного интеллекта в интегрированных цифровых экосистемах. Показано, что вследствие самонастройки интеллектуальных систем происходит замещение механизмов экономического и административного принуждения механизмами самоконтроля и саморегуляции персонала и всей производственной системы, что обуславливает актуальность разработки приложений с применением ИИ как составной части единой цифровой экосистемы и предъявляет новые требования к системе образования.

Отмечена значимость фактора цифровой культуры в формировании цифровой среды комфортной для жизнедеятельности и взаимодействия, что обуславливает эффективность синергетических процессов самосборки и самоорганизации сложных динамических систем, каковыми являются современное общество и экономика в условиях глобальной цифровизации. Анализируется роль цифровой культуры как науки об отношении людей между собой в цифровой среде и самой среды с окружающим миром и как института для достижения совершенства в создании и применении цифровых технологий и технологий искусственного интеллекта.

Received 13.03.2023

Conceptual Design of Complex Integrated Systems

Victar Kochyn
Belarusian State University
Minsk, Belarus
Email: kochyn@bsu.by

Alexander Kurbatsky
Belarusian State University
Minsk, Belarus
Email: kurbatski@bsu.by

Abstract—The article describes the problems of digital transformation of the Republic of Belarus. The ways of solving the problem are proposed. A new approach to the design and development of complex information systems is proposed – the design of complex integrated systems. The basic properties of complex integrated systems are determined

Keywords—System design, complex systems, digital transformation, system of systems, interoperability, integration

I. INTRODUCTION

Currently, it is difficult to overestimate the importance of digitalization of key processes in any state. Digital technologies make it possible to optimize many management processes in the economy, healthcare, education, and industry. The development of the modern economy is largely based on the processes of digital transformation. Until 2020, digitalization was an evolutionary process, but the COVID-19 pandemic radically changed the role and perception of digitalization in the state and society and accelerated its pace. Digital technologies are now essential for work, learning, entertainment, communication, shopping, and access to everything from health services to culture. One of the important conditions for successful implementation of the digital transformation strategy is the development of new approaches to the design and development of information systems. As such an approach, it is proposed to use the design of complex integrated systems [1], [2].

A. Digital transformation of the Republic of Belarus

Despite the undoubted successes in the development of the information and communication infrastructure of the Republic of Belarus, the creation of individual elements of e-government, it is premature to talk about significant progress in the digitalization of the public sector of Belarus for a number of reasons:

- Many platforms and systems were originally developed to solve specific tasks and did not provide for the possibility and necessity of integration, as well as integration into the chain of industry, country and supranational platforms. Often the developed solutions were not integrated with each other.

- Digitalization in our country has developed chaotically and sometimes uncontrollably from the point of view of embedding in a single strategy of digital transformation of the country.
- Currently, there are practically no industry platforms into which digital platforms of enterprises can be integrated.
- Many institutions and departments use proprietary software to digitalize key processes. Often, when using such software, enterprises have to adapt their business processes to the functionality imposed by the manufacturer, and not vice versa.
- Comprehensive information security is not fully ensured. This problem includes both the development of software and hardware solutions and security in the information space.

The digital transformation of the economy and society of Belarus should contribute to the achievement of the following goals:

- Ensuring the digital sovereignty of the country.
- Creating conditions for the introduction of innovative solutions in the sphere of economy and society, as well as for the integration processes of both internal and external country digital platforms
- Implementation of the import substitution strategy in the field of digitalization of key processes of the economy and society.
- Creation of complex information security systems.
- Creating conditions and guidelines for young people.

One of the significant factors of ensuring the sovereignty of the state in cyberspace is the desire for independence (sovereignty) ICT or more broadly digital sovereignty. Securing digital sovereignty is becoming increasingly difficult in a globalized world. At the same time, there is currently no clear definition of the digital sovereignty of the state. The author Ashmanov I.S. defines digital sovereignty as the right of a state to determine its information policy independently, manage infrastructure, resources, ensure information security, etc. From the point of view of staffing digital sovereignty, this process involves high-quality personnel rotation (the arrival of

responsible specialists in the relevant ministries who are thoroughly versed in the processes of digitalization and IT industries), the creation of educational programs at universities that train multidisciplinary specialists – at the junction of IT technologies and public administration, public policy, innovative economy, the creation of new jobs in the country, providing the state with useful innovations in the field of artificial intelligence, e-government, the Internet of things, electronic services, new weapons systems, etc. From a technological point of view, digital sovereignty is determined by the presence of a sovereign complex of integrated and complementary digital services and platforms in all key spheres of the life of the state and society, including its own hardware base, technological solutions in the field of content delivery, as well as national digital platforms (social networks, cloud storage, messengers, information storage services, etc.). Thus, the digital sovereignty of a country is closely linked to the ability to independently form an information policy, manage information flows, ensure information security, and ensure the storage and processing of digital data regardless of external influence. Achieving these goals requires increased expertise in the digital sphere. The rules of behavior in the virtual space are being actively discussed now. Probably, partly because of the insufficient expert level, the world community as a whole, and ours in particular, have not made much progress on this issue. Currently, no State has been able to fully achieve digital sovereignty. For example, China, which has one of the most technologically advanced and developed economies in the world, is heavily dependent on a number of Western technologies (microchips, processors, etc.). The USA is a world leader in creating ICT solutions. At the same time, a number of high-tech industries have been transferred to other states.

B. Conditions for the introduction of innovative solutions in the sphere of economy and society

Currently, many countries, including the Republic of Belarus, are striving to create conditions for the introduction of digital innovative solutions in the sphere of economy and society. As a rule, these issues are regulated by various fundamental documents such as the Digital Development Strategy, various state digitalization programs, etc. In the Republic of Belarus, the issues of innovative development are reflected in the Resolution of the Council of Ministers of the Republic of Belarus No. 66 dated February 2, 2021 on the approval of the State Program "Digital Development of Belarus for 2021-2025". This state program was adopted in order to ensure the introduction of information and communication and advanced production technologies in the branches of the national economy and the sphere of life of society. The program provides for the implementation of measures for the introduction of digital innovative solutions in the

sphere of economy and society. But for the successful implementation of innovative solutions in the sphere of economy and society, the following conditions must be met:

- Development and implementation of new approaches, methodologies in the field of design, development, standardization and implementation of industry and digital platforms.
- Training of elite specialists in the field of development and implementation of innovative solutions in the sphere of economy and society.

The analysis of successful examples in the field of digital transformation of the state shows that one of the important conditions for the introduction of innovative solutions in the sphere of economy and society was the development of unified country approaches to the design, development and implementation of innovative solutions. When developing such solutions, it is advisable to use the experience of leading countries, the existing level of digitalization of the country, as well as the conditions and features of the development of the economy and society of the Republic of Belarus.

C. Approaches to system design

The rapid development of global networks in the late 90s - early 00s, primarily the Internet, created the prerequisites for a sharp increase in the needs for various information systems, in fact, the process of their creation began to be massive. This was due to the massive introduction of computer technology in various spheres of government and society, the development of data transmission networks. At the initial stage, digitization of existing documents and automation of individual processes took place. One of the first directions of automation of business processes was the development of information systems for managing individual processes of enterprises, such as automation of accounting, personnel accounting, material values, etc. As information technologies were introduced into production and business processes, the complexity of information systems and services grew. For this reason, approaches to the design and development of information systems have changed. Classical approaches no longer allowed the effective development and implementation of complex systems. New design approaches were required that could take into account the complexity of systems, the possibility of scaling, integration with other systems. A separate scientific and methodological discipline, system engineering, is devoted to the issues of designing complex systems. As the complexity of information systems and services grew, not only approaches to system design evolved, but also processes in the digital sphere. In general, it is possible to identify the main processes of digitalization, which were formed as digital technologies penetrated into various spheres of the economy and society:

- Automation. Currently, there are many definitions of this process: from general conceptual definitions to descriptions of specific processes of an enterprise or organization. For example, in [3], the automation process is defined as "a direction of scientific and technological progress that uses self-regulating technical means and mathematical methods in order to free a person from participating in the processes of obtaining, converting, transmitting and using energy, materials, products or information, or significantly reducing the degree of this participation or the complexity of the operations performed." According to GOST [4], automation is the introduction of automatic means for the implementation of processes; a system of measures aimed at increasing human productivity by replacing part of this labor with the work of machines. It is based on the use of modern computer technology and scientific methods. In [5], automation is described as the first stage on the way to digital transformation, when human labor is replaced by machine labor. Summarizing the considered approaches to the definition of the automation process, it can be concluded that automation is a business or production process that is digitized, while there is no optimization or change in the business or production processes themselves.
- Computerization. The widespread introduction of computer technology is closely connected with the process of computerization. According to GOST [4], computerization is the process of automating any processes in any field of human activity through the use of computers. In [6], it is defined that computerization is the widespread introduction of computers into various spheres of human activity (for example, for the management of technology, transport, energy, etc. production processes). In the encyclopedia [7], computerization is described as a process of expanded introduction of electronic computing technology into all spheres of human activity. Based on the results of the analysis of the presented definitions, it can be concluded that computerization is the process of mass introduction of personal computers for the purpose of full-scale use of automation in production or business processes.
- Informatization. Many authors associate the next stage of development and implementation of digital technologies in the state and society with the process of informatization. At the same time, there are various definitions of this process in the literature. Thus, in [8] informatization is described as an organizational, socio-economic, scientific and technical process that provides conditions for the formation and use of information resources and the implementation of information relations. The Law of the Republic of Belarus "On Information, Informatization and Information Protection" [9] provides the following definition: informatization is an organizational, socio-economic, scientific and technical process that provides conditions for the formation and use of information resources and the implementation of information relations. According to [10], informatization is an organizational, socio-economic, scientific and technical process of creating favorable conditions for meeting information needs, realizing the rights and freedoms of subjects of the information sphere, which is based on the mass application of information systems and technologies in all types of activities of individuals and legal entities. The author [11] gives the following definition: informatization is an unprecedented increase in the speed and quantity of production and dissemination of information, as well as the increased role of information processes, systems and networks using ICT in society. Based on the results of the analysis and generalization, it is possible to define informatization as a scientific and technical process for the creation and implementation of information systems and services in various fields of activity, characterized by the massive penetration of information technologies into all spheres of the economy and society.
- Digital transformation. Currently, many institutions, departments, companies, and industries have developed a digital transformation strategy. However, there are many definitions of this process in the literature. As a rule, definitions of digital transformation are based on the size of the object of digital transformation (institution, industry, country). Thus, in [8], the authors define digital transformation as a manifestation of qualitative, revolutionary changes, consisting not only in individual digital transformations, but in a fundamental change in the structure of the economy, in the transfer of value-added centers to the sphere of building digital resources and end-to-end digital processes. The following definition is given in [12]: digital transformation is the process of introduction of digital technologies by an organization, accompanied by optimization of the control system of the main technological processes. Digital transformation is designed to accelerate sales and business growth or increase the efficiency of organizations that are not purely commercial (for example, universities and other educational institutions). In [13], the authors conclude that digital transformation is simultaneously aimed at improving existing business processes and creating competitive advantages by changing and creating new business processes within the enterprise. Based on the results of the analysis, we will determine that digital transformation is a process of integrated

implementation of information technologies in all spheres of activity, enterprises, institutions, industries, countries, which is characterized by optimization and change of existing business, production and other processes, as well as a high level of integration processes between them.

Information system design methodologies have evolved in stages. The following stages can be distinguished:

- The first stage was associated with the digitization of the main local processes. This stage was characterized by the creation of information systems on a local scale. Various programs, as a rule, digitized a specific organizational process and worked locally on separate computing stations. Instances of the system were run on various workstations. With this approach, digital data was not accumulated in a single data warehouse. These programs did not assume and did not have the possibility of integration with other systems. The methodology of designing such information systems was quite simple. The main functions of the system under development, input and output data, processing algorithms were determined. Based on these data, a system was developed. This approach can be conditionally attributed to the classical methodology on the principle of "bottom-up". This stage is closely related to the automation process in the early stages of development.
- The second stage of the development of the system design methodology is closely related to the development of technologies for organizing local area networks. During this period, systems were created that worked on a client-server architecture, in which the storage and processing of information was carried out on a dedicated server. Client instances of the program were launched on local computing stations. These programs provided an interface for working with the server part of the program, while some of the computing functions could be performed locally. At this stage, only individual business and production processes were subject to automation. Information systems did not involve integration with other systems. As a design methodology at this stage, the classical version of system design based on the "bottom-up" principle was also used. The basic design principle at this stage was that the system being designed is created by summing up individual subsystems. Accordingly, the functionally new system consisted of the functions of the subsystems included in it. This stage is closely related to the processes of automation and computerization.

D. Complex integrated system

A complex integrated system is a system that consists of many elements and has a multi-layered structure both in the software-technical plane and in the organizational-legal one. A multi-layer structure has many integrated relationships both between multiple elements and in a multi-layer structure. As a result, a complex integrated system acquires new functions and capabilities that are not typical for individual elements of the system. A complex integrated system can consist of independent autonomous systems, as well as subsystems and other components. Components of a complex integrated system can be fully integrated into a complex integrated system or partially integrated. Figure 1 shows a complex integrated system.

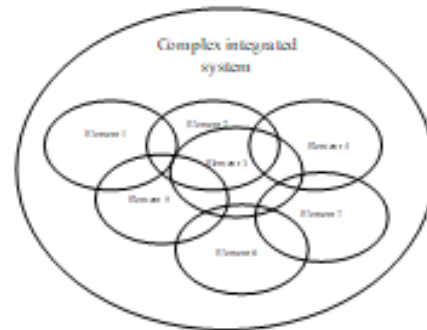


Figure 1. Complex integrated system

In a broad sense, various processes and objects can act as a complex integrated system. Examples include the construction of power plants, the creation of a complex traffic management system, and the design and construction of new city neighborhoods. However, within the framework of this work, the processes of designing and creating information systems of various scales are considered: information systems at the enterprise level, industry, country, and supranational. As such a campaign, it is proposed to consider all the processes of digitalization of the state and society as complex integrated systems. The main focus when developing a methodology for designing complex integrated systems is the integration processes of various systems, processes and services. Complex and integrated systems design is the process of designing and developing complex systems, taking into account the current state of elements, integration processes between elements and functions, as well as identifying new functions and properties obtained as a result of creating a complex integrated system. According to the definition, any sociotechnical system, biological system, etc. can act as a complex integrated system. Within the framework of this work, the object of research is information systems and platforms of various levels, as well as digital transformation processes of various levels. For this reason, research has been con-

ducted in relation to information systems and platforms. A complex integrated system has all the properties of complex systems discussed earlier. With this in mind, any complex system is a special case of a complex integrated system. As noted earlier, a complex integrated system has a number of distinctive properties. The main properties of a complex integrated system are:

- **Integration.** This is a key property of complex integrated systems. Currently, for complex systems, there is a concept of interoperability. According to [14], interoperability is the ability of two or more information systems or components to exchange information and to use information obtained as a result of exchange. The key difference between integration and interoperability is that interaction and integration occurs not only between systems, but also between different elements of a complex integrated system on different layers and planes. Systems and platforms of various levels can act as a design object: from composite subsystems to super-country platforms. And when assigning such systems to complex integrated systems, the main criterion will be the presence of the integration property. Figure 2 shows the set of connections of a complex integrated system.

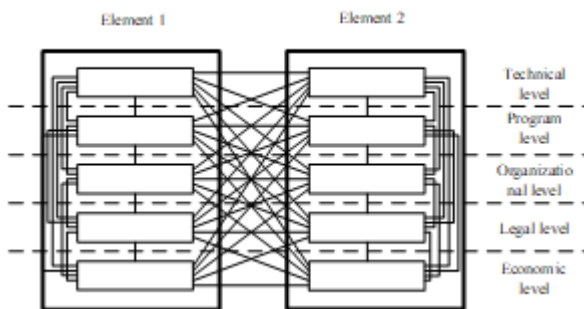


Figure 2. Multiple connections of a complex integrated system

- **Temporality.** In the process of designing complex integrated systems, it is necessary to take into account the current state of individual elements, which was achieved both through the evolutionary development of various processes and systems, and as a result of combining various elements into one system. This property characterizes the degree of development of individual elements of a complex integrated system.
- **Distribution.** The distribution property is inherent in the SoS model discussed earlier. However, when applied to complex integrated systems, this property is more extensive and includes the following types of distribution: geographical, logical, physical, software-technical, and legal.
- **Ownership.** Elements that are part of a complex integrated system can have different owners in the

following areas: owners of individual systems, owners of individual processes.

- **Emergence.** This is a property of a complex integrated system, when when creating a system, new functions and capabilities appear that are not inherent in individual elements of the system
- **Evolutionary development.** Complex integrated systems can acquire new functions and capabilities as a result of evolution, both with external intervention and independently.
- **Complexity.** a system consisting of many interacting components (subsystems), as a result of which a complex system acquires new properties that are not present at the subsystem level and cannot be reduced to the properties of the subsystem level. The complexity of a system is determined by the number of its constituent elements and possible connections between them. The degree of complexity is measured by the diversity of the system. Diversity characterizes the number of possible states of the system.
- **Stress tolerance and self-adaptation to external changes.**

When designing complex integrated systems, the following approaches can be used, which depend on the input parameters of the model:

- **Designing a complex integrated system from scratch.** In this case, the main input parameters will be the requirements for the characteristics and integration relationships of the designed system.
- **Designing a complex integrated system based on existing systems and services.** In this approach, the current state of functioning systems and services is added as input parameters in addition to the requirements for the characteristics and integration relationships of the designed system.

From the point of view of the scale of the object of designing a complex integrated system, the following levels can be distinguished:

- **Subsystem level.** If a complex integrated system is a subsystem, it must have many integration links both within the system of which it is a part and with external systems.
- **Process level.** In the case of designing a complex integrated system at the process level, the object is a dedicated process of an enterprise, company, industry, or country. At the same time, integration with other processes of the designed level should be ensured.
- **The level of an enterprise or company institution.** In this case, as a rule, platforms take into account production and business processes in the industry and include solutions for integration with all digital platforms at the enterprise, institution, department, and company level.

- Country level. As a rule, this level includes various industry and departmental platforms and systems and accumulates all the main information flows and digital data of the designed platform at the country level. At this level, the emergence properties of complex integrated systems are most clearly manifested.
- Super-country level (super-country level). When designing complex integrated systems at the supranational level, the main emphasis is shifted towards data format harmonization, organizational and legal aspects of integration. At this level, complex integrated systems usually have all the basic properties of a complex integrated system.

A complex integrated system includes many different systems and subsystems with varying degrees of integration at different levels.

II. CONCLUSION

Currently, various industry-specific and corporate digital platforms have already been created in the Republic of Belarus, and some elements of e-government have been created. But the level of development of digital platforms in different spheres of the economy and society, and in different regions, can vary greatly. In this regard, it is proposed to develop new approaches that can take into account and integrate into the overall strategy of digital transformation of the state already created and functioning platforms and services, as well as create conditions for further development of industry, country and supranational systems and services as a whole. As such a campaign, it is proposed to consider all the processes of digitalization of the state and society as complex integrated systems. The main focus when developing a methodology for designing complex integrated systems is the integration processes of various systems, processes and services. Applying the proposed approach will allow achieving the goals set in the field of digital transformation of the country.

REFERENCES

- [1] A. N. Kurbatsky, V. P. Kochyn World experience of digital transformation of the state. Key success factors. *Bulletin of Svyaz*, 2022, No. 4, pp. 37–43.
- [2] V. P. Kochyn Designing complex integrated systems as a basis for successful implementation of the digital transformation strategy. *Bulletin of Svyaz*, 2022, No. 5, pp. 41–47.
- [3] Methodological recommendations on the digital transformation of state corporations and companies with state participation. Available at: <https://digital.ac.gov.ru/upload/iblock/044/0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5%20%D1%80%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%86%D0%B8%D0%B8%20%D0%BF%D0%BE%20%D1%86%D1%82.pdf> (accessed 2022)
- [4] GOST 33707-2016 (ISO/IEC 2382:2015) Interstate standard. Information technology. Dictionary. Available at: https://standart.gost.ru/g/%D0%93%D0%9E%D0%A1%D0%A2_33707-2016 (accessed 2022)

- [5] Digital transformation and digital economy: technologies and competencies. Glossary. Available at: https://cto.wiki/%D0%A1%D1%81%D1%8B%D0%BB%D0%BA%D0%B8:%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F_%D0%B8_%D1%86%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D1%8D%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D0%BA%D0%B0:%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B8_%D0%B8_%D0%BA%D0%BE%D0%BC%D0%BF%D0%B5%D1%82%D0%B5%D0%BD%D1%86%D0%B8%D0%B8_%D0%93%D0%BB%D0%BE%D1%81%D1%81%D0%B0%D1%80%D0%B8%D0%B9 (accessed 2022)
- [6] A large encyclopedic dictionary. Available at: <https://dic.academ.ru/dic.nsf/enc3p/161134> (accessed 2022)
- [7] Sociology: Encyclopedia. Available at: <https://sociologyencyclopedia.academic.ru/481/> (accessed 2022)
- [8] Digital transformation. Terms and definitions: STB 2583-2020. Introduction. 2021-03-01. Minsk, Gosstandart, 2020, P. 16.
- [9] Article 1 of the Law of the Republic of Belarus, No. 455-Z of November 10, 2008 "On Information, informatization and information protection"
- [10] Article 2 of the Model Law "On Information, Informatization and Information Security", adopted by Resolution No. 41-15 of the Interparliamentary Assembly of the Commonwealth of Independent States of November 28, 2014).
- [11] IGI Global Available at: <https://www.igi-global.com/dictionary/democratic-governance/35426> (accessed 2022)
- [12] Digital Transformation Available at: https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F (accessed 2023)
- [13] I. M. Zaichenko, P. D. Gorshechnikova, A. I. Levina, A. S. Dubgorn Digital transformation of business: approaches and definition. *Scientific Journal of NIU ITMO. Economics and environmental management series*, 2020, No. 2, pp. 205–212.
- [14] GOST R 55062-2021. Information technologies. Interoperability. Main provisions.

Концептуальное проектирование сложных интегрированных систем

Кочин В. П., Курбацкий А. Н.

В статье описываются проблемы цифровой трансформации Республики Беларусь. Предлагаются пути решения поставленной проблемы. Предложен новый подход к проектированию и разработке сложных информационных систем — проектирование сложных интегрированных систем. Определены основные свойства сложных интегрированных систем.

Received 27.03.2023

Associative Semantic Computers for Intelligent Computer Systems of a New Generation

Vladimir Golenkov and Daniil Shunkevich and Natalia Gulyakina
 Valerian Ivashenko and Vadim Zahariev
Belarusian State University of Informatics and Radioelectronics
 Minsk, Belarus
 Email: {golen, shunkevich, guliakina, ivashenko, zahariev}@bsuir.by

Abstract—The paper considers the shortcomings of the currently dominant von Neumann architecture of computer systems as the basis for building intelligent computer systems of a new generation, analyzes modern approaches to the development of hardware architectures that eliminate some of these shortcomings, substantiates the need to develop fundamentally new hardware architectures, which are hardware version of the implementation of the interpretation platform for systems built on the basis of OSTIS Technology, — associative semantic computers.

Keywords—OSTIS Technology, ostis-platform, platform independence, ontology, associative semantic computer.

I. INTRODUCTION

For the development of *ostis-systems*, the use of modern software and hardware platforms focused on address access to data stored in memory is not always effective, since when developing intelligent systems, it is actually necessary to model nonlinear memory based on linear one. Improving the efficiency of problem solving by intelligent systems requires the development of specialized platforms, including hardware ones, focused on unified semantic models of information representation and processing. Thus, the main purpose of creating *associative semantic computers* is to increase the performance of ostis-systems.

II. CURRENT STATE OF DEVELOPING COMPUTERS FOR INTELLIGENT SYSTEMS

The vast majority of modern software and hardware platforms used in the development of modern computer systems and, in particular, intelligent computer systems are based on the principles of the abstract *von Neumann machine*, or “von Neumann architecture” (see [1], [2]). Let us consider the basic principles underlying the *von Neumann machine*.

von Neumann machine

:= [abstract von Neumann machine]
 ∈ *abstract information processing machine*
 ⇒ *underlying principles**:
 (• [The information in memory is represented as a sequence of strings of characters in a binary alphabet (“0” or “1”).]
 •

[The machine memory is a sequence of addressable memory cells.]

- [Any string of characters in the binary alphabet can be recorded to each cell. At the same time, the length of the lines for all addressable cells is the same (in the current standard of cells, called bytes, it is equal to 8 bits).]
- [Each memory cell uniquely corresponds to a bit string that denotes this cell and represents its address.]
- [To each type of elementary actions (operations) performed in the memory of the von Neumann machine, its identifier is uniquely assigned, which is also represented in memory as a bit string.]
- [Each specific operation (command) performed in memory is represented (specified) in memory as a string consisting of
 - the code of the corresponding type of operation;
 - the sequence of addresses of memory fragments containing operands on which operations are performed – the source arguments and results. Any such fragment is specified by the address of the first byte and the number of bytes. The number of operands is unambiguously set by the operation type code.
]
- [A program running in memory is stored in memory as a sequence of specifications of particular operations (commands).]
- [Thus, both the processed data and the programs for processing this data are stored in the same memory (unlike, for example, the Harvard architecture) and are encoded in the same way.]

Let us consider in more detail the features of the logical

organization of the traditional (von Neumann) architecture of computer systems, which significantly complicate the effective implementation of *ostis-systems* based on it:

- sequential processing that limits the efficiency of computers by the physical capabilities of the element base;
- low level of access to memory, i.e. complexity and awkwardness of performing the procedure of associative search for the necessary fragment of knowledge;
- linear memory organization and an extremely simple view of constructive objects directly stored in memory. This leads to the fact that in intelligent systems built on the basis of modern computers, the manipulation of knowledge is carried out with great difficulty. Firstly, it is necessary to operate not with the structures themselves but with their inconvenient linear representations (lists, adjacency matrices, incidence matrices); secondly, the linearization of complex structures destroys the locality of their transformations;
- the representation of information in the memory of modern computers has a level very far from the semantic one, which makes the processing of knowledge rather awkward, requiring consideration of a large number of details concerning not the meaning of the processed information but the way it is represented in memory;
- in modern computers, there is a very low level of hardware-implemented operations on non-numeric data and there is no hardware support for logical operations on knowledge fragments with a complex structure, which makes manipulating such fragments very difficult.

Attempts to overcome the limitations of traditional von Neumann computers have led to the appearance of many approaches associated with particular changes in the principles of logical organization of computers, primarily depending on the classes of problems and subject domains that a particular class of computers focuses on. All these tendencies, considered together, allow outlining some key principles of the logical organization of computers focused on the knowledge processing (knowledge processing machines – KPM). Let us list the main of these tendencies:

- transition to nonlinear memory organization (see [3], [4]) and hardware interpretation of complex data structures (see [5], [6], [7]);
- hardware implementation of associative access to information (see [3], [8], [9], [10], [11], [12], [13]);
- implementation of parallel asynchronous processes over memory (see [3], [14]), and, in particular, development of computing machines controlled by data flow (see [15], [16], [17], [18]);
- hardware interpretation of high-level languages (see [19], [20], [21]);

- development of database management hardware tools (database processors) (see [22], [23], [24]).

At the intersection of these tendencies, different classes of computing devices have appeared at different times. Let us list some of them:

- machines with hardware interpretation of complex data structures (see [6], [25], [26]);
- machines with developed associative memory (see [10], [27], [28]);
- associative parallel matrix processors (see [29]);
- homogeneous parallel structures for solving combinatorial logic problems on graphs and hypergraphs (see [30]);
- various graph processing devices (see [31], [32], [33], [34]), in particular, based on FPGA (see [35], [36], [37]) and vector processors (see [38]);
- systems that process information directly in memory by evenly distributing functional means in memory and, in particular, the processor-memory proposed by M. Weinzweig, focused on solving artificial intelligence problems (see [39], [40]);
- machines controlled by data flow (see [15], [18], [29]) and, in particular, processors, that are reconfigurable taking into account the semantics of the input data flow (see [41]);
- recursive computing machines (see [3]);
- relational database processors (see [9], [18], [22]);
- computers with restructurable memory (see [42], [43], [44], [45]);
- active semantic networks (M-networks) (see [46]);
- associative homogeneous environments (see [47]);
- neural-like structures (see [48], [49]). In recent years, the active development of the theory of artificial neural networks has led to the development of various approaches to the building-up of high-performance computers designed for training and interpretation of artificial neural networks (see [50], [51], [52]) and their implementation in various software and hardware complexes. In a separate direction, the so-called neuromorphic processors (see [53]) are distinguished by high performance and low power consumption.
- machines for interpreting logical rules (see [54]).

In addition, the development of graphics processors (graphics processing unit, GPU) has led to the possibility of organizing parallel computing directly on the GPU, for which specialized software and hardware architectures are being developed, for example, CUDA (see [55]) and OpenCL (see [56]). The advantage of the GPU in this case is the presence of a large number of cores within one GPU (compared to the central processor), which makes it possible to effectively solve problems with natural parallelism on such an architecture (for example, operations with matrices). Works dedicated to the principles of processing graph structures on the GPU

are also being developed (see [57], [58], [59]).

In general, it can be said that due to the increase in the performance of modern computers, the number of developments of specialized hardware solutions has decreased in recent decades, since many complex computing problems can now be solved on traditional universal architectures in an acceptable time. As shown above, the exception is mainly specialized computers for processing artificial neural networks and other graph models, which is conditioned by the high demand for such models and their complexity.

At the same time, most of these approaches (even if they deviate far enough from the basic principles of computer organization proposed by von Neumann) implicitly retain the point of view of the computer as a large arithmometer and thereby retain its orientation to numerical problems. Works aimed at developing hardware architectures, designed to process information represented in more complex forms than in traditional architectures, have not been widely distributed and used due, firstly, to the specifics of the proposed solutions and secondly, due to the lack of a common universal and unified coding language for any information, in the role of which within the *OSTIS Technology*, the *SC-code* acts, as well as the appropriate proven technology for developing software systems for such hardware architectures. Thus, developers of such architectures often face the need to develop specialized software for these architectures, which ultimately leads to a strong limitation in the scope of application of such architectures, since their use turns out to be reasonable only if the complexity of developing specialized software proves itself, taking into account the low efficiency of solving the corresponding problems on more traditional architectures.

The *SC-code*, which is the formal basis of the *OSTIS Technology*, was originally developed as a language for encoding information in memory of *associative semantic computers*, so it originally contains principles such as universality (the ability to represent knowledge of any kind) and unification (uniformity) of representation, as well as minimization of the *Alphabet of the SC-code*, which, in turn, makes it easier to create a hardware platform that allows storing and processing texts of the *SC-code*.

The main methodological feature of the proposed approach to the development of hardware implementation tools for intelligent systems support is that such tools should be developed not before but after the main terms of the corresponding technology for the design and operation of intelligent systems will be tested on modern technical means. Moreover, within the *OSTIS Technology*, the methodology of transition to new hardware tools has been clearly thought out, which affects only the lowest level of the technology – the level of implementation of the basic semantic network processing machine (interpreter

of the *SCP Language*).

The project of the *associative semantic computer* has a long history, the main stages of which are the following ones:

- 1984 – at the Moscow Institute of Electronic Technology, V. Golenkov defended the PhD dissertation on the topic “Structural organization and processing of information in electronic mathematical machines controlled by the flow of complex structured data”, in which the basic principles of semantic associative computers were formulated and considered (see [60]).
- 1993 – the Goskomprom Commission carried out successful tests for the prototype of the *associative semantic computer* developed on the basis of transputers within the research project “Parallel graph computing system focused on solving artificial intelligence problems” (see [61], [62]).
- 1996 – V. Golenkov defended the doctoral dissertation on the topic “Graphodynamic models and methods of parallel asynchronous processing of information in intelligent systems” (see [63]).
- 2000 – at the Institute of Management Problems of the Russian Academy of Sciences, P. Gaponov defended the PhD dissertation on the topic “Models and methods of parallel asynchronous processing of information in graphodynamic associative memory” (see [64]).
- 2000 – at the Institute of Software Systems of the Russian Academy of Sciences, V. Kuzmitsky defended the PhD dissertation on the topic “Principles of building a graphodynamic parallel computer focused on solving artificial intelligence problems” (see [65]).
- 2004 – at the Belarusian State University of Informatics and Radioelectronics, R. Serdyukov defended the PhD dissertation on the topic “Basic algorithms and tools for information processing in graphodynamic associative machines”, in which the basic software of semantic associative computers was considered (see [66]).

At the same time, despite the presence of a working prototype of the *associative semantic computer*, based on transputers, the main attention within the corresponding project and other listed works was paid to the principles of organizing distributed parallel processing of *SC-code* constructions, in particular, the *SCD Language* (Semantic Code Distributed) was developed for distributed storage of *SC-code* constructions and the *SCPD Language* for their distributed parallel processing. However, the general principles of information storage and the general architecture of each of the processor elements (of transputers) remained behind von Neumann. In particular, to encode *SC-code* constructions in traditional address memory, appropriate data structures have been developed, close to

those described in [67].

Thus, we can say that the reasonableness and necessity of the development of the *associative semantic computer*, as well as the competence of the authors in this field is confirmed by more than 30 years of experience and a number of successful projects in this direction, however, at the same time, in the previous works, all the shortcomings of the von Neumann architecture discussed above have not been fully eliminated, and the development and implementation of the *associative semantic computer* project that eliminates these shortcomings remain relevant.

III. ANALYSIS OF EXISTING COMPUTING SYSTEM ARCHITECTURES

As shown in the previous paragraph, in order to overcome the shortcomings of existing computing system architectures, including the von Neumann one, many different approaches have been proposed. When developing new architectures and, in particular, the architecture of the *associative semantic computer*, it is advisable to identify the main features of classification and the corresponding classes (types) of computing system architectures in the form of an appropriate ontology. Let us consider a fragment of such an ontology developed on the basis of an analysis of existing solutions and the approaches identified by its results.

architecture of a computing system

⇒ *subdividing**:

- *architecture of a computing system with global RAM*
 - ⇒ *subdividing**:
 - *architecture of a computing system with global data RAM*
 - *architecture of a computing system with global program RAM*
 - *architecture of a computing system with global program and data RAM*
 - ⇒ *note**:
 - [An example of such an architecture is the von Neumann architecture.]
- *architecture of a computing system without global RAM*

⇒ *subdividing**:

- *architecture of a computing system with a single global internal memory*

- *architecture of a computing system with multiple global internal memory*

} *subdividing**:

- *architecture of a computing system with restructurable interprocessor connections*
- *architecture of a computing system without restructurable interprocessor connections*

} *subdividing**:

- *architecture of a computing system with structurally evolving memory*
- *architecture of a computing system without structurally evolving memory*

} *subdividing**:

- *architecture of a computing system with associative access to global (internal) memory*
 - ⇒ *note**:
 - [The associative type of access is important in systems focused on data storage with a complex structure and focused on scalable (including local) information processing mechanisms.]
- *architecture of a computing system without associative access to global (internal) memory*

} *subdividing**:

- *architecture of a computing system with addressable access to global memory with linear address space*
 - ⇒ *note**:
 - [Examples of such an architectures are ones that are most used at the moment, including the von Neumann architecture. The problems of device and data management for such architectures are considered in the work [68].]
- *architecture of a computing system without addressable access to global memory with linear address space*

} *subdividing**:

- *architecture of a computing system with a system of register data processing commands*
 - ⇒ *note**:
 - [Most of the architectures currently in use are examples of architectures of this class, including the

von Neumann architecture. Architectures with a system of register data processing commands are convenient for data management problems both for image processing systems in user interface problems and for machine learning problems based on linear algebra apparatus.]

- *architecture of a computing system without a system of register data processing commands*
- }
⇒ subdividing*:
{• *architecture of a computing system with a command system for stack data processing*
⇒ *note**:
[Examples of the application of such an architecture are LISP machines (see [69], [70], [71]), other examples can also be found in the works [72], [73].]
- *architecture of a computing system without a command system for stack data processing*
- }
⇒ subdividing*:
{• *architecture of a computing system with support for a command system for processing generalized strings*
⇒ *note**:
[An example of such an architecture is the architecture that supports the operations of the generalized strings and lists processing model proposed in the work [74]. The underlying model makes it possible to efficiently perform operations not only on strings and lists but also to work with key-value relations in order to integrate them into knowledge-driven systems. The software implementation of this model uses B-trees.]
- *architecture of a computing system without support for a command system for processing generalized strings*
- }
⇒ subdividing*:
{• *architecture of a computing system with support for a command system for processing graph structures*
⇒ *note**:

[An example of such an architecture is the architecture of the Leonhard computer (see [75]). This computer is focused on processing graph and hypergraph structures of various types, including hierarchical graphs (see [76]). The representation is supported in the form of strings and a list of adjacent vertices, ordered local lists of incident edges, and a global ordered list of incident edges.]

- *architecture of a computing system without a support system for processing graph structures*
- }
⇒ subdividing*:
{• *architecture of a computer system with a command system for (hardware) knowledge processing*
⇒ *note**:
[An example of such an architecture is the architecture of the Leonhard computer (see [75]). The Leonhard computer supports the DISC command systems (Discrete Instruction Set Computing) (see [76]). The DISC command system supports the following commands: creating an integer relation with a schema that is a set of objects of a formal context (the first domain of a binary relation), whereas the corresponding set of images is a set of non-negative integers (the second domain of a binary relation); adding a pair to a formal context containing an object (key) to be added, which is added as a tuple by adding elements of this tuple, together with an integer image (value) for this object; getting the next or previous object in a linearly (lexicographically) ordered list of objects; getting the next larger or previous smaller object in a linearly (lexicographically) ordered list of objects; getting the minimum or maximum object in a linearly (lexicographically) ordered list of objects; getting the number (cardinality of the set) of images for a given object (key tuple); searching pairs by key; deleting pairs; deleting all pairs of

formal context, including objects (keys) and images (values); a slice (subset) of formal contexts; combination, intersection, and complementation of formal contexts. B+ trees are used to represent the processed data. Other architectures consider the implementation of knowledge processing operations using a logical model of knowledge representation (see [77]), LISP structures (see [69], [70]), generalized formal languages (see [74], [78]). In the last case, for the development of a system of knowledge processing commands, the transition from knowledge processing to meta-knowledge processing (based on the semantics of becoming relevant and irrelevant) is considered, the result of which is a system of meta-operations (see [78]).]

- *architecture of a computer system without a command system for (hardware) knowledge processing*

}
⇒ subdividing*:

- *architecture of a computing system with adaptive data distribution*
⇒ *note**:
[Adaptive data distribution (including, as a special case, virtual address space) is important for data (and knowledge) management and virtualization problems for multitasking and multi-user systems, as well as also closely related to the scalability capabilities of the system.]

- *architecture of a computing system without adaptive data distribution*

}
⇒ subdividing*:

- *architecture of a computing system with a command system for non-local information processing*
⇒ *note**:
[An example of such an architecture is a cellular automaton. Elementary cellular (binary) automata are divided into: ones rapidly transitioning into a homogeneous state (a state consisting only of zeros or units); rapidly transitioning into a

stable or cyclic state; remaining in a chaotic (random) state; forming both areas with a stable or cyclic state and areas in which complex interactions of elements of states manifest themselves, up to Turing-complete ones.

Information processing using cellular automata allows building computing systems, including ones with a tunable (including fractal-like) structure based on local parallel (competitively) performed simple rules. There are varieties of cellular automata that support irreversible, reversible, deterministic, non-deterministic, specialized, universal (including Turing-complete) computations. The work of cellular automata resembles wave processes propagating in the environment of processor-memory elements of the *associative semantic computer*, considered below.]

- *architecture of a computing system without a command system for non-local information processing*

}
⇒ subdividing*:

- *architecture of a computing system with strictly binary data representation in RAM*
⇒ *note**:
[Most modern architectures of digital computing systems, including implementations of the von Neumann architecture, use binary representation]

- *architecture of a computing system with not strictly binary data representation in RAM*

}
⇒ subdividing*:

- *architecture of a computing system with strictly discrete data representation*
⇒ *note**:
[An example of such an architecture is the von Neumann architecture.]

- *architecture of a computing system without strictly discrete data representation*

}
⇒ subdividing*:

{• *architecture of a computing system with discrete data representation*
 \subset *architecture of a computing system with strictly discrete data representation*
 • *architecture of a computing system without discrete data representation*
 }
 \Rightarrow *subdividing**:
 {• *architecture of a computing system with data flow control*
 \Rightarrow *note**:
 [Architectures of computing systems with data flow control are seen as more natural when solving many artificial intelligence problems. Variants of such architectures are considered in the works []. The architecture of cellular automata can be considered as the architecture of a computing system controlled by a data flow.]
 • *architecture of a computing system without data flow control*
 }
 \Rightarrow *subdividing**:
 {• *architecture of a computing system with data flow control*
 \Rightarrow *note**:
 [An example of such an architecture is the von Neumann architecture.]
 • *architecture of a computing system without command flow control*
 }
 \Rightarrow *subdividing**:
 {• *architecture of a computing system with a processor with an arithmetic-logical unit*
 \Rightarrow *note**:
 [An example of such an architecture is the von Neumann architecture.]
 • *architecture of a computing system without a processor with an arithmetic-logical unit*
 }
 \Rightarrow *subdividing**:
 {• *architecture of a computer system with a control unit with an instruction counter*
 \Rightarrow *note**:
 [An example of such an architecture is the von Neumann architecture.]
 • *architecture of a computer system without a control unit with an instruction counter*
 }

\Rightarrow *subdividing**:
 {• *architecture of a computer system with a control unit with a command register*
 \Rightarrow *note**:
 [An example of such an architecture is the von Neumann architecture.]
 • *architecture of a computer system without a control unit with a command register*
 }
 \Rightarrow *subdividing**:
 {• *architecture of a computing system with an input-output device*
 \Rightarrow *note**:
 [An example of such an architecture is the von Neumann architecture.]
 • *architecture of a computing system without an input-output device*
 }
 \Rightarrow *subdividing**:
 {• *architecture of a computing system with access to an external (operational) storage device*
 \Rightarrow *note**:
 [An example of such an architecture is the von Neumann architecture.]
 • *architecture of a computing system without access to an external (operational) storage device*
 }
 \Rightarrow *subdividing**:
 {• *architecture of a computing system with scalable (modular) global memory*
 \Rightarrow *note**:
 [Scalability as a feature of an architecture is important for systems focused on training (self-training) in order to solve a wide class of problems. Such architectures can be focused on the processing of knowledge structures integrated into a single semantic space.]
 • *architecture of a computing system without scalable global memory*
 }
 \Rightarrow *subdividing**:
 {• *architecture of a computing system with support for an active graph memory model*
 \Rightarrow *note**:
 [An active graph memory model in the architectures of computa-

- tional systems is important for the efficient and consistent (convergent) implementation of parallel knowledge processing operations, including mechanisms of excitation and inhibition of knowledge processing operations.]
- *architecture of a computing system without support for an active graph memory model*
- }
⇒ *subdividing**:
{• *architecture of a computing system with support for parallel information processing*
⇒ *note**:
[Architectures of computing systems with support for parallel knowledge processing are important for the effective implementation of knowledge processing processes (see [78]), improving the performance and scalability of knowledge processing systems, including multi-agent systems in the form of intelligent computer systems and collectives of intelligent computer systems. Various models of architectures of computing systems with support for parallel knowledge processing are considered in the work [65].]
- *architecture of a computer system with sequential information processing*
- }
⇒ *subdividing**:
{• *architecture of a computing system with support for a sequential consistency model of global RAM*
⇒ *note**:
[Architectures with support for consistency models are focused on solving the problem of controlling interacting processes, including their synchronization and synchronous and asynchronous mechanisms for running knowledge processing algorithms. The purpose of supporting a sequential consistency model (see [78]) is to ensure the existence of global states of the knowledge base as structures of a single semantic space in intelligent computer systems. To ensure a particular consistency
- model, various mechanisms can be used, considered in the works [64], [66], as well as [67].]
- *architecture of a computing system without support of a sequential consistency model of global RAM*
- }
⇒ *subdividing**:
{• *architecture with support for causal memory consistency model*
⇒ *note**:
[The purpose of supporting the causal consistency model (see [78]) is to ensure interoperability and convergence in a single semantic space of knowledge structures of agents in collectives of intelligent computer systems. To ensure a particular consistency model, various mechanisms can be used, considered in the works [64], [66], as well as [67].]
- *architecture without support for causal memory consistency model*
- }
⇒ *subdividing**:
{• *architecture of the computing system asymmetric*
⇒ *note**:
[The architectures of computing systems with asymmetry are important for the evolution of multi-agent systems, intelligent computer systems, and their collectives, in which asymmetry is considered in a broad sense, including as the heterogeneity of such systems or collectives. A special case of heterogeneity is the heterogeneity of the architecture, which allows implementing both integrated and hybrid models of knowledge processing within intelligent computer systems and their collectives.]
- *architecture of the computing system symmetric*
- }
- To determine the architecture of *associative semantic computers*, in accordance with the identified classes and features, as well as the general principles underlying such architectures, it is necessary to consider specific sets of architectures within the appropriate feature space and conduct a comparative analysis of the elements of these sets in order to justify the choice of (optimal) architecture

variants for *associative semantic computers*.

At this stage of the work, several main variants for the architecture of *associative semantic computers* have been specified, which are considered in more detail below.

IV. GENERAL PRINCIPLES UNDERLYING ASSOCIATIVE SEMANTIC COMPUTERS FOR OSTIS-SYSTEMS

The proposed approach to the development of the *associative semantic computer* is based on the ideas considered in the works of V. Golenkov (see [63]) and developed in the work of V. Kuzmitsky [65].

When formalizing subject domains that have a rather complex semantic organization, the processed data is naturally grouped into some complex structures. The efficiency of solving problems related to the processing of complex structured data on multiprocessor computing systems increases significantly when the structure of the connections between the processor elements of the computing system solving this problem coincides with the structure of the data processed during its solution (or, more generally, is displayed in the structure of the processed data in a simple and natural way). With the transition to data processing of an increasingly complex structural and semantic organization (and then to knowledge processing), the maintenance of high efficiency of the computing system is ensured mainly by increasing the number of processor elements working simultaneously and complicating the structure of connections between them (see [65]).

We will consider such a tendency in the development of computer hardware as the main line of evolution that creates prerequisites for the appearance of *associative semantic computers*. It includes parallel regular special processors (vector, matrix ones), special calculators for solving problems on graphs, and hardware support for semantic and neural networks. This line is also joined by associative processors (in which associative memory cells act as processor elements), database processors, and computing systems that effectively solve certain classes of problems due to the coincidence of the structure of connections between processor elements with the structure of the information graph of the algorithm (systolic calculators, data flow machines) (see [65]).

A natural result of the development of computing systems is the transition to systems that change the structure of connections between processor elements in the process of functioning. Such systems adjust their internal structure to the structure of the processed data and the information graphs of the algorithms of the problems being solved and can solve different classes of problems while maintaining high efficiency.

Thus, a developed computer focused on the knowledge processing should generally be a collective of special processors focused on the most effective solution of certain classes of problems and have the following features:

- Special processors are a multiprocessor computing system.
- The structure of the connections between the processor elements of special processors coincides with the data structure or (less often) with the structure of the information graph of the algorithm for solving the problem.
- The connections between the processor elements of special processors have a reconfigurable structure.
- The set and functions of special processors are determined for each knowledge processing machine specifically depending on the set of subject domains that this machine is focused on and the specifics of the problems solved in these domains.
- The set of knowledge processing mechanisms determined for some semantic processor should be “immersed” in the language of knowledge representation and processing. At the same time, the languages of semantic networks seem to be the most convenient for this purpose.
- Processor elements correspond to vertices or fragments of a semantic network.
- Information processing is reduced to a change in the structure of connections between processor elements, corresponding to a change in the configuration of the semantic network.

As a semantic special processor, we can propose a nonlinear (graph) restructurable (dynamic) processor-memory that implements some kind of semantic network processing language in hardware, and the computer of this kind itself can thus be called a graphodynamic parallel associative computer, or an *associative semantic computer*.

Taking into account the above, as well as the general principles of information processing in ostis-systems described in [67], let us consider more specifically the principles underlying the implementation of *associative semantic computers*:

- Nonlinear memory – each elementary fragment of a text stored in memory can be logically incident to an unlimited number of other elementary fragments of this text. Thus, memory cells, unlike ordinary memory, are connected not by fixed conditional connections that specify a fixed sequence (order) of cells in memory but by a logically or even physically (using technical means of switching) conducted connections of accidental configuration. These connections correspond to arcs, edges, hyperedges of the graph (sc-text) recorded in memory.
- Restructurable (reconfigurable) memory – the operation of processing the information stored in memory is reduced not only to changing the state of the elements but also to reconfiguring the connections between them. That is, during the processing of information in restructurable memory, not only and

not even so much the states of memory cells change, as is the case of ordinary memory, but the configuration of connections between these cells. That is, in restructurable memory, during the information processing, not only the labels on the vertices of the graph recorded in memory are redistributed, but also the structure of this graph itself changes.

- As an internal way of encoding knowledge stored in the memory of the *associative semantic computer*, a universal (!) method of nonlinear (graph-like) semantic representation of knowledge – SC-code – is used.
- Information processing is carried out by a collective of agents working on shared memory. Each of them reacts to the corresponding situation or event in memory (a computer controlled by stored knowledge).
- There are software-implemented agents whose behavior is described by agent-oriented programs stored in memory, which are interpreted by the corresponding collectives of agents.
- There are basic agents that cannot be software implemented (in particular, these are agents of interpretation of agent programs, basic receptor agents-sensors, basic effector agents).
- All agents work on shared memory at the same time. Moreover, if several conditions of its application arise for an agent at some point in time in different parts of memory, different information processes corresponding to the specified agent in different parts of memory can be run simultaneously.
- In order for information processes of agents running in parallel in shared memory not to “interfere” with each other, its current state is recorded and constantly updated in memory for each information process. That is, each information process informs others about its intentions and wishes, which other information processes should not interfere with. The implementation of such an approach can be carried out, for example, on the basis of the mechanism for locking elements of semantic memory, considered in [67].
- The processor and the memory of the *associative semantic computer* are deeply integrated and form a single processor-memory. The processor of the *associative semantic computer* is evenly “distributed” over its memory so that processor elements are simultaneously computer memory elements. That is, each cell is supplemented by a functional (processor) element, and tunable connections between cells become switched communication channels between processor elements. At the same time, each processor element has its own special internal register memory, reflecting aspects of the current state of performing elementary operations of the micro-program language that provide interpretation of a higher-level

language (SCP Language) that are important for this processor element.

Information processing in the *associative semantic computer* is reduced to reconfiguration of communication channels between processor elements, therefore the memory of such a computer is nothing but a switchboard (!) of the specified communication channels. Thus, the current state of the configuration of these communication channels is the current state of the information being processed. This principle provides a significant acceleration of information processing by eliminating the stages of transferring information from memory to the processor and back, but it is paid for at the cost of a large redundancy of processor (functional) means evenly distributed over memory.

V. ARCHITECTURE OF ASSOCIATIVE SEMANTIC COMPUTERS FOR OSTIS-SYSTEMS

associative semantic computer

- ⊂ *computer with graphodynamic associative memory*
- := [associative semantic computer]
- := [sc-computer]
- := [hardware implemented basic interpreter of semantic models (sc-models) of computer systems]
- := [hardware implemented ostis-platform]
- := [hardware variant of the ostis-platform]
- := [associative semantic computer controlled by knowledge]
- := [computer with a nonlinear restructurable (graphodynamic) associative memory, the processing of information in which is reduced not to a change in the state of memory elements but to a change in the configuration of the connections between them]
- := [universal computer of a new generation, specially designed for the implementation of semantically compatible hybrid intelligent computer systems]
- := [universal computer of a new generation, focused on hardware interpretation of logical and semantic models of intelligent computer systems]
- := [universal computer of a new generation, focused on hardware interpretation of ostis-systems]
- := [ostis-computer]
- := [computer for the implementation of ostis-systems]
- := [computer controlled by the knowledge represented in the SC-code]
- := [computer focused on processing SC-code texts]
- := [computer whose internal language is an SC-code]
- := [computer that implements sc-memory and interprets scp-programs]
- := [our proposed new generation computer focused on the implementation of intelligent computer

systems and using SC-code as an internal language]

⇒ *subdividing**:

- {• *scp-computer*
:= [sc-computer that provides interpretation of scp-programs]
⇒ *generalized model**:
basic ostis-platform
- *sc-computer with an extended set of hardware-implemented sc-agents*
:= [sc-computer that provides interpretation of scp-programs]
⇒ *generalized model**:
specialized ostis-platform

}

scp-computer

:= [minimum configuration of a hardware-implemented ostis-platform, within which the interpretation of scp-programs is provided]

:= [minimum configuration of a hardware-implemented ostis-platform, within which only basic sc-agents are implemented in hardware]

⇒ *explanation**:
[Within the scp-computer, (1) sc-memory, (2) basic sc-agents providing interpretation of scp-programs, (3) elementary receptor sc-agents, (4) elementary effector sc-agents are implemented in hardware.]

To refine the architecture of *associative semantic computers*, it is necessary to clarify:

- the basic structure of the computer and, in particular, its processor-memory;
- the alphabet of elements stored in the processor-memory of the computer;
- the system of commands interpreted by a computer;
- principles of controlling the process of interpreting these commands;
- the system of micro-programs that ensure the implementation of the principles for controlling the specified process.

Since the internal language for encoding information of the *associative semantic computer* is the SC-code, the alphabet of elements stored in the processor-memory of the computer coincides with the *Alphabet of the SC-code*, considered in [67]. At the same time, the alphabet of physically encoded syntactic labels can be expanded, for example, for performance reasons, by analogy with how it is done in the software implementation of the *ostis-platform* [67].

As a command system for the *associative semantic computer*, the *SCP Language*, discussed in detail in [67], is proposed. Thus, as already mentioned in the specified paragraph, the *SCP Language* is an assembler for the *associative semantic computer*.

To determine the basic structure of the *associative semantic computer*, let us clarify the variants of such a structure proposed in the works of V. Golenkov and V. Kuzmitsky (see [63], [65]). In particular, in the work of V. Kuzmitsky, a transition from coarse-grained architectures of graphodynamic machines to fine-grained ones is proposed (see [65]).

Models of coarse-grained architectures have parallel functioning modules with the following features:

- each module has a strictly fixed functional purpose within the architecture of the graphodynamic machine as a whole (the so-called global functional purpose);
- each module has a relatively large amount of memory (the number of memory elements is much larger than the total number of modules);
- the memory of each type of module has its own non-elementary set of operations that perform some completed transformations on sufficiently large fragments of memory.

The main formal difference for models of fine-grained architectures is a different ratio between the total number of modules and the number of memory elements of each module (module memory capacity), which tends to unity, and the level of complexity of model operations. Accordingly, the features of models of fine-grained architectures can be considered as the following (see [65]):

- For each module separately, its functional purpose may not be viewed within the graphodynamic machine as a whole. At the same time, each separate module at a particular time can have some so-called local functional purpose, the corresponding set of which can already be considered as having a certain so-called global functional purpose within the graphodynamic machine as a whole.
- The amount (number of elements) of memory of each module is minimal and tends to unity. As a result, the total number of modules is comparable to the total number of memory elements of all modules.
- For each module (in general), the set of operations performed on its memory is elementary and limited (finite), since it affects only one element (or only a few elements) of memory and is determined by the obvious limitation (finiteness) in the semantics of interpreting the contents of the graph memory element in a graphodynamic machine.

The reasonableness of the transition from coarse-grained to fine-grained architectures is conditioned by a corresponding increase in the degree of potential parallelism in knowledge processing procedures.

Taking into account the above, we can talk about several options for clarifying the basic structure of the *associative semantic computer*, each of which has certain advantages

and disadvantages. Let us consider these options in more detail.

A. Architecture of an associative semantic computer based on the von Neumann architecture

One of the most logical and architecturally simplest options for the hardware implementation of the ostis-platform is the implementation of means for storing SC-code constructions and interpreting scp-programs at the hardware level, similar to how it is done in software versions of the ostis-platform based on modern computers [67]. In this case, the overall architecture of the computer remains behind von Neumann (with the explicit allocation of a separate processor module and a separate memory module). The main features of this implementation option include the following ones:

- A memory module (implementation of *sc-memory*) is a set of cells, each of which can store some sc-element or can be empty. Each cell of such memory has some unique internal address, similar to the address of the von Neumann memory cells. At the same time, when processing information stored in such memory at the level of the command language (SCP Language), unlike von Neumann memory, cell addresses are not taken into account; access to sc-elements is carried out strictly by incidence relations between them. The exception is some key sc-elements, the set of which is specified separately and accessed by some other identifier, for example, the system sc-identifier or the main sc-identifier for some external language but not by address. It is assumed that if some sc-element is stored in a cell, it stores information characterizing this sc-element, namely:
 - a syntactic label specifying the type of the corresponding sc-element;
 - contents of the sc-file or a link to an external file system (if the sc-file is stored);
 - a list of the incidence relations of this sc-element with other sc-elements, which actually means storing a set of memory cell addresses corresponding to the sc-elements incident to this sc-element. The specific list of types of stored relations can be specified depending on the implementation. For example, by analogy with how it is done in the software implementation of the ostis-platform [67], it is advisable to store in a memory cell the address of the cell corresponding to the first sc-connector incident to the corresponding sc-element with the corresponding incidence type, and within the cell corresponding to this sc-connector, to store the address of the cell corresponding to the next sc-connector, incident to the same sc-element with the same incidence type, etc. With this approach, the size of each memory cell can be fixed.

- a label of the sc-element lock indicating the label of the corresponding process;
- an access level label and any other labels, if necessary.
- The processor module implements a set of commands corresponding to *atomic types of scp-operators*.
- To connect with the external environment, a *terminal module* is introduced (see [63], [65]), which in general can be implemented in different ways and whose tasks are:
 - to prepare (generate) information coming from the external environment for its subsequent loading into the processor module and the module memory;
 - to transfer (use, implement) information prepared (received, represented) in the processor module and memory module to the external environment.
- To store the contents of large sc-files, it may be advisable to have a separate file memory built according to von Neumann principles. Then the semantic memory cells corresponding to such sc-files will store not their contents directly but the address of this file on the file memory.
- To implement the principles of multi-agent information processing proposed within the OSTIS Technology [67], it is necessary to implement (for example, within the terminal module) an event registration and processing subsystem that will allow the initiation of sc-agents when the corresponding events occur in memory.

The advantages of this implementation option include:

- Relative simplicity and low labor intensity of implementation compared to the development of the full-fledged processor-memory variant discussed below. In particular, with the availability of a stable version of the ostis-platform software implementation, in which at least the lower level is implemented in sufficiently low-level languages, such as C, to simplify the process of developing hardware architecture, it is possible to use automation tools for the transition from C programs to descriptions in hardware description languages (HDL, for example, VSDL and Verilog), also known as “C to HDL”. Popular tools and languages of this class include LegUp (see [79]), VHDLPlus (see [80]), SystemC (see [81]), MyHDL for Python (see [82]), and many others.
- Simplicity of integration with modern computer systems, in particular, a hybrid variant can be considered, in which the *associative semantic computer* is implemented as a separate plug-in module for a modern computer designed to increase the efficiency of processing sc-constructions.

The obvious key disadvantage of this option is its

orientation to the von Neumann architecture with all its disadvantages listed above. In addition, in this option, by default, parallel processing of sc-constructions is not provided at the hardware level. This disadvantage is partially eliminated in the next version of the coarse-grained architecture of *associative semantic computers*.

B. Variant of the coarse-grained architecture of *associative semantic computers*

The goal of the transition to the coarse-grained architecture of *associative semantic computers* is to implement parallel processing of sc-constructions at the hardware level.

The main features of this implementation option include the following ones:

- The *associative semantic computer* is divided into several modules of the same type, arranged in a similar way as the implementation option for the associative semantic computer considered in the previous paragraph, built on the basis of the von Neumann architecture. Such modules will be called “combined modules”, since such a module has its own processor module and its own memory module (storage module); there are no separately allocated common processor modules. There may be a separate shared memory module, into which, if necessary, information that does not fit into the memory of a particular combined module will be recorded.
- The terminal module that provides the connection of the system of combined modules with the external environment is still allocated.
- Separately, a file memory module can be allocated.
- The number of combined modules is relatively small (2 – 16), each module is a sufficiently powerful device (in fact, it is a separate *associative semantic computer*), and, accordingly, one combined module may be enough to solve problems of some classes.
- At the same time, in general, it is necessary to use several combined modules to solve the problem. In this case, the processed sc-construction is distributed among several modules, for which sc-nodes-copies are created, allowing for semantic communication between fragments of the sc-construction stored in different combined modules. To record such constructions, an extension of the SC-code was developed, called SCD-code (Semantic Code Distributed, see [63], [65]), respectively, the constructions of such a language were called scd-constructions, their elements – scd-elements (scd-nodes, scd-arcs).
- Similarly, an extension of the SCP Language, called the SCPD Language, was developed for processing scd-constructions, taking into account the fact that different fragments of the processed construction can be physically stored in different combined modules. At the same time, it is assumed that all elements of

the scd-construction representing an scpd-program (a program of the SCPD Language) should be located in the memory of one combined module, but each scpd-program can have several complete copies in different combined modules.

- To synchronize parallel information processing operations, combined modules exchange messages that can contain both fragments of processed scd-constructions and commands of the SCPD Language. Accordingly, the SCPD Language, in comparison with the SCP Language, has additional tools that support distributed processing of graph constructions (see [63], [65]):
 - The SCPD Language has built-in tools that allow recognizing “your” and “foreign” combined module; for this purpose, operators are introduced to work with module identifiers.
 - It is possible to create a copy of the scd-element in the memory of another module. For this purpose, a group of operators is introduced to work with copies: creating a copy of the scd-element in the specified module, transferring the connections of the original element to the copy, gluing copies of the element together, searching for a copy of this element in a given module, etc.
 - It is possible to explicitly call the scpd-program remotely in the specified processor module. To run the same processes performing in parallel in different processor modules, operators are specified, which run the program in modules from the specified list.
 - There are means of inter-process and intra-process synchronization: message generation operators, message waiting operators, process transfer operators in the waiting mode for completing execution of distributed executing operators, waiting operators for completing execution of all distributed executing operators.
- For message exchange, each combined module has corresponding submodules that allow sending and receiving messages, as well as a message buffer for storing a queue of received messages waiting to be processed and messages waiting to be sent.
- To interpret SCPD-programs, a family of micro-programs is being developed in a language that generally depends on the selected hardware components from which the combined modules are built.

The described implementation option for *associative semantic computers* with coarse-grained architecture also includes the previously mentioned multi-transputer implementation (see [62], [63]). This implementation is based on IBM PC 386 (486, Pentium) and 8 T805 transputers. In Figure 1, this implementation option is schematically shown on 8 transputers, where each transputer simultaneously performs the role of a switching

node (“SN”) and a processor module (“PM”) or a storage module (“SM”). The entire system interacts with the external environment through a terminal module (“TM”).

The main advantage of the coarse-grained architecture of *associative semantic computers* is the orientation to hardware support for parallel processing of SC-code constructions. At the same time, this implementation option has a number of disadvantages:

- Each combined module is built according to the principles of the *von Neumann machine*, accordingly, its disadvantages are not fully eliminated.
- Despite the preservation of the general principles of the SC-code and SCP Language, distributed storage and processing of sc-constructions requires the development of separate language tools, such as an SCD-code and SCPD Language, and their support based on the selected hardware architecture. In addition, as can be seen from the principles of the SCPD Language discussed above, when developing scpd-programs, it is necessary to explicitly take into account the fact that processing is performed distributed.

The next step from the point of view of the hierarchy in the architectures of *associative semantic computers* is the fine-grained architecture of associative semantic computers.

C. Variant of the fine-grained architecture of associative semantic computers

As already mentioned, the reasonableness of the transition from coarse-grained to fine-grained architectures is conditioned by a corresponding increase in the degree of potential parallelism in knowledge processing procedures. At the same time, the maximum possible parallelism will obviously take place with the maximum implementation of fine-grained architectures in which one structural module of processor-memory will correspond to one memory element, that is, in our case, one sc-element.

Let us consider in more detail the principles underlying the fine-grained architecture of the *associative semantic computer*:

- The processor-memory of the *associative semantic computer* consists of modules of the same type, which will be called processor elements of sc-memory, or simply *processor elements*. Each *processor element* corresponds to one sc-element (stores one sc-element). At the same time, at any given moment, each *processor element* can be empty (not store any sc-element) or filled, that is, have a mutually unambiguously corresponding stored sc-element. At the physical level, an appropriate attribute with two meanings is introduced to describe this fact. Thus, each *processor element* is “responsible” for only one sc-element and, unlike the coarse-grained version of the *associative semantic com-*

puter architecture, the problem cannot be solved by one processor element, and the number of such *processor elements* is quite large (corresponds to the maximum possible number of sc-elements stored in the knowledge base of some ostis-system). Experience in the development of applied ostis-systems shows that, on average, the number of sc-elements in the knowledge base of such an ostis-system ranges from several hundred thousand to several million. The situation when it is necessary to represent an sc-construction within the processor-memory, the number of elements of which is greater than the number of *processor elements*, is not currently being considered and requires additional research.

- Each *processor element* (by analogy with a memory cell in the case of the implementation of the *associative semantic computer* on the von Neumann architecture) has some unique internal identifier – an address of the processor element. Addresses of *processor elements*, unlike addresses of von Neumann memory cells, do not provide direct access to processor elements but allow unambiguously identify the processor element when exchanging messages according to the principles discussed below.
- Each processor element has a memory which stores:
 - a syntactic label specifying the type of the corresponding sc-element;
 - the contents of the sc-file or a link to an external file system (if this processor element corresponds to the sc-file);
 - a list of logical connections of this processor element with others, that is, a list of addresses of processor elements associated with this processor element by *logical communication channels*, indicating the type of communication (for more information about *logical communication channels*, see below);
 - a label of blocking sc-elements, indicating the label of the corresponding process;
 - other labels, if necessary (for example, labels of the access level to the stored sc-element);
 - wave micro-programs run by this processor element at the moment (for more information about *wave micro-programs*, see below) and temporal data for these micro-programs, as well as a queue of micro-programs, if necessary.
- Processor elements are interconnected by two types of communication channels – *physical communication channels* and *logical communication channels*:
 - In general, the number of *physical communication channels* for each *processor element* can be arbitrary, in addition, theoretically, *physical communication channels* between processor elements can be rebuilt (reconnected) over time, for example, in order to optimize the time of message transmission

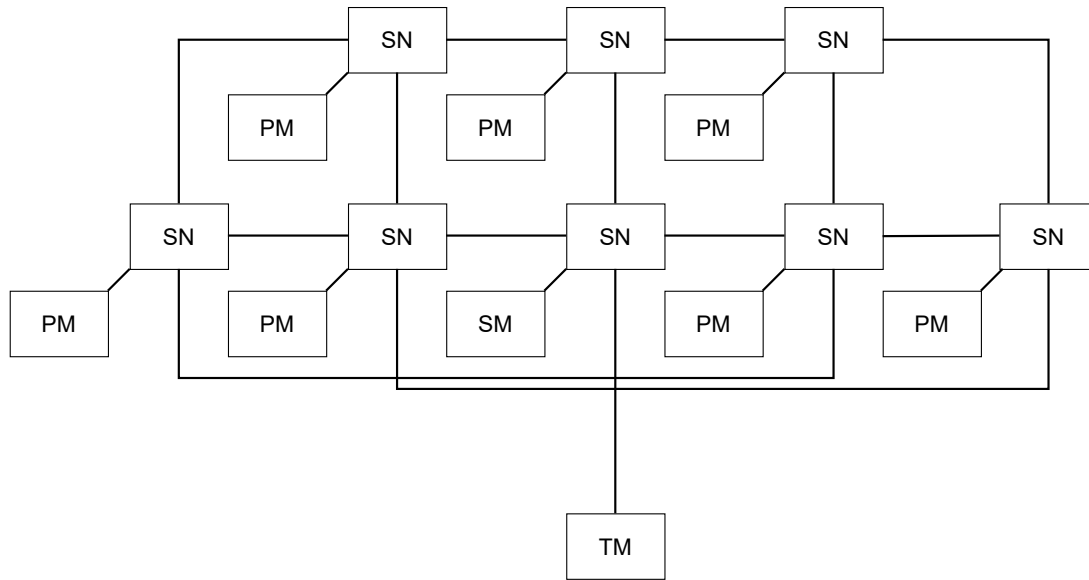


Figure 1. Example of implementing the coarse-grained architecture

- between processor elements. The configuration of *physical communication channels* is not taken into account at the level of logical knowledge processing, both at the level of the SCP Language and at the level of the language of the micro-programs, providing interpretation of commands for the SCP Language, that is, *scp-operators*. For simplification, within this work, we will consider an option of the physical implementation of *sc-memory*, in which each processor element has a fixed and the same number of *physical communication channels* (N) for all processor elements, while the configuration of such communication channels does not change over time. Obviously, the minimum value of N is 2, in this case we will get a linear chain of *processor elements*. With N equal to 4, we will get a two-dimensional “matrix” of *processor elements*, with N equal to 6 – a three-dimensional “matrix” of *processor elements*, etc. As “adjacent” *processor elements* we will call ones that are directly connected by a *physical communication channel*.
- In this case, we can say that each processor element has its own “address” (unique identifier) in some multidimensional space, the number of dimensions (features) of which is determined by number N of *physical communication channels* associated with one *processor element*. In the examples above, the dimensionality of such a space is $N/2$, which suggests that it is advisable to make number N even-numbered.
 - Each *physical communication channel* and each *logical communication channel* are thus defined

- by a pair of *addresses of processor elements*.
- *Logical communication channels* between processor elements are formed dynamically and correspond to *incidence relations* between *sc-elements*. Thus, *logical communication channels* can describe two types of incidence relations – *incidence of sc-pair designations with their components* and *incidence of oriented sc-pair designations with their second components* [67]. At the same time, the configuration of *logical communication channels* in general is not related in any way to the configuration of *physical communication channels*: incident *sc-elements* can be physically stored in processor elements that are not adjacent. At the same time, it is obvious that, in general, some *physical communication channels* may correspond to logical ones.
- In addition to the incidence relations, *logical communication channels* can correspond to other types of connections between *sc-elements*, by analogy with how it is done in the software implementation of the *ostis-platform* [67]. For example, to simplify the implementation of search algorithms in the knowledge base and reduce the amount of memory that each *processor element* should have, it is advisable to store in the memory of the processor element the address of only the first *sc-connector* incident to the corresponding *sc-element* with the corresponding incidence type, and within the processor element corresponding to this *sc-connector*, the address of the next *sc-connector* incident to the same *sc-element* with the same incidence type, etc. With this approach, the

amount of processor element memory that stores logical connections between processor elements can be fixed.

- Each processor element can send messages (micro-programs) to other processor elements and receive messages from other processor elements via *logical communication channels* and has corresponding receptor-effector submodules. At the physical level, messages are transmitted, in turn, via *physical communication channels*, the configuration of which, as mentioned above, is fixed and generally does not depend on the configuration of logical communication channels.
- Thus, processor elements form a homogeneous processor-memory, in which there are no separately allocated modules designed only for storing information and separately allocated modules designed only for its processing.
- To connect such a processor-memory with the external environment, a *terminal module* is introduced, which in general can be implemented in different ways and whose tasks are:
 - preparation (generation) of information coming from the external environment for its subsequent loading into processor modules;
 - transfer (use, implementation) of information prepared (received, represented) in processor modules to the external environment.
- To store the contents of large sc-files, it may be advisable to have a separate file memory associated with processor-memory and built according to traditional von Neumann principles. This is conditioned by the fact that the main purpose of building-up processor-memory is to ensure as much parallelism as possible when processing SC-code constructions, while in the case of storing and processing the contents of sc-files, which by definition are information constructions external to the SC-code, it is advisable to use modern traditional approaches.

These principles allow formulating a key feature of processing information stored within such a processor-memory. Unlike the von Neumann architecture (and other architectures developed around the same time, for example, the Harvard architecture) and even from the *ostis-platform software version*, the proposed processor-memory architecture has no shared memory available for all modules that process information. Due to this, parallel processing of information is greatly simplified, but the implementation of a set of micro-programs for interpreting information processing commands in such memory becomes more complicated, since each processor element becomes very “short-sighted” and “sees” only those processor elements that are connected to it by *logical communication channels*.

Thus, the language for describing the micro-programs

for interpreting commands of the *associative semantic computer* cannot be built as a traditional programming language, for example, of a procedural type, since all such languages assume the possibility of direct address or associative access to random memory elements. The proposed micro-program description language is proposed to be built according to the principles of *wave programming languages* (see [83], [84]) and insertion programming (see [85], [86]).

Within such a micro-programming language, two types of waves are distinguished:

- waves transmitted only via *logical communication channels* (for example, when searching for incident sc-elements);
- waves transmitted over all communication channels (for example, when creating new logical communication channels, that is, when generating new sc-elements).

Let us consider in more detail the principles for interpreting commands (of scp-operators) within the processor-memory considered above:

- Each *processor element* can interpret some limited set of micro-programs. Taking into account the fact that one processor element corresponds to one sc-element, the set of operations associated with the transformation of this sc-element is very limited (generate an sc-element of the specified type, delete an sc-element, change the contents of the sc-file, set or remove the lock label, etc.). Thus, an important task of the processor element is to generate messages for other processor elements and send them.
- Each processor element can generate and store temporary data for micro-programs in memory. It is assumed that the amount of memory available to the processor element is sufficient to represent all the necessary data for a possible set of micro-programs, since such micro-programs are quite simple (see the previous principle). In case, for some reason, overflow still occurs, then various approaches can be used, for example, described in the work [65].
- Each processor element can form a micro-program and send it as a wave message for running by other processor elements. Messages are transmitted via physical communication channels. Since the configuration of physical communication channels is generally not related to the configuration of logical communication channels, each processor element independently decides whether to run the micro-program and transfer it further. Here we can draw an analogy with the wave algorithm for finding a path in a graph (a variant of the breadth search).
- Frequently, processor elements will not run the micro-program but transmit it further, thus, the processor elements themselves also perform the role of switching elements, while, in general, each

processor element can enter an arbitrary number of routes when transmitting messages through logical communication channels between processor elements.

- As in the case of the coarse-grained architecture, each processor element has a queue of micro-programs to be run (incoming messages) and a queue of micro-programs to be sent (outgoing messages). At the same time, within each processor element, it is also possible to talk about the possibility of performing any operations in parallel (for example, generating outgoing messages and processing the current stored sc-element).

Accordingly, a good case can be made about the existence of a hierarchy of micro-programs:

- Micro-programs for changing the stored sc-element:
 - perform the specified transformation of the contents of this sc-node;
 - change the label of the sc-element type (if such a change does not contradict the *Syntax of the SC-code*);
 - replace the lock of this sc-element for the specified process (including removing the label);
 - delete the sc-element.
- Micro-programs for processing sc-elements stored in others (not necessarily adjacent processor elements):
 - generate an incident sc-connector (and a new *logical communication channel*), possibly together with an adjacent sc-element;
 - generate both or one sc-element connected by this sc-connector;
 - find all sc-connectors (that is, the addresses of their corresponding processor elements) of the specified type, incident to this sc-element by the specified incidence type;
 - find sc-nodes incident to this sc-connector.
- Micro-programs for managing the running processes of other micro-programs:
 - forward the specified micro-program for running from this processor element through all specified channels (incident sc-connectors of the specified type) to all adjacent sc-elements of the specified type;
 - wait for the running of the specified type of micro-programs generated by the specified processor element and transmit the result of their running to the processor element that requested the appropriate information.
- And others.

Obviously, when solving a specific problem, these micro-programs can be combined into more complex micro-programs. The above hierarchy is not complete at the moment and requires further clarification.

Based on the principles represented, a hierarchy of

programming languages is formed for the proposed fine-grained architecture of *associative semantic computers*:

- The SCP Language, independent of the implementation of the ostis-platform, on which the programs of sc-agents of knowledge processing are written. The SCP Language is a “watershed” between the platform-dependent part and the platform-independent part of the ostis-system, so it is the lowest-level language among all possible platform-independent languages and at the same time a high-level language from the point of view of the ostis-platform.
- The language of the micro-programs that the processor elements exchange with each other and which are run by these processor elements. In fact, an interpreter of the SCP Language is being developed in this language. It is important to note that the micro-program language is focused on the transmission of messages via *logical communication channels* and does not take into account the configuration of *physical communication channels*. For this, another lower-level language is introduced.
- A language for writing programs for managing processes of exchanging messages (micro-programs). The introduction of such a language is necessary because, as it was said, the micro-programming language itself does not take into account:
 - Configuration of physical communication channels. Thus, when sending a message via a logical communication channel, it is necessary to generate the necessary number of messages depending on the number of available physical communication channels, encode the transmitted message for transmission over a physical communication channel, transmit a message taking into account that the same physical communication channel can generally be included in an arbitrary number of routes between processor elements, decode the message on the receiving processor element. All these problems require the development of appropriate programs.
 - Queuing incoming and outgoing messages inside the *processor element*, adding messages to the queue, extracting messages from the queue for execution, etc.

Advantages of the proposed fine-grained architecture variant of *associative semantic computers*:

- Within the proposed fine-grained architecture, unlike coarse-grained one, there is no need to create copies of sc-elements and to develop special coding languages for the resulting constructions, such as the SCD-code, since each processor element stores one atomic fragment of the entire stored sc-construction, and the number of logical connections with other processor elements is unlimited.

- The above clearly distinguished hierarchy of programming languages makes it possible to exclude at the level of development of user programs (in the SCP Language and higher-level languages based on it) the need to take into account the fact of distributed storage of sc-constructions and the general principles of organizing the ostis-platform. In other words, the development of languages such as the SCPD Language is not required.
- The extensibility of the architecture makes it easy to increase the number of processor elements without significantly reducing performance, since there are no explicitly allocated processor modules and storage modules in the proposed architecture, respectively, the need to transfer information between such modules is eliminated; in addition, the processor module ceases to be a shared resource for a large number of simultaneously run processes. All of the above will eventually solve the problem known as the “bottleneck” problem of the von Neumann architecture (see [87]).
- The key advantage of the proposed fine-grained architecture is its orientation to the maximum possible support for parallel information processing at the hardware level and, ultimately, the possibility of implementing any parallelism models taking into account the problem being solved. In support of this thesis, we can cite the theory of A-systems described in the work of V. Kotov and A. Narinyani [88]. According to the authors, this concept should be interpreted as a universal model for a certain class of parallel systems, which requires clarification in the case of specific implementations. In particular, within this theory, processor elements are distinguished, activation/deactivation of which is carried out by means of the so-called trigger function, which takes the values 0 and 1. It is clear that in a particular implementation, any attribute with the values “true” and “false” can be used as such a function, indicating that a particular processor element should be activated at the next moment in time. The authors show the possibility of formalizing any parallel algorithms based on this model, consider the possibility of reducing such algorithms to sequential ones, synchronization options within such a model. An obvious parallel can be drawn between A-systems and the proposed fine-grained architecture of *associative semantic computers*, taking into account the presence of a wave programming language:
 - processor elements from the theory of A-systems correspond to *processor elements* of the processor-memory;
 - in the role of trigger functions for processor elements, the micro-programs act, transmitted by waves from one processor element to another and,

accordingly, activating the activity of processor elements.

It is worth noting that despite the fact that the considered work on the theory of A-systems has been known for more than half a century, the authors of this work failed to implement the ideas of this theory in hardware. In our opinion, this is conditioned by the fact that the level of development of microelectronics at that time did not meet the requirements necessary for the implementation of the theory of A-systems.

Together with the listed advantages, we can highlight the key disadvantage for the proposed fine-grained version of the architecture of *associative semantic computers*, which consists in a strong dependence of the processor-memory performance on the time of transmission of wave micro-programs from one processor element to another. At the same time, since at the logical level messages are transmitted via *logical communication channels* and in reality – via *physical communication channels*, the processor-memory performance will depend on how closely the configuration of *logical communication channels* corresponds to the configuration of *physical communication channels*. Obviously, in the general case, one-to-one correspondence of these configurations is impossible, since the number of *physical communication channels* incident to a given processor element is limited, unlike the number of *logical communication channels*. Nevertheless, there are several options for optimizing the placement of sc-constructions in the processor-memory:

- When recording (“stacking”) sc-constructions into processor-memory (especially in the case of sufficiently large sc-constructions), it is possible to take into account the semantics of the fragments being recorded and write them in such a way that those sc-elements, to which the message will be transmitted from this sc-element most likely, were physically closer to this sc-element. So, for example, it is possible to take into account the denotational semantics of searching scp-operators, which are focused on processing *three-element sc-constructions* and *five-element sc-constructions*, as well as store sc-elements incident to a given sc-connector as close to it as possible.
- If the number of logical connections between the elements of the sc-construction does not exceed the number of available physical communication channels of the processor element and the sc-graph is planar (although the sc-graph is not a classical graph, we can talk about its planarity by analogy with the planarity of classical graphs), then it is possible to write the sc-construction to the processor-memory in such a way that the configuration of *logical communication channels* mutually uniquely corresponds to some subset of physical communication channels. Thus, it is relevant to develop algorithms

for optimal “stacking” of sc-graphs into processor-memory to ensure the subsequent efficiency of message transmission between processor elements.

- Since the configuration of *logical communication channels* changes during the processing of sc-constructions, it is also advisable to talk about the development of algorithms for repositioning (“defragmentation”) of the sc-construction already recorded in the processor-memory in order to ensure the subsequent efficiency of message transmission. Such reallocation can be performed, for example, according to a schedule during a period when the processor-memory is not used for solving other problems.
- In addition, if there is a hardware capability, the *physical communication channels* can also be re-switched in order to approximate their configuration to the configuration of *logical communication channels*.

Let us consider an example of the optimal variant of writing the simplest *five-element sc-construction* into the proposed processor-memory within the fine-grained architecture of *associative semantic computers*.

In Figure 2, the record of some *five-element sc-construction* in the SCg-code is shown.

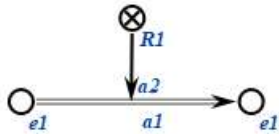


Figure 2. SCg-text. Example of a *five-element sc-construction*

In Figure 3, an incidence graph for the same *five-element sc-construction* is shown, which allows reducing the sc-construction to a classical graph with two types of connections. For clarity, the syntactic types of the corresponding sc-elements are not shown in the Figure.

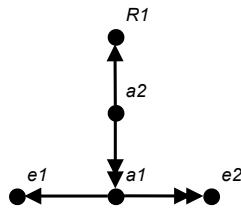


Figure 3. Incidence graph for a *five-element sc-construction*

In Figure 4, one of the possible optimal options for recording the resulting incidence graph into processor-memory is shown. Dotted lines show *physical communication channels* between processor elements, solid lines show *physical communication channels* corresponding to *logical communication channels*. Note that it is advisable to record element **R1** in the *processor element* adjacent

to the *processor element* storing element *e1* or element *e2*, as shown in the Figure. Due to this, the processor elements storing the specified sc-elements are directly connected by a *physical communication channel*, which simplifies communication in the case of sending messages via *physical communication channels* without taking into account *logical communication channels*.

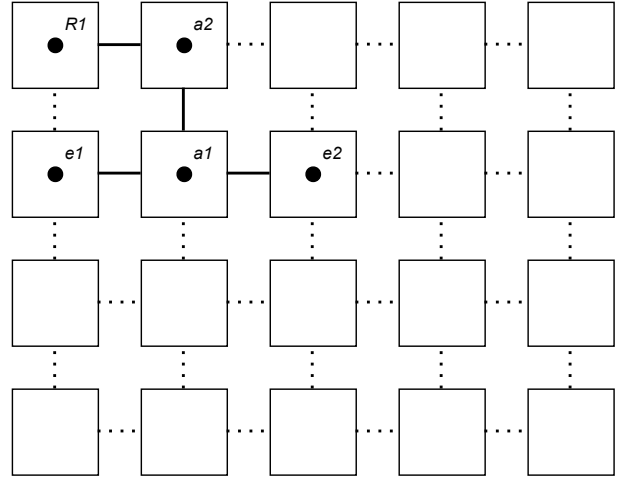


Figure 4. Example of stacking an sc-construction into a processor-memory

VI. CONCLUSION

In the article, the disadvantages of the currently dominant von Neumann architecture of computer systems as a basis for building-up intelligent computer systems of a new generation are considered, the analysis of modern approaches to the development of hardware architectures that eliminate some of these disadvantages is carried out, the need for the development of fundamentally new hardware architectures representing a hardware implementation of ostis-platforms – *associative semantic computers* – is demonstrated.

The general principles underlying *associative semantic computers* are proposed, three possible variants of the architecture of such computers are considered, their advantages and disadvantages are represented.

Further development of the approaches proposed in the work requires solving a number of problems, both technical and organizational ones:

- development of a wave language for recording micro-programs, that are exchanged between processor elements and run by these processor elements;
- development of a language for writing programs for controlling the exchange of micro-programs and managing the queue of micro-programs;
- organization of active participation of specialists in the field of microelectronics in clarifying the principles of implementation of processor elements

and processor-memory in general, clarifying the element base and lower-level architectural features of *associative semantic computers*;

- development of algorithms for optimizing the ways of recording sc-constructions to processor-memory and repositioning already recorded sc-constructions in order to ensure the subsequent efficiency of message transmission between processor elements;
- clarification of the typology of information processes in the processor-memory, their features, and the corresponding typology of labels;
- clarification of the principles of implementing multi-agent knowledge processing within the processor-memory, in particular, the development of principles for implementing event-based information processing in such memory.

ACKNOWLEDGMENT

The author would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help in the work and valuable comments.

REFERENCES

- [1] J. von Neumann, "First draft of a report on the EDVAC," *IEEE Annals of the History of Computing*, vol. 15, no. 4, pp. 27–75, 1993.
- [2] M. D. Godfrey and D. F. Hendry, "The computer as von Neumann planned it," *IEEE Annals of the History of Computing*, vol. 15, no. 1, pp. 11–21, 1993. [Online]. Available: <https://doi.org/10.1109/85.194088>
- [3] V. Glushkov [et al.], *Rekursivnye mashiny i vychislitel'naya tekhnika [Recursive machines and computer engineering]*. IC of the Academy of Sciences of the Ukrainian SSR, 1974.
- [4] J. Aylif, *Principy postroeniya bazovoj mashiny [Principles of building the basic machine]*. Mir, 1973.
- [5] D. Moldovan, W. Lee, C. Lin, and M. Chung, "SNAP: Parallel Processing Applied to AI," *Computer*, pp. 39–49, 1992.
- [6] Y. Chu, "Evolution of computer memory structure," *Proc. National Computer Conf. AFIPS Press*, pp. 733–748, 1976.
- [7] L. Kalinichenko and V. Ryvkin, *Mashiny baz dannyh i znaniy [Database and knowledge machines]*. M.: Nauka, 1990, (In Russ.).
- [8] J. Martin, *Organizatsiya baz dannyh v vychislitel'nyh sistemah: Per. s angl. [Organization of databases in computing systems: Trans. from English]*. Mir, 1980.
- [9] E. Ozkarahan, *Mashiny baz dannyh i upravlenie bazami dannyh [Database machines and database control]*. Mir, 1989.
- [10] T. Kohonen, *Assosiativnaya pamyat': Per. s angl. [Associative memory: Trans. from English]*. Mir, 1980, (In Russ.).
- [11] V. Ignatushchenko, "K postanovke zadachi povysheniya effektivnosti vychislitel'nyh sistem na osnove assosiativnykh metodov obrabotki informatsii [To the formulation of the problem of increasing the efficiency of computing systems based on associative methods of information processing]," *Voprosy kibernetiki. Mnogoprocessornye vychislitel'nye sistemy s perestraivayemoy strukturoj (Arhitektura. Struktura. Primeneniya) [Issues of cybernetics. Multiprocessor computing systems with a tunable structure (Architecture. Structure. Applications)]*, pp. 14–21, 1981.
- [12] S. Berkovich, Yu. Kochin, and V. Molchanov, "Ob effektivnosti primeneniya assosiativnoj pamyati v vychislitel'nyh procedurah [On the effectiveness of using associative memory in computational procedures]," *Vychislitel'nye sistemy. Vyp. 62. [Computing systems. Issue 62.]*, pp. 97–105, 1975.
- [13] M. Aizerman, L. Gusev, S. Petrov, and I. Smirnova, "Dinamicheskij podhod k analizu struktur, opisyyvaemyh grafami (osnovy grafodinamiki) [Dynamic approach to the analysis of structures described by graphs (fundamentals of graph dynamics)]," *Avtomatika i telemekhanika [Automation and telemechanics]*, no. 7/8, pp. 135–151/123–136, 1977.
- [14] G. Marchuk [et al.], *Modul'naya asinhronnaya razvivayushchayasya sistema: V 2-h ch. [Modular asynchronous developing system: In 2 parts]*. Academy of Sciences of the USSR. Sib. department. Computing Center, 1978.
- [15] I. Prangishvili and G. Stetsyura, "Sovremennoe sostoyanie problemy sozdaniya EVM s netradicionnoj strukturoj i arhitekturoj, upravlyaemyh potokom dannyh [Current state of the problem of creating computers with an unconventional structure and architecture controlled by the data flow]," *Izmerenie, kontrol', avtomatizatsiya [Measurement, control, automation]*, no. 1, pp. 36–48, 1981.
- [16] Y. Zatuliver and I. Medvedev, "Voprosy postroeniya i mnogoprocessornoj realizatsii yazyka strukturno-parallel'nogo programirovaniya s upravleniem potokami dannyh [Issues of building and multiprocessor implementation of a structurally parallel programming language with data flow control]," *Voprosy kibernetiki. Mnogoprocessornye vychislitel'nye sistemy s perestraivayemoy strukturoj (Arhitektura. Struktura. Primeneniya) [Issues of cybernetics. Multiprocessor computing systems with a tunable structure (Architecture. Structure. Applications)]*, pp. 123–166, 1981.
- [17] W.B. Ackerman, "Data flow language," *Proc. National Computer Conf. AFIPS Press*, pp. 1087–1095, 1979.
- [18] G. Myers, *Arhitektura sovremennyh EVM: V 2-h kn. Kn. 2. / Per. s angl. [Architecture of modern computers: In 2 books. Book 2. / Transl. from English.]*. Mir, 1985.
- [19] V. Glushkov, "Fundamental'nye issledovaniya i tekhnologiya programirovaniya [Fundamental research and programming technology]," *Programirovanie [Programming]*, no. 2, pp. 3–13, 1980.
- [20] V. Glushkov, S. Pogrebinsky, and Z. Rabinovich, "O razvitiy struktury mul'tiprocessornyh EVM s interpretatsiej yazykov vysokogo urovnya [On the development of multiprocessor computer structures with interpretation of high-level languages]," *Upravlyayushchie mashiny i sistemy [Control machines and systems]*, no. 6, pp. 61–66, 1978.
- [21] Z. Rabinovich, "O koncepcii mashinnogo intellekta [About the concept of machine intelligence]," *Kibernetika i sistemnyj analiz [Cybernetics and system analysis]*, no. 2, pp. 163–173, 1995.
- [22] I. Zadykhailo [et al.], *Proekt assosiativnogo parallel'nogo processora na CMD, orientirovannogo na podderzhku relyatsionnyh baz dannyh [Project of an associative parallel processor on the CMD, focused on the support of relational databases]*. Academy of Sciences of the USSR. Institute of applied mathematics, 1979.
- [23] S. Schuster, H. Nguyen, E. Oskarochan, K. Smith, "RAP.2 – an associative processor for databases and its applications," *IEEE Trans. on Computers*, no. 6, pp. 446–458, 1979.
- [24] E. Suvorov and Ya. Fet, "Processory baz dannyh [Database processors]," *Iss. of the USSR Academy of Sciences. Tech. Cybernet.*, no. 6, pp. 63–75, 1985.
- [25] H.J. Brukle, "High level language oriented hardware and post – von Neumann era," *Proc. 5-th Symp Computer Architecture*, pp. 60–65, 1978.
- [26] Y. Chu, "Architecture of a hardware data interpreter," *Proc. 4-th IEEE Symp. on Computer Architecture*, pp. 1–9, 1977.
- [27] T. Kohonen, *Assosiativnye zapominayushchie ustrojstva: Per. s angl. [Associative storage devices: Trans. from English.]*. Mir, 1982.
- [28] K. Foster, *Assosiativnye parallel'nye processory [Associative parallel processors]*. Energoizdat, 1981.
- [29] A. Ershov, *Algoritmy, matematicheskoe obespechenie i arhitektura mnogoprocessornyh vychislitel'nyh sistem [Algorithms, mathematical support, and architecture of multiprocessor computing systems]*. Nauka, 1982.
- [30] L. Berstein, V. Lisyak, and V. Rabinovich, "Odnorodnaya programiruemyaya struktura dlya resheniya kombinatorno-logicheskikh zadach na grafah i gipergrafah [Homogeneous programmable

- structure for solving combinatorial logic problems on graphs and hypergraphs],” *Metody rascheta i avtomatizaciya proektirovaniya ustrojstv v mikroelektronnyh CVM [Calculation methods and automation of device design in microelectronic digital computers]*, pp. 39–52, 1975.
- [31] V. Vasiliev and E. Raldugin, *Elektronnyye modeli zadach na grafov [Electronic models of graph problems]*. Naukova dumka [Scientific thought], 1987.
- [32] P. Sapaty, “Aktivnoe informacionnoe pole kak model’ strukturnogo resheniya zadach na grafov i setyah [Active information field as a model of structural problem solving on graphs and networks],” *Iss. of the USSR Academy of Sciences. Tech. Cybernet.*, no. 5, pp. 184–208, 1984.
- [33] A. Popov, “Primenenie geterogennoj vychislitel’noj sistemy s naborom komand diskretnoj matematiki dlya resheniya zadach na grafov [Application of a heterogeneous computing system with a set of discrete mathematics commands for solving graph problems],” *Informacionnye tekhnologii [Information technologies]*, vol. 25, no. 11, pp. 682–690, 2019. (In Russ.).
- [34] A. Popov, “Principy organizacii geterogennoj vychislitel’noj sistemy s naborom komand diskretnoj matematiki [Principles of organization of a heterogeneous computing system with a set of discrete mathematics commands],” *Informacionnye tekhnologii [Information technologies]*, vol. 26, no. 2, pp. 67–79, 2020. (In Russ.).
- [35] J. Zhang, S. Khoram, and J. J. Li, “Boosting the Performance of FPGA-based Graph Processor using Hybrid Memory Cube: A Case for Breadth First Search,” *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2017.
- [36] Y. Hu, Y. Du, E. Ustun, and Z. Zhang, “GraphLily: Accelerating Graph Linear Algebra on HBM-Equipped FPGAs,” *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–9, 2021.
- [37] W. S. Song, V. Gleyzer, A. Lomakin, and J. Kepner, “Novel graph processor architecture, prototype system, and results,” in *2016 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, Sep. 2016. [Online]. Available: <https://doi.org/10.1109/hpec.2016.7761635>
- [38] I. V. Afanasyev, V. V. Voevodin, K. Komatsu, and H. Kobayashi, “VGL: a high-performance graph processing framework for the NEC SX-aurora TSUBASA vector architecture,” *The Journal of Supercomputing*, vol. 77, no. 8, pp. 8694–8715, Jan. 2021. [Online]. Available: <https://doi.org/10.1007/s11227-020-03564-9>
- [39] M. Weinzweig, *Obuchayushchayasya sistema iskusstvennogo intellekta s associativnoy pamyat’yu-processorom [Learning Artificial Intelligence system with associative memory processor]*. USSR Academy of Sciences, Scientific Council. according to the complex. probl. “Cybernetics”, 1980.
- [40] M. Weinzweig and M. Polyakova, “Mekhanizm myshleniya i modelirovanie ego raboty v real’nom vremeni [Mechanism of thinking and modeling its work in real time],” *Intellektual’nye processy i ih modelirovanie [Intelligent processes and their modeling]*, pp. 208 – 229, 1987.
- [41] S. Somsabhra, “Reconfigurable semantic processor,” Oct 2006.
- [42] Z. Rabinovich, “Nekotoryj bionicheskij podhod k strukturnomu modelirovaniyu celenapravlenno go myshleniya [Some bionic approach to structural modeling of purposeful thinking],” *Kibernetika [Cybernetics]*, no. 2, pp. 115–118, 1979.
- [43] —, “Razvitie struktur universal’nyh EVM v svyazi s problemami avtomatizacii nauchnyh issledovanij [Development of universal computer structures in connection with the problems of automation of scientific research],” *Avtomatika [Automation]*, no. 5, pp. 63–72, 1979.
- [44] V. Gladun, *Evristsicheskij poisk v slozhnyh sredah [Heuristic search in complex environments]*. Naukova dumka [Scientific thought], 1977.
- [45] —, *Planirovanie reshenij [Planning solutions]*. Naukova dumka [Scientific thought], 1987.
- [46] N. Amosov, A. Kasatkin, L. Kasatkina, and S. Talaev, *Avtomaty i razumnoe povedenie [Automata and reasonable behavior]*. Naukova dumka [Scientific thought], 1973.
- [47] E. Zolotov and I. Kuznetsov, *Rasshiryayushchiesya sistemy aktivnogo dialoga [Expanding active dialogue systems]*. Nauka, 1982.
- [48] A. Galushkin, “Sovremennye napravleniya razvitiya nejrokomput’uternykh tekhnologii v Rossii [Modern directions of development of neurocomputer technologies in Russia],” *Otkrytye sistemy [Open systems]*, no. 4, pp. 25–28, 1997. (In Russ.).
- [49] R. Hecht-Nielsen, “Nejrokomput’uting: istoriya, sostoyanie, perspektivy [Neurocomputing: history, state, prospects],” *Otkrytye sistemy [Open systems]*, no. 4, pp. 23–28, 1998.
- [50] L. Komartsova and A. Maksimov, *Nejrokomput’utery. - 2-e izd. [Neurocomputers. – 2nd ed.]*, ser. Informatika v tekhnicheskoy universitete [Computer Science at the Technical University]. Moscow: Bauman Moscow State Technical University, 2004.
- [51] (2022, Dec) USB Accelerator | Coral. [Online]. Available: <https://coral.ai/products/accelerator/>
- [52] M. Moussa, A. Savich, and S. Areibi, “Architecture, system and method for artificial neural network implementation,” Jun 2013.
- [53] “Nejromorfnyj processor “Altaj” Neuromorphic processor “Altaj”,” mode of access: <https://motivnt.ru/neurochip-altaj/>. — Date of access: 29.03.2023.
- [54] J. D. Allen, J. Philip, and L. Butler, “Parallel machine architecture for production rule systems,” Jun 1989.
- [55] “CUDA Toolkit,” mode of access: <https://developer.nvidia.com/cuda-toolkit>. — Date of access: 29.03.2023.
- [56] “OpenCL,” mode of access: <https://www.khronos.org/opencl/>. — Date of access: 29.03.2023.
- [57] H.-N. Tran and E. Cambria, “A survey of graph processing on graphics processing units,” *The Journal of Supercomputing*, vol. 74, no. 5, pp. 2086–2115, Jan. 2018. [Online]. Available: <https://doi.org/10.1007/s11227-017-2225-1>
- [58] X. Shi, Z. Zheng, Y. Zhou, H. Jin, L. He, B. Liu, and Q.-S. Hua, “Graph processing on GPUs,” *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–35, Nov. 2018. [Online]. Available: <https://doi.org/10.1145/3128571>
- [59] Y. Lü, H. Guo, L. Huang, Q. Yu, L. Shen, N. Xiao, and Z. Wang, “GraphPEG,” *ACM Transactions on Architecture and Code Optimization*, vol. 18, no. 3, pp. 1–24, Sep. 2021. [Online]. Available: <https://doi.org/10.1145/3450440>
- [60] V. Golenkov, “Strukturnaya organizaciya i pererabotka informacii v elektronnyh matematicheskikh mashinah, upravlyayemyh potokom slozhnostrukturirovannykh dannykh [Structural organization and processing of information in electronic mathematical machines controlled by the flow of complex structured data],” PhD diss.: 05.13.01 ; 05.13.13, Minsk, 1996.
- [61] —, *Parallelnyj grafovyj komp’yuter (PGC), orientirovannyj na reshenie zadach iskusstvennogo intellekta, i ego primenenie (Preprint; No 2) [Parallel graph computer (PGC), focused on solving artificial intelligence problems, and its application (Preprint; No. 2)]*. Institute of Technical Cybernetics, 1994.
- [62] V. Golenkov [et al.], *Terminal’nyj modul’ parallelnogo grafovogo komp’yutera (PGC): Interfejs s pol’zovatelem i processornymi modul’yami, struktura, logicheskaya organizaciya: materialy po matematicheskomu obespecheniyu EVM [Terminal module of a parallel graph computer (PGC): User interface and processor modules, structure, logical organization: materials on computer mathematical support]*. Institute of Technical Cybernetics, 1994.
- [63] V. Golenkov, “Grafodinamicheskie modeli i metody parallelnoj asinhronnoj pererabotki informacii v intellektual’nykh sistemah [Graphodynamic models and methods of parallel asynchronous processing of information in intelligent systems],” Doct. diss.: 05.13.11 ; 05.13.17, Minsk, 1996, 396 p.
- [64] P. Gaponov, “Modeli i metody parallelnoj asinhronnoj pererabotki informacii v grafodinamicheskoy associativnoj pamyati [Models and methods of parallel asynchronous processing of information in graphodynamic associative memory],” PhD diss.: 05.13.11, Minsk, 2000, 114 p.
- [65] V. Kuzmitsky, “Principy postroeniya grafodinamicheskogo parallelnogo komp’yutera, orientirovannogo na reshenie zadach iskusstvennogo intellekta [Principles of building a graphodynamic parallel computer focused on solving artificial intelligence problems],” PhD diss.: 05.13.11, Minsk, 2000, 236 p.

- [66] R. Serdyukov, "Bazovye algoritmy i instrumental'nye sredstva obrabotki informacii v grafodinamicheskikh associativnykh mashinakh [Basic algorithms and tools for information processing in graphodynamic associative machines]," PhD diss.: 05.13.11, Minsk, 2004, 114 p.
- [67] V. Golenkov, N. Gulyakina, and D. Shunkevich, *Standart otkrytoy tekhnologii ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheskii sovmestimyyh gibridnyh intellektual'nyh komp'yuternyyh sistem [Standard of the open technology for ontological design, production, and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint, 2021.
- [68] J. von Neumann, *Teoriya samovosproizvodyashchihsvya avtomatov [Theory of self-reproducing automata]*. M.: Mir, 1971, (In Russ.).
- [69] D.A. Moon, "Symbolics architecture," *Computer*, vol. 20, no. 1, pp. 43–52, 1987. [Online]. Available: doi:10.1109/MC.1987.1663356
- [70] S. Smith, "The LMI Lambda Technical Summary. Technical report, LMI Inc." Los Angeles, CA, 1984.
- [71] G. L. Steele and W. D. Hillis, "Connection Machine Lisp: fine-grained parallel symbolic processing," in *Proceedings of the 1986 ACM conference on LISP and functional programming (LFP '86)*. New York: ACM, 1986, pp. 279–297.
- [72] P. McJones. (2018, Dec.) Parallel Lisps: Connection Machine Lisp (StarLisp). Computer History Museum. Mode of access: [https://www.softwarepreservation.org/projects/LISP/parallel#Connection_Machine_\protect\discretionary{\protect\protect\leavevmode@ifvmode\kern+.1667em\relax\OMS/cmsy/m/n/\char2}{\}\Lisp_\(StarLisp\)](https://www.softwarepreservation.org/projects/LISP/parallel#Connection_Machine_\protect\discretionary{\protect\protect\leavevmode@ifvmode\kern+.1667em\relax\OMS/cmsy/m/n/\char2}{\}\Lisp_(StarLisp).). — Date of access: 29.12.2018.
- [73] V. van der Leun, *Introduction to JVM Languages*. Packt Publishing, Jun. 2017.
- [74] V. Ivashenko, "String processing model for knowledge-driven systems," *Doklady BGUIR*, vol. 18, pp. 33–40, 10 2020.
- [75] B. Rasheed and A. Popov, "Network Graph Datastore Using DISC Processor," in *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, Jan. 2019, pp. 1582–1587.
- [76] E. Dubrovin and A. Popov, "Graph representation methods for the discrete mathematics instructions set computer," in *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, Jan. 2020, pp. 1925–1930.
- [77] C. Hewitt. Middle History of Logic Programming: Resolution, Planner, Prolog and the Japanese Fifth Generation Project. Mode of access: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=07A4074D9BBEC56C92085D21A10A6B4D?doi=10.1.1.363.8517&rep=rep1&type=pdf>. — Date of access: accessed 20.03.2020.
- [78] V. Ivashenko, *Modeli resheniya zadach v intellektual'nyh sistemah. V 2 ch. CH. 1 : Formal'nye modeli obrabotki informacii i parallel'nye modeli resheniya zadach : ucheb.-metod. posobie [Problem-solving models in intelligent systems. In 2 p. P. 1 : Formal models of information processing and parallel problem-solving models: study guide]*. Minsk: BSUIR, 2020, (In Russ.).
- [79] "LegUp High-Level Synthesis," mode of access: <http://legup.eecg.utoronto.ca/>. — Date of access: 29.03.2023.
- [80] "VHDPlus," mode of access: <https://vhplus.com/>. — Date of access: 29.03.2023.
- [81] "SystemC Community Portal," mode of access: <https://systemc.org/>. — Date of access: 29.03.2023.
- [82] "MyHDL," mode of access: <https://www.myhdl.org/>. — Date of access: 29.03.2023.
- [83] P. Sapaty, "Yazyk VOLNA-0 kak osnova navigacionnyh struktur dlya baz znaniy na osnove semanticheskikh setej [language VOLNA-0 as the basis of navigation structures for knowledge bases based on semantic networks]," *Iss. of the USSR Academy of Sciences. Technical cybernetics*, no. 5, pp. 198–210, 1986.
- [84] D. I. Moldovan and Y.-W. Tung, "SNAP: A VLSI architecture for artificial intelligence processing," *Journal of Parallel and Distributed Computing*, vol. 2, no. 2, pp. 109–131, 1985. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0743731585900310>
- [85] A. Letichevsky, Yu. Kapitonova, V. Volkov, V. Vyshemirsky, and A. Letichevsky (j.), "Insercionnoe programmirovaniye [Insertion programming]," *Kibernetika i sistemnyy analiz [Cybernetics and system analysis]*, no. 1, pp. 19–32, 2003.
- [86] A. Letichevsky, "Insercionnoe modelirovaniye [Insertion modeling]," *Upravlyayushchie sistemy i mashiny [Control systems and machines]*, no. 6, pp. 3–14, 2012.
- [87] J. Backus, "Can programming be liberated from the von Neumann style?" *Communications of the ACM*, vol. 21, no. 8, pp. 613–641, Aug. 1978. [Online]. Available: <https://doi.org/10.1145/359576.359579>
- [88] V. Kotov and A. Nariniani, "Asinhronnye vychislitel'nye processy nad obshchej pamyat'yu [Asynchronous computing processes over shared memory]," *Kibernetika [Cybernetics]*, no. 3, pp. 64–71, 1966.

Ассоциативные семантические компьютеры для интеллектуальных компьютерных систем нового поколения

Голенков В. В., Шункевич Д. В.,
Гулякина Н. А., Ивашенко В. П.,
Захарьев В. А.

В работе рассмотрены недостатки доминирующей в настоящее время фон-Неймановской архитектуры компьютерных систем в качестве основы для построения интеллектуальных компьютерных систем нового поколения, проведен анализ современных подходов к разработке аппаратных архитектур, устраняющих некоторые из указанных недостатков, обоснована необходимость разработки принципиально новых аппаратных архитектур, представляющих собой аппаратный вариант реализации платформы интерпретации систем, построенных на базе Технологии OSTIS, — ассоциативных семантических компьютеров.

Предложены общие принципы, лежащие в основе ассоциативных семантических компьютеров, рассмотрены три возможных варианта архитектуры таких компьютеров, представлены их достоинства и недостатки.

Received 13.03.2023

Open Semantic Technology as the Foundation for New Generation Intelligent Systems

Mikhail Tatur and Anton Paramonov

Belarusian State University of Informatics and Radioelectronics

Minsk, Republic of Belarus

tatur@bsuir.by, a.paramonov@bsuir.by

Abstract—Issues of new generation intelligent systems development are discussed. A hypothesis is proposed: for the effective implementation of the OSTIS Ecosystem project, it is necessary to create a fundamentally new computer architecture (specialized computers)..

Keywords—OSTIS technology, Ecosystem, Intelligent Systems, Specialized computers, Hardware Platform

I. INTRODUCTION

Artificial Intelligence, like any young scientific direction, develops in leaps and bounds, replacing one generation of technical systems with others. Let's try to present an condition ontology of this direction, which reflects the subjective vision of the authors.

We will make a reservation right away that in this article we will try to use generally accepted statements and trends to avoid unnecessary controversy. We will use the generally accepted understanding of Artificial Intelligence, which stages of evolution are distinguished, at what stage of progress we are now, etc.

We will consider an intelligent system as a product obtained as a result of mathematical apparatus applying, developing algorithms and software and hardware implementation without specifying the application. The mnemonic scheme of the intelligent systems ontology can be represented as three components, as shown in Fig.1. It's no coincidence that in the figure the Mathematical Model (as a paradigm, in a broader sense) intersects with the Algorithms that implement it, and the Hardware Platform intersects with the Software.

In most cases, when creating intelligent systems, they use the mathematical apparatus of Neural Networks and Machine Learning - in Data Processing, as well as Semantic Networks, Inference and Ontologies – in Knowledge Processing (Knowledge Discovery). In addition, Evolutionary Design and Fuzzy Sets are used as extended tools for these paradigms to increase the efficiency of solving intellectual problems.

Remark. In the context of this work, we will not refer to the role of Linguistics, Neurophysiology, and other fundamental disciplines. They, of course, influenced both the formation of Artificial Intelligence, and still influence its development, but are not the topic of the current discussion.

For each mathematical direction, there is a wide range of technological software support to creating intelligence applications (Tensor Flow, Coffee, ScikitLearn Python, Keras, R, etc.). The hardware platform is traditionally presented as “universal” and “specialized”. The universal platform includes conventional computers with GPUs, as well as supercomputers and computing clusters. A specialized platform is mainly neurocomputers and graphic computers (semantic, associative). It's easy to see that specialized “intelligent” computers correspond to the main artificial intelligence mathematical paradigms. If we trace the development chronology of this area of hardware platforms, we will notice obvious surges and falls in the interest of the scientific community and developers. You can also note a wide variety of architectures and technical solutions proposed at different times [1] [2] [3].

As an example, we can highlight one of the latest developments announced by the Moscow State Technical University named after N. E. Bauman – “Leonhard” [4]. As the author's state: “... for the first time in the history of computing, a universal computing system with many instruction streams and one data stream (MISD) has been developed that implements the DISC (Discrete Mathematics Instruction Set computer) instruction set.” There have been quite a few such statements in the past, but none of the specialized «intelligent computers» has found mass industrial use. In our opinion, this is explained by the simple victory of “universal” computers in competition with specialized processors. As a rule, intelligent computing algorithms are highly complex, and therefore require the appropriate performance of the hardware platform. Therefore, throughout the development of the intelligent computing direction, there have been periodic attempts to create specialized (problem-oriented) architectures, the so-called. neurocomputers, graphic computers, semantic computers, associative computers, focused on supporting the corresponding computing paradigm. However, over and over again, special processors were inferior in their characteristics to modern “universal” (multi-core and graphics) computers. In particular, the narrow application focus of specialized computers with large (even huge) development costs in terms of time and

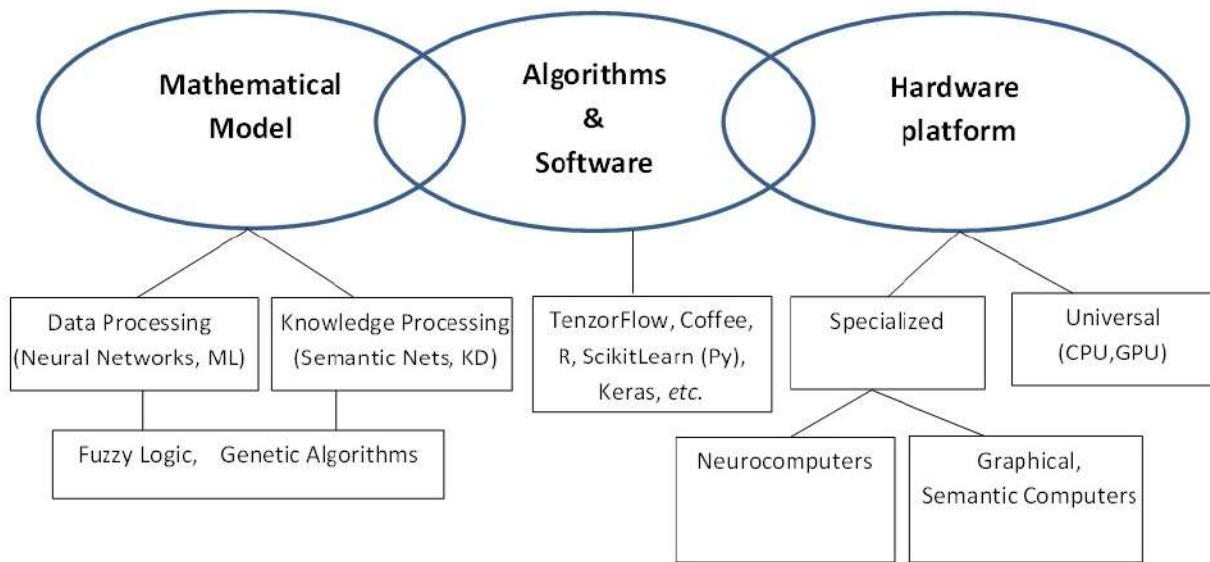


Figure 1. Ontology of intelligent systems

price offset the expected performance gain. In addition, general electronic technologies consistently improve the performance of «universal» computers. In general, a paradoxical situation has occurred in the Artificial Intelligence field. On the one hand, there is the possibility of a quick and relatively inexpensive (using existing libraries and universal software and hardware platforms) creation of highly specialized, commercially successful products with a formal manifestation of intellectual properties. On the other hand, this hinders the development of new (FUNDAMENTALLY NEW) intelligent systems with qualitatively new properties and higher technical characteristics.

The purpose of this presentation is to reflect on what it could be like a New Generation of Intelligent Systems, and what impact OSTIS will have in its emergence.

II. GENERAL TREND IN THE DEVELOPMENT OF ARTIFICIAL INTELLIGENCE TECHNOLOGIES

As a result of a series of industrial and information revolutions, society has moved into a qualitatively new stage of its evolution. Now the information sector occupies a decisive and important position in the context of the development of fundamentally new information technologies aimed at acquiring, processing, and storing knowledge. Currently, UNESCO is promoting the concept of the knowledge society as an antithesis of the concept of the Information Society, the success of which depends primarily on the development of the knowledge-based economy [5]. The indicated trends inevitably lead to digitalization, automation, and finally Digital Transformation of current business processes. It's important to highlight that the use of Artificial Intelli-

gence technologies is one of the fundamental principles of the society digital transformation [6]. Today we can state certain successes and practical achievements in the implementation of commercially successful projects of intelligent computer systems. However, despite the apparent prospects, it can be assumed that with the development of the information society (with a further increase in the volume of information), the approaches underlying such systems will reach the limit of their capabilities, just as it happened before with many computer systems for data processing. Combining many intelligent services into Digital Platforms significantly increases the efficiency of solving a particular range of tasks. This is achieved by Distributing Computations and powers between resources that are available within the platform to all its participants (Fig. 2). The use of an architecture based on autonomous services (also known as Microservices) allows you to speed up the development process and organize the formation of various platform configurations. At the same time, the Knowledge Base of each such service is built autonomously and is often encapsulated within its implementation. This gives rise to multiple duplication of information and redundancy in the knowledge processing. In this connection, it should be noted that the problematic issues of metadata and ontologies of the modern information society are strongly connected with the problems of "conceptual plurality of information" and "data representation in subject areas of knowledge".

However, the most promising approach to building AI applications is the collaboration of intelligent systems, which forms a single complete ecosystem. This solution is projected to solve a wide class of problems in

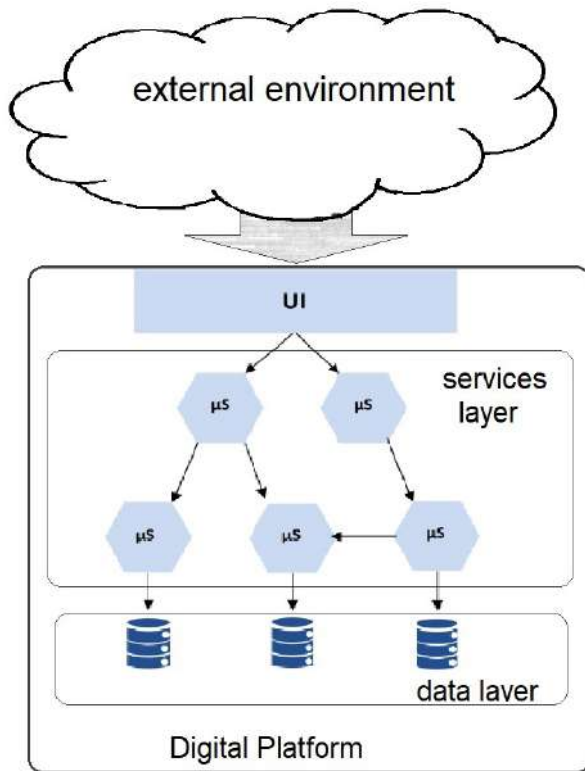


Figure 2. Generalized scheme of the Intelligent Services Platform

various sectors of the economy. Based on the Systems Theory by Ludwig von Bertalanffy, an ecosystem is commonly understood as «a complex open self-organizing, self-regulating and self-developing system». A Digital Ecosystem is often understood as a single common space in which many different services operate seamlessly, both from one company and several partner participants. Integration between them allows you to achieve maximum speed and transparency of processes, detect problems and improvement points in different business areas. An Intellectual Ecosystem implies a qualitative change in approaches to the creation of intelligent computer systems. The main goal is the formation of a common information space in the form of a general knowledge base and a general mechanism for the semantic compatibility of all intelligent agents of this system. Figure 3 shows the general principle of building an Intelligent Ecosystem.

The transition from data processing systems to knowledge management systems involves conceptually new approaches not only and not so much in mathematical and algorithmic support, but to a greater extent in the creation of hardware platforms. Various ideas have been put forward for a long time to achieve high real performance of a computing system by adapting its architecture to the structure of the problem being solved and creating a computing device that equally effectively implements both structural and procedural fragments of calculations

[7] [8]. However, as a promising and comprehensive solution, the construction of new generation intelligent computers is considered. They will make it possible to remove a few issues that arise due to the technical limitations of existing computer architectures and bring the development of intelligent systems to a qualitatively new level.

III. TOWARDS A NEW GENERATION OF INTELLIGENT COMPUTERS

Let's define a new generation of intelligent systems through a description of their properties, distinguishing features and expected technical characteristics..

What can be expected from innovative mathematical models?

In terms of functional capabilities, intelligent systems should exhibit some (not necessarily all) qualitatively new properties, such as:

- semantic clustering;
- convergence (gradual blurring of the boundaries between data processing and knowledge processing);
- automatic knowledge acquisition;
- processing of fuzzy knowledge;
- generation (recreation) of images from patterns;
- modeling of cognitive functions and phenomena and so on.

In terms of applications, innovative mathematical models should be more versatile compared to neural and semantic networks.

In terms of technical characteristics, intelligent systems should provide:

- reduction of labor costs for the construction, operation, and development of new applied intelligent systems;
- increase in the level of intelligence (cognitivism) compared to known intelligent systems.

In terms of infrastructure, mathematical models and language tools for representing images and knowledge will be standardized and ensure cross-platform compatibility, meaning that high-level models, algorithms, and software will be portable across specialized, problem-oriented, and universal computers.

What to expect from specialized hardware platforms?

In terms of architecture, intelligent computers should support the infrastructure of mathematical models, possibly incorporating features of both neural and semantic computations.

In terms of technical specifications, specialized computers may have performance that exceeds that of general-purpose computers by several orders of magnitude.

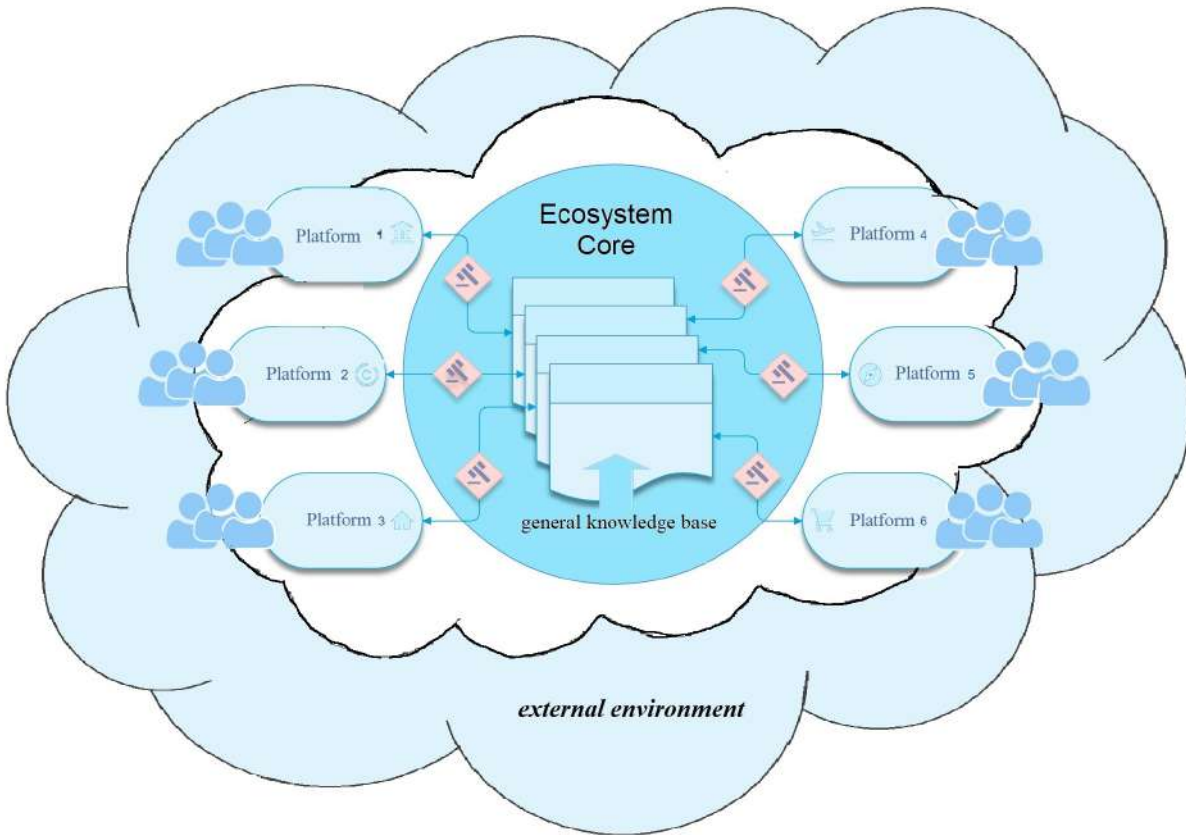


Figure 3. Generalized scheme of the Intellectual Ecosystem

In terms of the component base, modern electronic technologies, primarily FPGA and GPU, allow for the development and rapid prototyping of original technical solutions for innovative architectures of intelligent computers.

In terms of the structure and technical solutions topology, the architectures of modern neural networks, including convolutional ones, are tree-like schemes with re-convergent irregular connections, often with non-homogeneous functions in processor elements. The architectures of modern semantic networks also represent graph schemes with irregular connections and non-homogeneous functions of processor elements. Attempts to structurally implement such architectures "bluntly" usually encounter technological limitations of the component base and do not have any significant effect. Therefore, in both cases, developers strive to find "workarounds." One of the most common solutions is to map the irregular connections of architectures onto a structure with a regular topology of connections.

The simplest examples can be a bipartite graph, which is reduced to a technical solution in the form of a line of processor elements (Fig. 4), as well as lattice graphs, which are reduced to a technical solution in the form of

a matrix of processor elements [2] [3].

In more complex implementation variants, one should expect more complex variants of mapping the architecture onto the topology of processor elements connections.

In terms of profitability, specialized computers may find their niche by unifying a wide range of intelligent computations and integrating into the overall infrastructure of artificial intelligence.

IV. OSTIS IN THE CONTEXT OF AMBITIOUS OBJECTIVES

Over the past 10-15 years, as OSTIS began "to take shape", it has primarily been viewed as a tool for building applied intelligent systems. Examples of this include an information service system for employees and a decision-making module for the quality control system of PJC "Savushkin Product", a set of prototypes for intelligent learning systems in various disciplines of secondary and higher education, and intelligent dialogue systems for medical and educational purposes implemented on universal software and hardware platforms. Currently, OSTIS can be considered as an independent infrastructure [9], within which all prerequisites exist for the creation of innovative specialized processors. These specialized

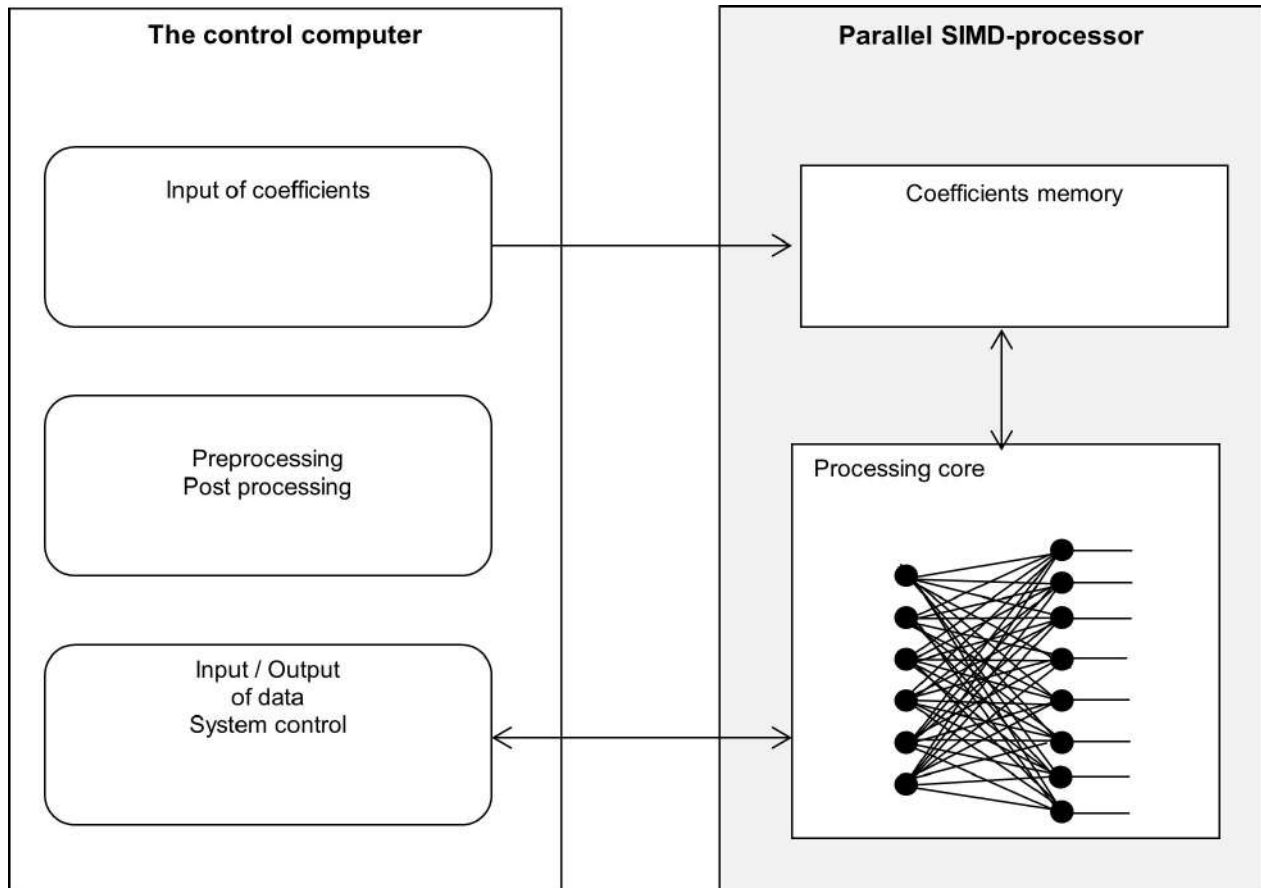


Figure 4. Simplified scheme of a computing system with a parallel processor core

processors within OSTIS are viewed as an option for interpreting computer system models developed based on OSTIS platforms [10]. One of the principles of OSTIS is to ensure platform independence of computer systems (OSTIS systems) developed on its basis. In other words, the development of such a system is reduced to the development of its model (including functional modules for solving tasks), described by means of a knowledge representation language and the subsequent loading of this model onto an OSTIS platform that meets certain requirements. Figure 5 illustrates a fragment of the OSTIS ecosystem, which is a comprehensive infrastructure for the interaction of OSTIS systems, users, and other computer information systems. As shown in the figure, specialized (associative semantic computers) developed under the infrastructure conditions will directly support distributed knowledge processing, significantly increasing their efficiency compared to universal computers.

Thus, in general, an OSTIS platform can be implemented either in hardware (e.g., as a specialized processor) or in software (e.g., as a virtual machine based on modern von Neumann computers). At the same time, there can be many implementation options for both hardware and software OSTIS platforms, and each such

option may have its own advantages and disadvantages, considering the classes of tasks solved by specific OSTIS systems.

REFERENCES

- [1] A. I. Galushkin Supercomputers and Neurocomputers. Neurocomputers: design and applications, 2000, No. 6, Begel House Publ.
- [2] N. Verenik, Y. Seitkulov, A. Girel, M. Tatur Some regularities and objective limitations of implementing semantic processing algorithms on computing systems with massive parallelism. *Eurasian Journal of Mathematical and Computer Application*, 2014, Vol. 2, Issue 2, pp. 92–101.
- [3] M. Tatur Problem-Oriented Processors for the Solving of Classification Tasks. *Journal of Information, Control and Management Systems*, Slovakia, 2013, Vol. 11, No. 2, pp. 155–164.
- [4] A. Yu. Popov, Printsipial'no novyi protsessor dlya obrabotki mnozhestv, struktur dannykh i grafov [Fundamentally new processor for processing sets, data structures and graphs]. Available at: <https://elearning.bmstu.ru/iu6/mod/page/view.php?id=1690> (accessed 2021)
- [5] Towards knowledge societies: UNESCO world report. Paris: UNESCO Publishing, 2005, P. 226.
- [6] A. V. Tuzikov Tsifrovaya transformatsiya. Osnovnye ponyatiya i terminologiya [Digital transformation. Basic concepts and terminology]. *Belaruskaya navuka*, Minsk, 2020, P. 267
- [7] L. Chernyak Arkhitektura fon Neimana, rekonfiguriruemye komp'yuternye sistemy i antimashina [Von Neumann architecture, reconfigurable computer systems and anti-machine]. *Otkrytye sistemy [Open Systems]*, Moscow, 2008, Vol. 6.

OSTIS Ecosystem Infrastructure

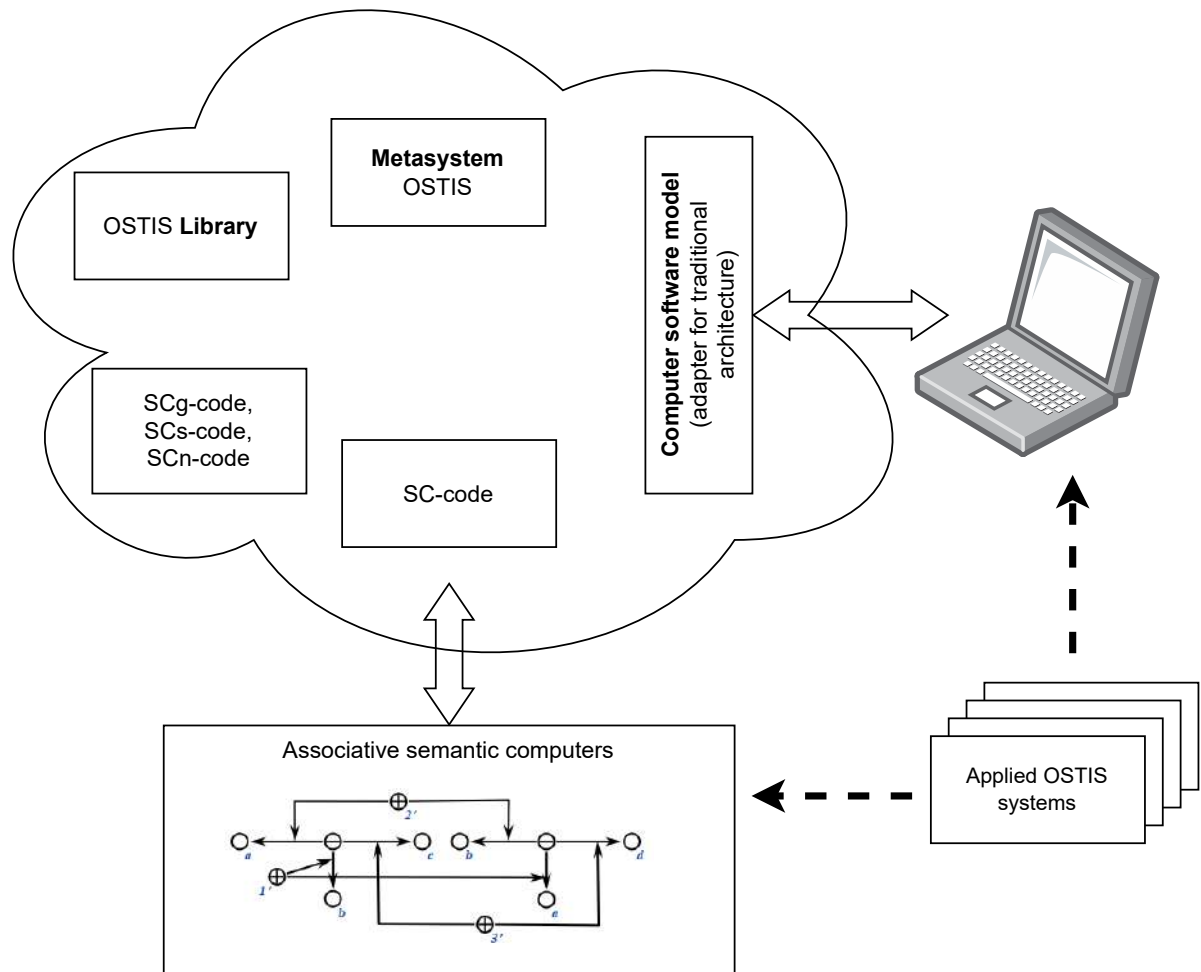


Figure 5. The OSTIS ecosystem and specialized computers place in it

- [8] I. I. Levin, A. I. Dordopulo, I. A. Kalyaev, Yu. I. Doronchenko, M. K. Raskladkin *Sovremennye i perspektivnye vysokoproizvoditel'nye vychislitel'nye sistemy s rekonfiguriruemoi arkhitekturoi* [Modern and perspective high-performance computing systems with reconfigurable architecture]. *Vestnik YuUrGU [Bulletin of the South Ural State University]*, Chelyabinsk, Ser. Comp. Math. and Software Eng., 2015, Vol. 5, No. 3. pp. 24–39.
- [9] V. Golenkov, N. Guliakina, D. Shunkevich, *Otkrytaya tekhnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheski sovmestimyykh gibridnykh intellektual'nykh komp'yuternyykh sistem* [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems], Minsk, Bestprint, 2021, P. 690.
- [10] D. Shunkevich *Universal'naya model' interpretatsii logikosemanticheskikh modelei intellektual'nykh komp'yuternyykh sistem novogo pokoleniya* [Universal model of interpreting logicsemantic models of intelligent computer systems of a new generation]. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem* [Open semantic technologies for intelligent systems], 2022, pp. 285–296.

Открытая семантическая технология как фундамент нового поколения интеллектуальных систем

Татур М. М., Парамонов А. И.

Рассматриваются вопросы развития нового поколения интеллектуальных систем. Приводятся примеры известных практических экспериментов по созданию специализированных платформ для достижения высокой реальной производительности вычислительной системы. Выдвигается гипотеза о необходимости создания принципиально новой архитектуры вычислительной машины для эффективной реализации проекта экосистемы ostis-платформ. Предлагается концепт инфраструктуры OSTIS экосистемы на базе специализированных компьютеров.

Received 13.03.2023

Design Principles, Structure, and Development Prospects of the Software Platform of ostis-systems

Nikita Zotov

*Belarusian State University of
Informatics and Radioelectronics*

Minsk, Belarus

Email: nikita.zotov.belarus@gmail.com

Abstract—In the article, the principles of design and development of the basic *Software implementation of the ostis-platform* are described. The advantages of the ontological approach to documenting software systems of this type are shown. The structure, problems, and prospects of developing the *Software implementation of the ostis-platform* are described.

Keywords—ontological design, automation tools for design and development of computer systems, knowledge base management system, universal interpreter, graph storage, ostis-platform

I. INTRODUCTION

Modern *software computer systems* should operate not just with *data* but with *knowledge*. To understand the *meaning* of knowledge, it is necessary to represent this *knowledge* in an *understandable* form for *any cybernetic system*: as for any *human*, so for any *artificial system* [1]. At the same time, the form of representing this knowledge must be *unified* and *independent* of the platform on which this knowledge can be interpreted. Nevertheless, computer systems remain dependent on highly qualified specialists and experts in subject domains in which the automation of the design of these systems is carried out; therefore, the implementation of these systems requires significant resources [2]. One of the reasons for this is the need for a computer system to work on different platforms, each of which, in general, may have its own characteristics and limitations, which must be taken into account at the implementation stage.

The solution of these problems is **design and development of fundamentally new platforms**, which should provide:

- *unambiguity of interpretation* and representation of software system models provided by the unified knowledge representation language and platform design ontology used;
- *semantic compatibility* of software system models and their components [3], including interoperability between them [4];

- *platform independence* of software system models implemented and interpreted on it;
- *simplicity* and *extensibility* of their functionality;
- *functional completeness* for creating software system models due to the presence of a formal methodology for designing its implementation;
- segregation of duties between platform components.

II. PROPOSED APPROACH TO THE DESIGN OF SYSTEMS FOR AUTOMATING THE DESIGN OF SOFTWARE SYSTEMS

The shortcomings of modern computer systems for design automation of other software systems, ways to solve them, as well as the approach to the solution described below were described earlier in the works [5] and [6].

Despite the vast variety of classical technologies used by mankind, there is no general solution that allows solving the problem in a complex. At the moment, the described problems can only be solved with the help of a general and universal solution — **OSTIS Technology** [7]. The *OSTIS Technology* is based on a unified variant of information encoding based on *semantic networks* with a basic set-theoretic interpretation, called *SC-code* [8]. The language of semantic representation of knowledge is based on two formalisms of discrete mathematics: *Set Theory* — defines the semantics of the language — and *Graph Theory* — defines the syntax of the language. Any types and models of knowledge can be described using *SC-code* [7].

One of the *key* principles of the *OSTIS Technology* [9] is providing **platform independence** of *ostis-systems* [10], i.e. strict separation of the *logical-semantic model of a cybernetic system* (*sc-model of a cybernetic system*) and the *platform for interpreting sc-models of a cybernetic system* (*ostis-platform*). The advantages of such a strict separation are quite obvious:

- transfer of the *ostis-system* from one *ostis-platform* to another is carried out with *minimum* overhead

costs (in the ideal case — comes down to simply loading the *sc-model of a cybernetic system* onto the ostis-platform);

- components of *ostis-systems* become universal, that is, can be used in any ostis-systems where their use is appropriate;
- the development of the ostis-platform and the development of *sc-models* of systems can be carried out in parallel and independently of each other, in the general case by separate independent development teams according to their own rules and methods [11].

logical-semantic model of a cybernetic system

:= [formal model (formal description) for the functioning of a cybernetic system, consisting of (1) a formal model of information stored in the memory of a cybernetic system and (2) a formal model of a collective of agents that process this information.]

⊃ *sc-model of a cybernetic system*

:= [logical-semantic model of a cybernetic system represented in the SC-code]

:= [logical-semantic model of an ostis-system, which, in particular, can be a functionally equivalent model of any cybernetic system that is not an ostis-system]

ostis-system

⊂ *subject*

⇒ *generalized decomposition**:

- *sc-model of a cybernetic system*
- *ostis-platform*

sc-model of a cybernetic system

⇒ *generalized decomposition**:

- *sc-memory*
- *sc-model of the knowledge base*
- *sc-model of the problem solver*
- *sc-model of a cybernetic system interface*

ostis-platform

:= [platform for interpreting *sc-models* of computer systems]

:= [interpreter for *sc-models* of cybernetic systems]

:= [interpreter of unified logical-semantic models of computer systems]

:= [family of platforms for interpreting *sc-models* of computer systems]

:= [platform for implementing *sc-models* of computer systems]

:= ["empty" ostis-system]

:= [sc-machine implementation]

⊂ *platform-dependent reusable ostis-systems component* [11]

sc-memory

:= [abstract *sc-memory*]

:= [sc-storage]

:= [semantic memory storing SC-code constructions]

:= [storage of SC-code constructions]

In general, *sc-memory* implements the following functions:

- storage of *SC-code* constructions;
- storage of information constructions (files) external to *SC-code*. In general, file storage can be implemented differently from storage of *sc-constructions*;
- access (reading, creating, deleting) to *SC-code* constructions, implemented through the corresponding *software (hardware) interface*. Such an interface is essentially a *microprogramming language* that makes it possible to implement on its basis more complex procedures for processing stored constructions, the set of which essentially determines the list of commands of such a *microprogramming language*. The *sc-memory* itself is passive in this regard and simply executes commands initiated from outside by some subjects.

Despite all the advantages of *graph databases* in comparison with *relational databases* [12], [13], *new generation computer software systems*, due to their properties, [14] should operate not simply with *data*, but *knowledge*. To understand the *meaning* of knowledge, it is necessary to represent this knowledge in an understandable form for any kind of *cybernetic system* [14]. Speaking about the unification of the representation of all *types of knowledge*, it is considered important to use *graph databases* not just as a means for storing *structured data*, but for storing *semantically coherent* and *related* knowledge among themselves. Therefore, *sc-memory* is based on a graph representation of data and knowledge.

III. PRINCIPLES UNDERLYING THE SOFTWARE PLATFORM OF OSTIS-SYSTEMS

The specification of such a complex program object as the *ostis-platform* must be represented in some formal knowledge representation language, in this case, in the *SC-code*, the texts of which it stores and processes. The language that should describe the *Software implementation of the ostis-platform* should be a *sublanguage** of the *SC-code*, i.e. it should inherit all the properties of the *Syntax* and *Denotational semantics of the SC-code* [15]. This representation of *software computer systems* specifications gives certainly strong advantages over other possible representations of specifications [16]:

- The language whose texts the system stores and processes and the language that specifies how the system represents the texts of the first language in its own memory are subsets of the same language. This simplifies not only the understanding of the developer who develops a complex *software computer system*, due to the fact that the form of representation of the language processed by this system and the

language of its specification are unified, but also allows discovering new functionalities for this system in cognition of itself. Thus, this approach allows full implementation of *intelligent computer system* properties, for example, *reflexivity*.

- It is impossible to design and implement *intelligent computer systems* on a *software computer system* that is not itself one. Representing the system specification in this form allows significantly increasing the level of its *intelligence* [14].
- There is no need to create additional tools for verification and analysis of the operation of the entire system, since the representation form of the system description language is unified with the language whose texts it stores and processes. This allows not only reducing the number of tools used in the design and implementation of the *ostis-platform* but also allows unifying the information stored in the *ostis-platform* and describing the *ostis-platform* with the purpose of using this information in the evolution of *ostis-platform* components. At the same time, the *ostis-platform* specification remains *platform-independent*, so when changing one implementation of the *ostis-platform* to another, an approach to describing the *ostis-platform* remains the same.

Software implementation of the ostis-platform

- := [Implementation of the sc-machine]
- ⇒ *frequently used sc-identifier**:
- [Software platform of ostis-systems]
- := [Basic software platform for mass creation of next-generation intelligent computer systems]
- := [Our proposed software implementation of an associative semantic computer]
- := [sc-machine]
- ∈ *specialized ostis-platform*
- ∈ *web-based implementation of the ostis-platform*
- := [an option of implementing a platform for interpreting sc-models of computer systems, involving the interaction of users with the system via the Internet]
- ∈ *multi-user ostis-platform implementation*
- ∈ *reusable ostis-systems component stored as source files*
- ∈ *non-atomic reusable ostis-systems component*
- ∈ *dependent reusable ostis-systems component*
- ⇒ *component address**:
- [<https://github.com/ostis-ai/sc-machine>]
- ⇒ *software system decomposition**:
- {
 - *Implementation of memory in the ostis-platform*
 - *Implementation of the subsystem of interaction with the external environment using languages of network interaction*
 - *Implementation of the interpreter for*

- *sc-models of user interfaces*
- *Implementation of the basic set of platform-specific sc-agents and their common components*
- *Implementation of the manager of ostis-systems reusable components*

}

⇒ *component dependencies**:

- {
 - *Implementation of memory of the ostis-platform*
 - *Implementation of the subsystem of interaction with the external environment using languages of network interaction*
 - *Implementation of the interpreter for sc-models of user interfaces*

}

Software implementation of the ostis-platform

⇒ *underlying principles**:

- The current *Software implementation of the ostis-platform* is web-oriented, so from this point of view, each ostis-system is a web site accessible online through the usual browser. This implementation option has an obvious advantage — access to the system is possible from anywhere in the world where the Internet is available, and no specialized software is required to work with the system. On the other hand, this implementation option allows multiple users to work with the system in parallel.
- The implementation is cross-platform and can be built from source texts on various *operating systems*. At the same time, the interaction between the client and server parts is organized in such a way that a web-interface can be easily replaced with a desktop or mobile interface, both universal and specialized ones.
- The current *Software implementation of the ostis-platform* is customized, i.e. does not include the *Implementation of the SCP Language interpreter*. At the current stage of development of the *Software implementation of the ostis-platform*, all functioning *ostis-systems* are *platform-dependent*. This problem is primarily related to the shortcomings of the chosen and implemented short-memory access control model, which does not allow fully creating distributed collectives of *sc-agents* working on *sc-memory*.
- The Core of the platform is the *Implementation of memory of the ostis-platform*, which can simultaneously interact with both *Implementation of the interpreter for sc-models of user interfaces* and with any third-party applications according to the corresponding languages of

network interaction (network protocols). From the point of view of the overall architecture, *Implementation of the interpreter for sc-models of user interfaces* acts as one of many possible external components that interact with the *Implementation of memory of the ostis-platform* over the network. From the point of view of the overall architecture, *Implementation of the interpreter for sc-models of user interfaces* acts as one of many possible external components that interact with the *Implementation of memory of the ostis-platform* over the network. The current *Implementation of the interpreter for sc-models of user interfaces of ostis-systems in the Software implementation of the ostis-platform* is *platform-dependent*, since the interpreter of the basic *SCP Language* [17] is not fully implemented.

- The current *Implementation of memory in the ostis-platform* allows storing and representing *sc-constructions* that describe any *sc-model of the ostis-system*, external *information constructions*, not belonging to the *SC-code*, as well as providing different levels of access for processing these constructions. In the context of this *Software implementation of the ostis-platform*, *Implementation of memory in the ostis-platform* consists of such components as: *Implementation of ostis-platform sc-memory*, inside which *sc-constructions for sc-models of ostis-systems* are represented, *Implementation of ostis-platform file memory*, inside which external *information constructions* are represented that do not belong to the *SC-code*, i.e. the contents of *ostis-system internal files*, but additionally describe, explain, and detail *sc-constructions for sc-models of ostis-systems*.
- Current *Software implementation of the ostis-platform* includes *Implementation of the manager of reusable ostis-systems components*. This is connected with the fact that the current *Implementation of the manager of reusable ostis-systems component* uses *Implementation of memory of the ostis-platform* to store and process the specification of installed components, regardless of their implementation language.

Principles underlying the Software implementation of the ostis-platform are only basic, all components included in the *Software implementation of the ostis-platform* have their own implementation features, as well as analogues that must be taken into account when implementing the entire *ostis-platform*.

IV. PRINCIPLES OF DOCUMENTING THE SOFTWARE PLATFORM OF OSTIS-SYSTEMS

Permanent reengineering of the components of the current *Software implementation of the ostis-platform* is provided by an open team of developers, while each component being developed is documented according to generally accepted principles.

Software implementation of the ostis-platform

⇒ *documentation principles**:

- Regardless of the implementation language of each *Software implementation of the ostis-platform* component, the specification of each component includes a specification directly described in the source files of the component itself, describing the programming interface of this component, as well as a specification as part of the *ostis-platform knowledge base*, describing in detail the implementation of this component, including the algorithms used. At the same time, duplication in the specification for the *Software implementation of the ostis-platform* components is strictly prohibited. So, for example, the specification, which is directly located in the source file with the implementation of the components themselves, describes the features of using the components from the point of view of an external or internal (that is, being part of the team) developer, and the specification, which is part of the *sc-text* for the knowledge base of the *Software implementation of the ostis-platform*, additionally includes features, proposed approaches to implementation, as well as the advantages and disadvantages of the components included in the composition.
- Each component of the *Software implementation of the ostis-platform* is described by means of the *OSTIS Technology*, that is, in the *SC-code*, the texts of which it processes and stores. Thus, it enables the platform to analyze its state and help maintain its life cycle without the participation of its developers. *Software implementation of the ostis-platform* acts as a full-fledged subject that is directly involved in its own development.
- *Specification of the Software implementation of the ostis-platform* is an *sc-language*, i.e. a sublanguage of the *SC-code*, for which the *Syntax* and *Denotational semantics of the SC-code* are specified. This *sc-language* can be represented as a family of more specific *sc-languages* that allow describing:
 - how *sc-constructions* are represented inside the *ostis-platform sc-memory*;

- how *information constructions* that do not belong to the *SC-code* are represented within the *file memory of the ostis-platform*;
- how different *ostis-platform* subsystems interact with each other;
- which methods and their corresponding agents interact with the *ostis-platform sc-memory*;
- how various interpreters for *sc-models of ostis-systems* (knowledge base, solver, interface) are represented and work;
- and so on.

This approach makes it possible to integrate descriptions of various components [18] that are part of the *Software implementation of the ostis-platform* without any particular obstacles, since the entire *Specification of the Software implementation of the ostis-platform* is its knowledge base with a clearly defined hierarchy of subject domains and ontologies (that is, *sc-languages* that describe its implementation).

- Each developer of the *Software implementation of the ostis-platform* takes care of the permanent support of not only the state of its components but also the specification of these components. A quality of *Software implementation of the ostis-platform* is guaranteed by its team of developers who are able not only to understand the implementation details of the *ostis-platform* but also to contribute to the creation of mutually beneficial cooperation to achieve the set goals.

These principles can be used to describe any other *software computer systems*, including those *software computer systems* that are not implemented on this *ostis-platform*.

V. STRUCTURE OF THE SOFTWARE PLATFORM OF OSTIS-SYSTEMS

Implementation of memory in the ostis-platform

- := [Implementation of ostis-platform sc-memory and file memory]
- := [Our proposed software implementation of ostis-platform sc-memory and file memory]
- ∈ *reusable ostis-systems component stored as source files [11]*
- ∈ *non-atomic reusable ostis-systems component*
- ∈ *dependent reusable ostis-systems component*
- ⇒ *component address**:
[<https://github.com/ostis-ai/sc-machine/tree/main/sc-memory>]
- ⇒ *software system decomposition**:
{
 - *Implementation of ostis-platform sc-memory*

- *Implementation of ostis-platform file memory*
- }
⇒ *component dependencies**:
{
 - *GLib library of methods and data structures*
 - *C++ Standard Library of methods and data structures*
 - *Implementation of ostis-platform sc-memory*
 - *Implementation of ostis-platform file memory*
}

Implementation of ostis-platform sc-memory

- := [Software implementation of graphodynamic associative memory in the Software ostis-systems platform]
- := [Our proposed implementation of graphodynamic associative memory for ostis-systems]
- ∈ *sc-memory implementation*
- ∈ *reusable ostis-systems component stored as source files*
- ∈ *atomic reusable ostis-systems component*
- ∈ *dependent reusable ostis-systems component*
- ⇐ *software model**:
sc-memory
- ⇐ *family of subsets**:
sc-memory segment
- := [page of sc-memory]
- ⇐ *family of subsets**:
sc-memory element
- ⇒ *component dependencies**:
{
 - *GLib library of methods and data structures*
 - *C++ Standard Library of methods and data structures*
- }
⇒ *programming language used**:
 - *C*
 - *C++*
- ⇒ *internal language**:
 - *SCin-code*

In general, *sc-memory* can be implemented in different ways. So, for example, another version of *ostis-platform sc-memory* can be implemented using the software implementation of the *Neo4j Platform* [19]. The difference between this possible implementation of *sc-memory* and the current one is that the storage of *graph constructions* and the management of the flow of actions on them should be carried out to a greater extent by means provided by the *Neo4j Platform*; at the same time, the representation of *graph constructions* must be implemented in its own way, since it depends on the *Syntax of the SC-code*.

Such an sc-memory model can be easily described in the *sc-language*, that is, in the sublanguage of the *SC-code*. Such a language allows describing how texts of a language are represented inside the memory of the *ostis-platform* in the same language. At the same time, not only the unification of representing information processed by the *ostis-platform* and information describing the *ostis-platform* itself is observed, but also opportunities are given for expanding and using the language in the process of evolution of the *ostis-platform* and its components, including those in the process of evolution of *Implementation of ostis-platform sc-memory*.

SCin-code

:= [Semantic Code interior]
 := [Language for describing the representation of the SC-code inside ostis-platform sc-memory]
 := [Metalanguage for describing the representation of sc-constructions in ostis-platform sc-memory]
 ⇒ *frequently used non-primary external identifier of the sc-element**:
 [scin-text]
 ∈ *common noun*
 ∈ *abstract language*
 ∈ *metalanguage*
 ∈ *sc-language*
 ⊂ *SC-code*
 ⊃ *sc-memory*

should be distinguished*

⊃ { • *SC-code*
 := [Universal language of internal semantic representation of knowledge in memory of ostis-systems]
 • *SCin-code*
 := [Metalanguage for describing the representation of the SC-code in ostis-platform sc-memory]
 ⊂ *SC-code*
 }

Software interface of Implementation of ostis-platform sc-memory

⇐ *software interface**:
Implementation of ostis-platform sc-memory
 ∈ *software interface*
 ∈ *reusable ostis-systems component stored as source files*
 ∈ *atomic reusable ostis-systems component*
 ∈ *dependent reusable ostis-systems component*
 ⇒ *component dependencies**:
 { • *GLib library of methods and data structures*
 • *C++ Standard Library of methods and data structures*
 }

⇒ *method representation language used**:
 • *C*
 • *C++*
 ⊃ *Software interface for information-forming methods of Implementation of ostis-platform sc-memory*
 := [information-forming methods of Implementation of ostis-platform sc-memory]
 := [subsystem that is part of the implementation of ostis-platform sc-memory, which allows creating, modifying, and deleting constructions of sc-memory]
 ⇐ *software interface**:
Implementation of the information-generating subsystem of Implementation of ostis-platform sc-memory
 ⊂ *Implementation of ostis-platform sc-memory*
 ⊃ *Software interface for information retrieval methods of Implementation of ostis-platform sc-memory*
 := [information retrieval methods of Implementation of ostis-platform sc-memory]
 := [subsystem that is part of Implementation of ostis-platform sc-memory that allows finding constructions in sc-memory]
 ⇐ *software interface**:
Implementation of the information retrieval subsystem of Implementation of ostis-platform sc-memory
 ⊂ *Implementation of ostis-platform sc-memory*

Implementation of ostis-platform file memory

∈ *file memory implementation based on the prefix tree*
 ⇐ *software model**:
ostis-platform file memory
 ∈ *reusable ostis-systems component stored as source files*
 ∈ *atomic reusable ostis-systems component*
 ∈ *dependent reusable ostis-systems component*
 ⇒ *component dependencies**:
 { • *GLib library of methods and data structures*
 }
 ⇒ *method representation language**:
 • *C*
 ⇒ *internal language**:
 • *SCfin code*

SCfin-code

:= [Semantic Code file interior]

:= [Language for describing the representation of information constructions that do not belong to the SC-code inside the ostis-platform file memory]
 := [Metalanguage for describing the representation of information constructions that do not belong to the SC-code inside the ostis-platform file memory]
 ⇒ *frequently used sc-identifier**:
 [sc.fin-text]
 ∈ *common noun*
 ∈ *abstract language*
 ∈ *metalanguage*
 ∈ *sc-language*
 ⊂ *SC-code*
 ⊃ *ostis-platform file memory*

should be distinguished*

⊃ {• *SC-code*
 := [Universal language of internal semantic representation of knowledge in memory of ostis-systems]
 • *SCfin-code*
 := [Metalanguage for describing the representation of external information constructions that do not belong to the SC-code in ostis-platform file memory]
 ⊂ *SC-code*
 }

should be distinguished*

⊃ {• *SCin-code*
 := [Metalanguage for describing the representation of the SC-code in ostis-platform sc-memory]
 ⊂ *SC-code*
 • *SCfin-code*
 := [Metalanguage for describing the representation of external information constructions that do not belong to the SC-code in ostis-platform file memory]
 ⊂ *SC-code*
 }

Implementation of the subsystem of interaction with the external environment using languages of network interaction

⇒ *software system decomposition**:
 {• *Implementation of the subsystem of interaction with the external environment using languages of network interaction based on the JSON language*
 }

Implementation of the network interaction subsystem with sc-memory based on JSON in the ostis-platform

:= [Subsystem for interacting with sc-memory based on the JSON format]
 := [Network software interface of Implementation of ostis-platform sc-memory]
 := [Our proposed option of implementing the mechanism for accessing the ostis-platform sc-memory in a distributed collective of ostis-systems]
 ∈ *reusable ostis-systems component stored as source files*
 ∈ *non-atomic reusable ostis-systems component*
 ∈ *dependent reusable ostis-systems component*
 ∈ *client-server system*
 ⇒ *method representation language used**:
 • *C*
 • *C++*
 • *Python*
 • *TypeScript*
 • *C#*
 • *Java*
 ⇒ *language used**:
 • *SC-JSON-code*
 ⇒ *software system decomposition**:
 {• *Implementation of the Server System based on Websocket and JSON, providing network access to memory of the ostis-platform*
 }
 = {• *Implementation of the client system in the Python programming language*
 • *Implementation of the client system in the TypeScript programming language*
 • *Implementation of the client system in the C programming language*
 • *Implementation of the client system in the Java programming language*
 }
 }

SC-JSON-code

:= [Semantic JSON-code]
 := [Semantic JavaScript Object Notation code]
 := [Metalanguage for describing the representation of messages between subsystems of the ostis-platform]
 ⇒ *frequently used sc-id**:
 [sc.json-text]
 := [The language we propose for interaction in a distributed collective of ostis-systems]
 ∈ *common noun*

- ∈ *abstract language*
- ⊂ *SC-code*
- ⊂ *JSON*

Implementation of the Server System based on Websocket and JSON, providing network access to memory of the ostis-platform

- := [Implementation of a Websocket-based system that provides parallel-asynchronous multi-client access to sc-memory of the sc-model interpretation platform using the SC-JSON code]
- := [sc-json-server]
- ⇒ *frequently used sc-identifier**: [sc-server]
- := [sc-server]
- ∈ *reusable ostis-systems component stored as source files*
- ∈ *atomic reusable ostis-systems component*
- ∈ *dependent reusable ostis-systems component*
- ⇒ *method representation language used**:
 - C
 - C++
- ⇒ *language used**:
 - SC-JSON-code
- ⇒ *component address**: [https://github.com/ostis-ai/sc-machine/sc-tools/sc-server]
- ⇒ *component dependencies**:
 - {
 - *Library of software components for processing json texts*
 - *Library of cross-platform software components for implementing server applications based on Websocket*
 - *Software component for setting up software components of ostis-systems*
 - *Implementation of sc-memory*

Implementation of the interpreter for sc-models of user interfaces

- := [Our proposed interpreter for interpreting sc-models of ostis-systems user interfaces]
- ∈ *reusable ostis-systems component stored as source files*
- ∈ *non-atomic reusable ostis-systems component*
- ∈ *dependent reusable ostis-systems component*
- ⇒ *method representation language used**:
 - JavaScript
 - TypeScript
 - Python
 - HTML
 - CSS
- ⇒ *component address**: [https://github.com/ostis-ai/sc-web]

- ⇒ *component dependencies**:
 - {
 - *Library of standard interface components in the JavaScript programming language*
 - *Library for implementing server applications in the Python Tornado programming language*
 - *Implementation of the client system in the TypeScript programming language*
 - *Implementation of the client system in the Python programming language*

VI. PROSPECTS FOR DEVELOPING THE SOFTWARE PLATFORM OF OSTIS-SYSTEMS

Software implementation of the ostis-platform

- ⇒ *prospects for development**:
 - Despite the fact that the *Implementation of ostis-platform sc-memory* is functionally complete for the development of *semantically compatible interoperable ostis-systems* and is *multi-user*, i.e. it can execute *actions* of different users in parallel, significant restrictions are imposed on the *actions* of these users. First of all, these restrictions are connected not so much with the memory model underlying the implementation but with the model of asynchronous access to it. The implemented model of asynchronous memory access requires blocking access to a group of related *sc-elements* and not to a particular one of these *sc-elements*. For example, to create an *outgoing sc-arc* from a given *sc-element*, it is necessary to lock not only the cell in memory in which this *sc-element* is stored but also the initial *incoming* and *outgoing sc-connectors* from the list of *incoming* and *outgoing sc-connectors* of this *sc-element*, respectively. In the process of parallel operation of *sc-memory*, blunders can often occur: deadlocking of processes performing actions on the same *sc-elements*, resource races on the same *sc-elements*, etc. To eliminate these problems, a transition to a new model of asynchronous access to *sc-memory* is required or a transition to a new implementation of *sc-memory* without changing the existing programming interface for the implementation of *sc-memory*.
 - The current *Software implementation of the ostis-platform* is *customized* and does not include the *Implementation of the SCP Language interpreter* (that is, when the *ostis-platform* is running, the *SCP Language interpreter* is not used), which hinders the development of *platform-independent ostis-systems*. This is in no way related to the complexity of developing

this kind of interpreter. On the contrary, the problem lies in the model of asynchronous access to sc-memory, which prevents the collection of sc-agents that are part of the *Implementation of the SCP Language interpreter* to work smoothly. To implement a full-fledged collective of *ostis-systems* interacting with each other, it is necessary to transfer the *Software implementation of the ostis-platform* from the *specialized ostis-platforms* class to the *basic ostis-platforms* class. Thus, it is necessary to switch to a new version of the *ostis-platform* (not a modification (!)), which will contain the current *Implementation of the SCP Language interpreter*.

- The current *Implementation of ostis-platform sc-memory* is efficient for storing large amounts of *knowledge* in *ostis-systems knowledge bases*. However, in information retrieval problems, rather complex tools and subsystems are required to ensure the most effective solution of these problems. So, for example, to find all pairs of a given relation whose first component is a given sc-element, it is necessary to check the entire list of outgoing sc-connectors of a given sc-element, including those sc-connectors that do not have the specified syntactic or semantic sc-element class. The solution to this problem is possible by modifying the existing *Implementation of ostis-platform sc-memory*, namely, the implementation of a new sc-memory model (for example, on the file system of the modern Linux operating system).
- Implementation of the *OSTIS Ecosystem* [20], [21] requires strong development of the *Implementation of the subsystem for interacting with the external environment using languages of network interaction*, with the help of which *ostis-systems*, which are developed on the current *Software implementation of the ostis-platform*, will be able to fully communicate with each other. The transition of *Software implementation of the ostis-platform* from the class of server platforms to the class of client-server platforms is required.

VII. CONCLUSION

Let us briefly list the main provisions of this work:

- The current *Software implementation of the ostis-platform* is cross-platform, which allows:
 - developing and maintaining the state of its components, regardless of the implementation of the platforms on which the tools for their design and development are used;
 - using it to solve problems on any available devices.
- The current *Software implementation of the ostis-platform* is multi-user, that is, it allows processing several actions at the same time.
- The current *Implementation of memory in the ostis-platform* is complete enough to:
 - one-to-one interpret *sc-models of ostis-systems*, including external information constructions that do not belong to the SC-code;
 - develop platform-specific components that require access to sc-memory (for example, the *Software interface of Implementation of ostis-platform sc-memory*).
- The current *Software implementation of the ostis-platform* is specialized, that is, it allows creating only platform-dependent *ostis-systems*.
- On the basis of the current *Software implementation of the ostis-platform*, the *interpreter of sc-models of ostis-systems user interfaces*, *interpreter of logical models for solving problems in ostis-systems*, as well as the *manager of reusable ostis-systems components* are used.

In this article, the principles underlying the Software implementation of the *ostis-platform*, the principles of documenting its components, as well as the prospects for further development are described.

ACKNOWLEDGMENT

The author would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help in the work and valuable comments.

REFERENCES

- [1] V. Golenkov, N. Guliakina, V. Golovko, V. Krasnoproshin, "Methodological problems of the current state of works in the field of artificial intelligence," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, pp. 17–24, 2021.
- [2] A. Sokolov, A. Golubev, "Sistema avtomatizirovannogo proektirovaniya kompozitsionnykh materialov. CHast' 3. Grafoorientirovannaya metodologiya razrabotki sredstv vzaimodejstviya pol'zovatel'-sistema [System of computer-aided design of composite materials. Part 3. Graph-oriented methodology for the development of user-system interaction tools]," *Izvestiya SPBGETU LETI*, pp. 43–57, 2021.
- [3] V. Golenkov, N. Guliakina, I. Davydenko, A. Eremeev, "Methods and tools for ensuring compatibility of computer systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, pp. 25–52, 2019.
- [4] A. M. Ouksel and A. Sheth, "Semantic interoperability in global information systems," *SIGMOD Rec.*, vol. 28, no. 1, p. 5–12, Mar. 1999.
- [5] N. Zotov, "Software platform for next-generation intelligent computer systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, pp. 297—326, 2022.

- [6] D. Shunkevich, D. Koronchik, "Ontological approach to the development of a software model of a semantic computer based on the traditional computer architecture," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 75–92, 2021.
- [7] V. Golenkov, N. Gulyakina, and D. Shunkevich, *Otkrytaya tekhnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheski sovmestimyykh gibridnykh intellektual'nykh komp'yuternykh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint, 2021.
- [8] V. Ivashenko, "General-purpose semantic representation language and semantic space," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 41–64, 2022.
- [9] V. Golenkov, N. Guliakina, G. V., and E. V., "The standardization of intelligent computer systems as a key challenge of the current stage of development of artificial intelligence technologies," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 73–88, 2018.
- [10] D. Shunkevich, "Universal model of interpreting logical-semantic models of intelligent computer systems of a new generation," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, p. 285–296, 2022.
- [11] M. Orlov, "Comprehensive library of reusable semantically compatible components of next-generation intelligent computer systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 261–272, 2022.
- [12] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen, and D. Wilkins, "A comparison of a graph database and a relational database: a data provenance perspective," in *Proceedings of the 48th annual Southeast regional conference*, 2010, pp. 1–6.
- [13] C. Chen *et al.*, "Multi-perspective evaluation of relational and graph databases," 2022.
- [14] A. Zagorskiy, "Factors that determine the level of intelligence of cybernetic systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, p. 13–26, 2022.
- [15] V. V. Golenkov, "Ontology-based design of intelligent systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 37–56, 2017.
- [16] T. Dillon, E. Chang, and P. Wongthongtham, "Ontology-based software engineering-software engineering 2.0," 04 2008, pp. 13–23.
- [17] D. Shunkevich, "Ontology-based design of hybrid problem solvers," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 101–131, 2022.
- [18] N. Zotov, "Semantic theory of programs in next-generation intelligent computer systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 297–326, 2022.
- [19] Ian Robinson, Jim Webber and Emil Eifrem, *Graph databases*. O'Reilly Media, Inc., 2015.
- [20] A. Zagorskiy, "Principles for implementing the ecosystem of next-generation intelligent computer systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, p. 347–356, 2022.
- [21] T. Dillon, C. Wu, and E. Chang, "Gridspace: Semantic grid services on the web-evolution towards a softgrid," in *3rd International Conference on Semantics, Knowledge, and Grid, SKG 2007*, 11 2007, pp. 7–13.

Принципы проектирования, структура и перспективы развития Программной платформы ostis-систем

Зотов Н. В.

Данная работа является краткой спецификацией текущего Программного варианта реализации ostis-платформы. Работа показывает принципы, структуру и перспективы развития программной платформы для логико-семантических моделей систем, построенных по принципам Технологии OSTIS.

Received 13.03.2023

Implementation of Information Retrieval Subsystem in the Software Platform of ostis-systems

Nikita Zotov

*Belarusian State University of
Informatics and Radioelectronics*

Minsk, Belarus

Email: nikita.zotov.belarus@gmail.com

Abstract—The article describes the purpose and implementation variants for information retrieval subsystems of next-generation intelligent computer systems. This paper is a formal specification of how the information retrieval subsystem in the current Software implementation of the ostis-platform, as well as its software interface, are implemented, and is a continuation of a series of works on the design and implementation of the basic Software implementation of the ostis-platform [1], [2].

Keywords—information retrieval, information retrieval problem, isomorphic search, graph template, graph information retrieval system, ontological design, graph storage, ostis-platform

I. INTRODUCTION

One of the most important tasks of *intelligent computer systems* [3] is to satisfy the information needs of users. *Intelligent computer systems* should not only find the necessary (relevant) information for the user, but also provide quality answers to the user's questions. Thus, *intelligent computer systems* based on *graph representation of knowledge* should include entire software complexes for searching information relevant to the user — *graph information retrieval subsystems* [4], [5], [6].

Existing *graph information retrieval systems* are based on the use of *graph algorithms* for searching, storing and presenting information [7], [8]. Graphs are used to model relationships between objects, such as web pages on the Internet, users on social networks, or others. In such systems, users can use search queries to find information in a graph. Queries may be similar to those used in traditional information systems, but instead of searching by keywords, the user searches for objects and the relationships between them.

Information retrieval tasks are of great relevance, since at present the amount of information available on the Internet is too large for a person to handle without using appropriate search engines [9]. Information flows are growing every day, and therefore a more efficient and accurate use of information is becoming increasingly important for decision-making, planning,

scientific research and other activities. Moreover, the ability to conduct high-quality and accurate information searches is a key skill for people in the modern world.

II. EXISTING ANALOGS OF GRAPH INFORMATION RETRIEVAL SYSTEMS

Modern *graph information retrieval systems* use the *PageRank* [10] algorithm to determine the relevance of search results. *PageRank* evaluates the importance of each object in the graph based on the links it contains from other objects, and those objects that are considered more important are ranked higher. Graphs also allow the use of analytical algorithms, such as community detection algorithms, to identify subgraphs that group objects according to certain criteria. This can help users find information that might not be found in a traditional keyword search.

Examples of *graph information retrieval systems* are *Google Knowledge Graph* [11], *Facebook Graph Search*, *LinkedIn Skills Graph* [12] and *Neo4j* [13].

The use of *graph data models* in solving *information retrieval tasks* is explained as follows:

- Data processing performance is improved by one or more orders of magnitude when representing data as *graphs*, due to the properties of *graphs* themselves. Unlike *relational databases*, where query performance degrades as the dataset grows with increasing query intensity, *graph data model* performance remains constant even as the dataset grows. This is due to the fact that data processing is localized in some part of a *graph*. As a result, the execution time of each request is proportional only to the size of the *graph* part traversed to satisfy this request, and not to the size of the entire *graph* [14].
- *Graph data models* have tremendous expressive power. *Graph databases* offer an extremely flexible data model and way of representing it. *Graphs* are additive, which provides the flexibility to add new data relationships, new nodes, and new subgraphs

to an existing *graph* structure without violating its integrity and coherence.

In general, *graph information retrieval systems* allow efficient organization and retrieval of information using a graph's structure. This allows you to quickly and efficiently process large amounts of data and provide the user with the most relevant information.

III. IMPLEMENTATION OF THE INFORMATION RETRIEVAL SUBSYSTEM IN THE CURRENT SOFTWARE IMPLEMENTATION OF THE OSTIS-PLATFORM

Based on the current *Software implementation of the ostis-platform* [2] for *next-generation intelligent computer systems*, implemented according to the principles of *OSTIS Technology* [15], there is a need to create an information retrieval subsystem that will allow:

- solve *information retrieval tasks* of any level of complexity [16];
- implement *information retrieval subsystems* in *platform-dependent* and *platform-independent ostis-systems* for application purposes.

In addition, to provide interaction between *information retrieval subsystem of ostis-platform* and information retrieval subsystems of platform-dependent components and ostis-systems required programming interface.

SCin-code described in [2] is enough to represent *sc-texts* inside the *sc-memory of the ostis-platform* [17]. To translate some *sc-text* into *sc-memory of the ostis-platform*, you must use the methods of the current *Implementation of sc-memory in the ostis-platform*. The *sc-memory* methods described below are the formal specification of the current ***Software interface of Implementation of sc-memory in the ostis-platform***, with which you can perform actions with *sc-memory*.

In the current *Implementation of sc-memory in the ostis-platform* all program methods are implemented in *method presentation languages C* and *C++*. The current *Software interface of the Implementation of sc-memory in the ostis-platform* contains the necessary functionality not only to perform actions on the *elements of sc-memory*, but also — on the *elements of file memory* [2]. This *software interface* is one of the languages of the current **Software implementation of the ostis-platform** for performing actions on *sc-memory* and can be used to solve *problems* of any information-based complexity. So, for example, this *software interface* is used in the current *Implementation of the Server system based on Websocket and JSON providing network access to this sc-memory* [2], *Implementation of the ostis-system reusable component manager* and *Implementation of the interpreter for logical models of problem solving* [18], as well as when implementing any *platform-dependent ostis-systems* for any purpose.

Software interface of Implementation of ostis-platform sc-memory

⇐ *software interface**:

Implementation of ostis-platform sc-memory software interface

∈ reusable ostis-systems component stored as source files

∈ atomic reusable ostis-systems component

∈ dependent reusable ostis-systems component

⇒ component dependencies*:

- GLib library of methods and data structures
- C++ Standard Library of methods and data structures

}

⇒ method representation language used*:

- C
- C++

⊃ Software interface for information-forming methods of Implementation of ostis-platform sc-memory

:= [information-forming methods of Implementation of ostis-platform sc-memory]

:= [subsystem that is part of the implementation of ostis-platform sc-memory, which allows to create, modify, and delete constructions of sc-memory]

⇐ software interface*:

Implementation of the information-generating subsystem of Implementation of ostis-platform sc-memory

⊂ Implementation of ostis-platform sc-memory

⊃ Software interface for information retrieval methods of Implementation of ostis-platform sc-memory

:= [information retrieval methods of Implementation of ostis-platform sc-memory]

:= [subsystem that is part of Implementation of ostis-platform sc-memory that allows to find constructions in sc-memory]

⇐ software interface*:

Implementation of the information retrieval subsystem of Implementation of ostis-platform sc-memory

⊂ Implementation of ostis-platform sc-memory

Logically, the current *Software interface of the implementation of sc-memory in the ostis-platform* is divided into two software interfaces: ***Software interface of information-forming methods of Implementation of sc-memory in the ostis-platform*** and ***Software interface of information retrieval methods Implementation of sc-memory in ostis-platform***. First of all, this division is due to the fact that the implementation of information retrieval methods in the current *Implementation of sc-memory in the ostis-platform* is rather complicated and

requires much more clarification when describing this implementation. Also, this separation of the *software interface* allows the specification of the methods of the *Software interface of the Implementation of sc-memory in the ostis-platform* to be singled out and structured in such a way that it remains uniform, compact and simple for an external user. As such, there is no physical separation in the *Software interface of the Implementation of sc-memory in the ostis-platform*, all methods of the *Software interface of the Implementation of sc-memory in the ostis-platform* can be used in the same programmatic way and are components **Reusable component library of the Software implementation of the ostis-platform**, that is, they can be used in the implementation of other special-purpose components.

IV. IMPLEMENTATION OF ITERATIVE SEARCH FOR CONSTRUCTIONS IN THE SC-MEMORY OF THE OSTIS-PLATFORM

In tasks solved in applied ostis-systems implemented on the basis of the current *Software implementation of ostis-platform*, it is necessary to use search mechanisms for already existing elements or *constructions in sc-memory*. Such mechanisms are part of the **Implementation of the information retrieval subsystem for the Implementation of sc-memory in the ostis-platform**, on the basis of which *information retrieval subsystems* can be implemented for *platform-dependent* and *platform-independent ostis-systems*. Despite the complexity of *information retrieval*, current *Software implementation of the ostis-platform* makes it possible to effectively use the implemented *information retrieval* methods in tasks solved by applied ostis-systems. This subsystem cannot be implemented independently of the implementation of the *ostis-platform*, that is, it cannot be made *platform-independent*, so it is necessary to separate *Implementation of the information retrieval subsystem for the Implementation of sc-memory in the ostis-platform* and *Implementation of the information retrieval subsystem of the OSTIS Metasystem* [19], which is implemented in the *SCP Language* [17], independently of the current *Software implementation of the ostis-platform*. The *scp-interpreter* itself must use *information retrieval* methods of the *Implementation of sc-memory in the ostis-platform*, and the *SCP language* must provide the ability to navigate through the *knowledge base* of any *ostis-systems*.

Software interface for information retrieval methods of Implementation of sc-memory in the ostis-platform

- ⊃ *Method for creating a three-element sc-memory construction search iterator*
- ⊃ *Method for creating a five-element sc-memory construction search iterator*

- ⊃ *Method for finding sc-memory constructions isomorphic to the specified graph template*
- ⊃ *Method for creating sc-memory constructions isomorphic to the specified graph template*
- ⊃ *Method for creating an object of the graph template*

Method for creating a three-element sc-memory construction search iterator

- ∈ *method*
- ⇒ *input argument classes of a method**:
 - {
 - *parameter of the Method for creating an sc-memory construction search iterator*
 - *parameter of the Method for creating an sc-memory construction search iterator*
 - *parameter of the Method for creating an sc-memory construction search iterator*
- }
 - ⇒ *method result class**:
 - *three-element sc-memory construction search iterator*
 - ⇒ *class of exceptions**:
 - *element with the specified sc-address does not exist in sc-memory*

Method for creating a five-element sc-memory construction search iterator

- ⇒ *method input argument classes**:
 - {
 - *parameter of the Method for creating an sc-memory construction search iterator*
 - *parameter of the Method for creating an sc-memory construction search iterator*
 - *parameter of the Method for creating an sc-memory construction search iterator*
 - *parameter of the Method for creating an sc-memory construction search iterator*
 - *parameter of the Method for creating an sc-memory construction search iterator*
- }
 - ⇒ *method result class**:
 - *five-element sc-memory construction search iterator*
 - ⇒ *class of exceptions**:
 - *element with the specified sc-address does not exist in sc-memory*

According to the rules of *SC-code* syntax, *sc-constructions*, i.e. constructions consisting of *sc-elements*, can consist of three *sc-elements* (Figure. 1), five *sc-elements* (Figure. 2), seven *sc-elements*, and so on [20]. In the *sc-memory of the ostis-platform*, the equivalent of an *sc-construction* is a construction consisting of *sc-memory elements (sc-memory construction)*. **Method for creating a three-element sc-memory construction search iterator** and **Method for creating a five-element sc-memory**

construction search iterator allow you to create iterators for searching for three- and five-element *constructions in sc-memory of the ostis-platform*, respectively. Using the parameters of these methods, you can create iterators of any necessary configuration to search for *three- and five-element sc-constructions*. So, for example, if it is necessary to find all *sc-memory elements corresponding to sc-elements* that belong to some *sc-set* to which a given *sc-memory element* corresponds, then it is necessary to use *Method for creating a three-element sc-memory construction search iterator* to create a search iterator, specifying as the first argument the *sc-memory element* corresponding to the specified *sc-set*, and as the second and third arguments — *class of sc-memory elements corresponding to base sc-arc* and *class of sc-memory elements corresponding to sc-elements of unspecified class*, respectively. To search *sc-memory* for more complex structures consisting of seven or more elements, you need to combine the search iterators for three- and five-element *constructions in sc-memory*, or use the **Method for searching for sc-memory constructions isomorphic to the specified graph template**.



Figure 1. SC.g-text. Example of three-element sc-construction

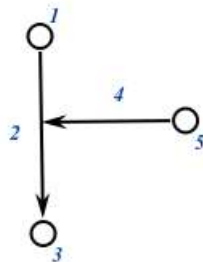


Figure 2. SC.g-text. Example of five-element sc-construction

The *software interface* of these *methods* is constrained by the *method representation language C++*. For example, using these *methods* you cannot create *iterators* by specifying only *sc-memory element classes* as arguments, by which you need to find all *corresponding constructions in sc-memory*, where *classes of their sc-storage* are the classes passed as arguments to the specified methods, nor is it possible to specify as arguments anything other

than an *sc-address* or an *sc-memory element class*. An attempt to perform any of the above will result in an error of "gluing" the interface of one of the specified *methods* specified in the *C++* header file with the implementation with the specified parameters specified in the *C++* source file because the *C++* compiler cannot find an implementation for the specified *software interface*. Thus, due to the aforementioned problem, the parameters for *Method for creating a three-element sc-memory construction search iterator* and *Method for creating a five-element sc-memory construction search iterator* can be *sc-address* and/or *sc-memory element class*.

The *iterators* created using the specified *methods* also have a *software interface*. The results of both *methods* are different *iterators*, that share, however, the same *software interface*. Such *iterators* allow you to work with *sc-memory constructions* at the same moment they are found. Using the *Method of moving to the next sc-memory construction "suitable" for the specified iterator*, the *iterator* updates its internal state each time a new *sc-memory construction* is found. By *next sc-memory construction "suitable" for the specified iterator* we mean an *sc-memory construction* whose elements match the configuration of the created *sc-memory construction search iterator* and which was not found earlier. The result of the latter *method* is the boolean *True* value if the next "suitable" construction for the specified *iterator* exists in *sc-memory*. If there are no "suitable" constructions for this *iterator* in *sc-memory*, then the *Method of moving to the next sc-memory construction "suitable" for the specified iterator* results in the boolean value *False*. To get the *sc-address* of some of the elements of the found *sc-memory construction*, you must use the *Method of accessing the sc-address of the specified sc-memory construction element by the position number of this element in the specified sc-memory construction*, specifying as an argument an integer value in the form of the position index of the searched element in this construction. In this case, the indexing of the positions of elements in the *sc-memory construction* in the current *Implementation of sc-memory in the ostis-platform* starts from zero, not from one, and the indexing order is determined by the order of arguments that have been used when creating the *iterator*. If you try to specify for this method an index for which there is no position in this construction, this method will result in an invalid element position in the specified *sc-memory construction exception*. Thus, the range of indices for *three-element constructions* is limited from zero to two, and for five-element constructions, from zero to four.

Software interface for information retrieval methods of Implementation of sc-memory in the ostis-platform

```

⊃=
{
⊃   Extension of Software interface for information

```


*retrieval methods of Implementation of
sc-memory in the ostis-platform*

**three- and five-element sc-memory construction search
iterator**

⊂ software object
:= [ScIterator]
∈ C++

**Software interface for three- and
five-element sc-memory construction
search iterator**

⊃=
{
⇐ software interface*:
three- and five-element sc-memory construction
search iterator

**Method of moving to the next sc-memory construction
"suitable" for the specified iterator**

∈ method
⇒ method header in method representation
language*:
[bool Next() const]
∈ C++
⇒ method result class*:
• boolean

**Method of accessing the sc-address of an element of
the specified sc-memory construction by the position
index of this element in the specified construction**

∈ method
⇒ method header in method representation
language*:
[ScAddr Get(size_t idx) const]
∈ C++
⇒ method input argument classes*:
{• 32-bit integer
}
⇒ method result class*:
• sc-address of sc-memory element
⇒ class of exceptions*:
• invalid element position in the specified
sc-memory construction
}
}

For Method for creating a three-element sc-memory construction search iterator, as well as for Method for creating a five-element sc-memory construction search iterator, various combinations of parameters can be used, except for combinations where all parameters are classes of sc-memory elements. For simplicity and compactness of the terms used at the level of implementation of methods for creating iterators for searching for structures in sc-memory, additional notations are introduced: the symbol

"f" (from the English word "fixed") denotes the fact that the parameter of a given method for creating an sc-memory construction search iterator is the sc-address of some sc-memory element, and the symbol "a" (from the English word "assign") denotes sc-memory element class[^]. For Method for creating a three-element sc-memory construction search iterator, the correct designation of the desired constructions will be a three character long combination of characters "f" and "a", and for Method for creating a five-element sc-memory construction search iterator — a five-character combination of "f" and "a". In the SCP Language, to indicate whether a variable has the specified value, the corresponding role relations are used: for variables of class "f" — scp-operand with the specified value', for variables of class "a" — scp-operand with free value' [17].

**Software interface for information
retrieval methods in the
Implementation of sc-memory in the
ostis-platform**

⊃=
{

**Method for creating a three-element sc-memory
construction search iterator**

⊃ Method for creating an fff-construction search
iterator
∈ method
⇒ method input argument classes*:
{• sc-address of sc-memory element
• sc-address of sc-memory element
• sc-address of sc-memory element
}

⊃ Method for creating an faa-construction search
iterator

∈ method
⇒ method input argument classes*:
{• sc-address of sc-memory element
• sc-memory element class[^]
• sc-memory element class[^]
}

⊃ Method for creating an aaf-construction search
iterator

∈ method
⇒ method input argument classes*:
{• sc-memory element class[^]
• sc-memory element class[^]
• sc-address of sc-memory element
}

⊃ Method for creating an faf-construction search
iterator

∈ method
⇒ method input argument classes*:
{• sc-address of sc-memory element

```

    • sc-memory element class^
    • sc-address of sc-memory element
  }
⊃ Method for creating an afa-construction search
   iterator
  ∈ method
  ⇒ method input argument classes*:
  {
    • sc-memory element class^
    • sc-address of sc-memory element
    • sc-memory element class^
  }
}

```

These variants of the implementation of the *Method for creating a three-element sc-memory construction search iterator* are sufficient for solving any search and navigation tasks. *Method for creating an ffa-construction search iterator* and *Method for creating an aff-construction search iterator* are possible, but in practice there is no need to look for a third sc-memory element by the known element corresponding to the sc-connector and the element corresponding to the sc-element from which this sc-connector exits or into which this sc-connector enters. Such a problem can be solved using *Method for creating an afa-construction search iterator*. However, the implementation of *Method for creating a five-element sc-memory construction search iterator* is not at all necessary, since all tasks solved using this method can also be solved using *Method for creating a three-element sc-memory construction search iterator*, however, the implementation of *Method for creating a five-element sc-memory construction search iterator* allows you to make the text of the method more compact compared to the method that would use *Method for creating a three-element sc-memory construction search iterator*.

The following can be specified as all three arguments for the *Method for creating a three-element sc-memory construction search iterator*:

- *sc-addresses of sc-memory elements* (for example, when solving the problem of checking the incidence of all three specified sc-memory elements),
- *sc-storage element address, class of sc-storage elements corresponding to sc-connectors*[^] that come out of the *sc-storage element* passed as the first argument, and *sc-address of the sc-memory element* corresponding to some *sc-element* that contains the required *sc-connectors* (for example, when solving the problem of finding all *sc-memory elements corresponding to sc-connectors between sc-elements* for which the specified *sc-memory elements* correspond),
- *sc-storage element address, class of sc-storage elements corresponding to sc-connectors*[^] that come out of the *sc-storage element* passed as the first argument, and *class of sc-memory elements corresponding to some sc-elements*[^], which include the required *sc-connectors* (for example, when solving the problem

of finding all *sc-memory elements* that correspond to *sc-connectors* coming from the *sc-element* that matches the *sc-memory element* passed as the first argument),

- *class of sc-memory sc-elements*[^], *class of sc-memory elements corresponding to sc-connectors*[^] that come out of the *sc-memory elements* specified as the first argument, and *sc-address of the sc-memory element corresponding to some sc-element*, which contains the required *sc-connectors* (for example, when solving the problem of finding all *sc-memory elements corresponding to sc-connectors*, contained in the *sc-element* that matches the *sc-memory element* specified as the third argument),
- *sc-memory element class*[^], *sc-address of the sc-memory element corresponding to the sc-connector* that comes out of the *sc-memory element* passed as the first argument, and *class of sc-memory elements corresponding to some sc-element*[^], which contains the required *sc-connector* (for example, the task of finding *sc-memory elements corresponding to sc-elements*, one of which is the *sc-element* from which the *sc-connector* emerges, for which the specified *sc-memory element* matches, and the second of which is the *sc-element* that includes this *sc-connector* for which the specified *sc-memory element* matches)
- and so on.

As all five arguments for the *Method for creating a five-element sc-memory construction search iterator*, other combinations can be specified that are not specified in the presented classification. However, this is not necessary, since all tasks solved using such *iterators* can be solved by already existing *five-element sc-memory construction search iterators*.

V. IMPLEMENTATION OF ISOMORPHIC SEARCH FOR SC-MEMORY CONSTRUCTIONS OF THE OSTIS-PLATFORM ACCORDING TO THE SPECIFIED GRAPH TEMPLATE

Method for creating a three-element sc-memory construction search iterator and *Method for creating a five-element sc-memory construction search iterator*, as well as *Software interface for three- and five-element sc-memory construction search iterator* are quite powerful tools for solving any *information retrieval problems* in applied *ostis-systems*. For example, in *inference* [18] problems, it is considered convenient to solve *problems* when *search for structures* of any necessary configuration in *sc-memory* reduces to *isomorphic search* of these constructions according to the specified *graph template* (Figure. 3). Such *graph templates* can be any *atomic logical formulas* included in any other *non-atomic formula* [18].

Isomorphic search is one way to solve the *problem* of finding a subgraph in a *graph* (see [21]). The *problem* consists of finding all occurrences of the specified *graph*

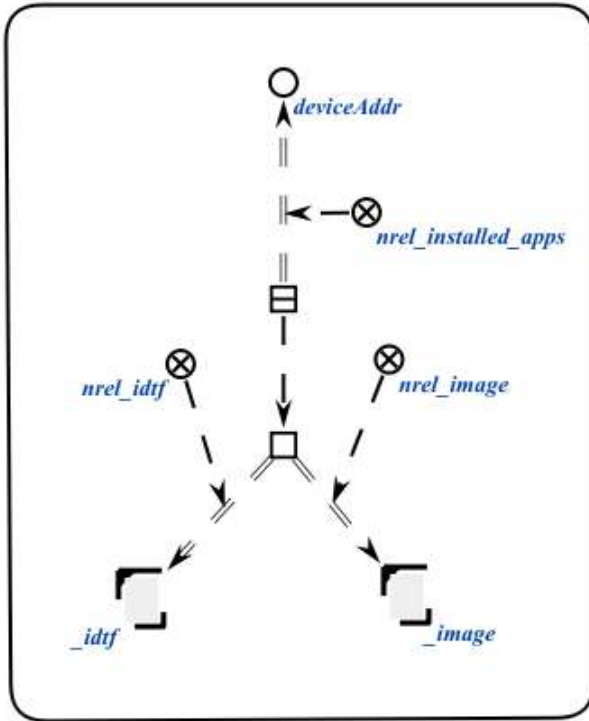


Figure 3. SC.g-text. Graph template example

template in the source graph. The *isomorphic search* process can be implemented using various algorithms. One of them is *Ullman's Algorithm* [22], which is based on using an adjacency matrix to determine the correspondence between graph vertices. Another algorithm is the *VF2 Algorithm*, which uses a comparison function to check if the corresponding nodes and edges in two graphs match (see [23]). In modern computer science, there are algorithms that allow solving the *problem of isomorphic search* in subexponential time (see [24]).

The fundamental principle of the *OSTIS Technology* that is currently under development is the principle of adopting the best existing technologies for the development of *ostis-systems* [25]. However, due to various circumstances, for example, connected with the specific features of the *Implementation of sc-memory in the ostis-platform*, as well as the requirements imposed on *sc-agents* involved in logical inference, it is necessary to apply and test new solutions. Within the framework of the current *Implementation of sc-memory in the ostis-platform*, a concept of *isomorphic search* has been developed, which allows to find graphs isomorphic to fragments of a given *graph template* in *optimal time*.

In general there is no need to implement *isomorphic search* in a generic way. This is explained by the following:

- *isomorphic search* is an NP-complete problem,

which means that the cost of solving it grows exponentially with the size of the input data, and there is no efficient algorithm for solving the *isomorphic search* problem yet;

- as a result of determining the isomorphism of two given graphs, several nodes can be found that correspond to each other, but are not actually isomorphic;
- due to the exponential growth in the number of possible isomorphism variants with increasing graph size, even small errors in the calculation of isomorphism can lead to severe distortion of the results;
- for large graphs, the time spent on enumeration of all possible isomorphisms can be very high. This can reduce search efficiency and limit the use of *isomorphic search* in real-world problems (see [26]);
- search complexity increases with the number of loops in the original *graph*, as it results in more iterations;
- existing algorithms are either slow or waste memory, resulting in *isomorphic search* being slow.

Some isomorphic search algorithms even have $O(n!)$ complexity and cannot be used for large graphs (see [21]). Despite all the problems associated with *isomorphic search*, for the convenience of solving logical problems, the current *Software implementation of the ostis-platform* implements the "most appropriate" *isomorphic search* algorithm. The current version of *isomorphic search* is implemented in the *Method for finding sc-memory constructions isomorphic to the specified graph template*. This method allows you to find *sc-memory constructions* that are isomorphic not just to some *graph template* that is represented in *sc-memory*, but to the *program object* of this *graph template*, i.e. *graph template*, which is presented in a program format convenient for quick processing.

- ⇒ *Method for finding sc-memory constructions isomorphic to the specified graph template*
- ∈ *method*
- ⇒ *method header in method representation language**:


```
[ScTemplate::Result      HelperSearchTemplate(ScTemplate const
& templ, ScTemplateSearchResult & result)]
∈ C++
```
- ⇒ *input argument classes of a method**:
 - *graph template program object*
 - *tuple of program objects of sc-memory constructions isomorphic to the specified graph template*
- ⇒ *method result class**:
 - *error code of the result of creating a program object by the specified element corresponding to the graph template*
- ⇒ *class of exceptions**:
 - *syntactically incorrect graph template*

- *program object*
- *semantically incorrect graph template*
- *program object*

⇒ *Method for creating a construction in sc-memory that is isomorphic to the specified graph template*

∈ *method*

⇒ *method header in method representation language*:*

```
[ScTemplate::Result      HelperGenTem-
plate(ScTemplate const &
templ, ScTemplateGenResult & result)]
∈ C++
```

⇒ *input argument classes of a method*:*

- {• *graph template program object*
- *program object of the sc-memory construction isomorphic to the specified graph template*

⇒ *method result class*:*

- *error code of the result of creating a program object by the specified element corresponding to the graph template*

⇒ *class of exceptions*:*

- *syntactically incorrect graph template program object*
- *semantically incorrect graph template program object*

Method for creating a graph template program object

∈ *method*

⇒ *method header in method representation language*:*

```
[ScTemplate::Result      HelperBuildTem-
plate(ScTemplate & templ, ScAddr const
& templAddr)]
∈ C++
```

⇒ *input argument classes of a method*:*

- {• *graph template program object*
- *sc-address of sc-memory element*

⇒ *method result class*:*

- *error code of the result of creating a program object by the specified element corresponding to the graph template*

⇒ *class of exceptions*:*

- *syntactically incorrect graph template program object*
- *semantically incorrect graph template program object*

Graph template program object can be generated using the *Method for creating graph template program object* by passing as arguments *graph template program object* as an output parameter and *sc-address of the sc-memory element corresponding to the sc-structure of an atomic*

logical formula (sc-template). Like *three- and five-element sc-memory construction search iterator*, *graph template program object* has a specialized *software interface*. Initially, *graph template program object* is set to empty. Using **Method of adding a three-element construction for the specified graph template** and **Method of adding a five-element construction for the specified graph template**, you can extend the specified *graph template program object*. The *graph template program object* is expanded with the addition of three-element constructions to the **tuple of program objects of three-element constructions in the specified graph template**, while the order of the program objects of constructions in this set is specified by the execution sequence *Method for adding a three-element construction for the specified graph template* and *Method for adding a five-element construction for the specified graph template*.

Software interface for information retrieval methods of Implementation of sc-memory in the ostis-platform

⊃=

{

graph-template program object

```
:= [atomic logical formula program object]
:= [sc-template program object]
⊂ program object
⇒ concept specifying the specified entity*:
{• tuple of program objects of three-element constructions in the specified graph template- local identifier of the sc-memory element and position indices of this element in the specified graph template*
- set of sets of indices of program objects of three-element construction in the specified graph template, ordered by the search priority
- local identifier of the sc-memory element and sc-address of this element in the specified graph template*
- local identifier of the sc-memory element and the class of this element in the specified graph template*

}
```

program object of a three-element construction of the specified graph template

```
:= [ScTemplateTriple]
∈ C++
∈ program object
⇒ concept specifying the specified entity*:
{• index of the program object of the three-element structure in the specified graph template
}
```

```

    := [size_t m_index]
    • tuple of three elements of a program
      object of a three-element construction
    := [std::array<ScTemplateItem, 3>
        m_values]
}

```

element of program object of three-element construction

```

:= [ScTemplateItem]
  ∈ C++
∈ program object
⇒ concept specifying the specified entity*:
{
• sc-address of the program object element
  of the three-element structure
  := [ScAddr m_addrValue]
  ⊂ sc-address of sc-memory element
• element class of a program object of a
  three-element construction
  := [ScType m_typeValue]
  ⊂ sc-memory element class^
• local identifier of the program object of
  the three-element construction
  := [std::string m_name]
}

```

Software interface of graph template program object

```

⊃=
{
⇐ software interface*:
  graph template program object
}

```

Method for adding three-element construction to the specified graph template

```

∈ method
⇒ method header in method representation
  language*:
[ScTemplate & Triple(ScTemplateItemValue const
  & param1, ScTemplateItemValue const & param2,
  ScTemplateItemValue const & param3)]
∈ C++
⇒ input argument classes of a method*:
{
• parameter of the Method for adding
  construction to the specified graph
  template
• parameter of the Method for adding
  construction to the specified graph
  template
• parameter of the Method for adding
  construction to the specified graph
  template
}
⇒ method result class*:
• graph template program object

```

```

⇒ class of exceptions*:
• incorrect parameter of the Method for
  adding construction to the specified graph
  template
  ⊃ local identifier is not previously
    bound to the sc-address of the
    program object element of the
    three-element construction in the
    specified graph template program
    object
  ⊃ local identifier has already been
    used for another sc-address of the
    three-element construction
    program object element in the
    specified graph template program
    object
  ⊃ the same local identifier is
    simultaneously specified for the
    second and first (third) element of
    the specified program object of the
    three-element construction
  ⊃ element class of the created
    program object of the
    three-element construction in the
    specified program object of the
    graph template is the class of
    elements in the sc-memory
    corresponding to the sc-constants
    element with the sc-address of the
    element of the created program
    object of the three-element
    construction in the specified
    program object of the graph
    template does not exist in
    sc-memory

```

Method for adding five-element construction to the specified graph template

```

∈ method
⇒ method header in method representation
  language*:
[ScTemplate & Fiver(ScTemplateItemValue
  const & param1, ScTemplateItemValue const &
  param2, ScTemplateItemValue const & param3,
  ScTemplateItemValue const & param4, ScTem-
  plateItemValue const & param5)]
∈ C++
⇒ input argument classes of a method*:
{
• parameter of the Method for adding
  construction to the specified graph
  template
• parameter of the Method for adding
  construction to the specified graph
  template
• parameter of the Method for adding
  construction to the specified graph
  template
• parameter of the Method for adding
  construction to the specified graph
  template

```

```

    construction to the specified graph
    template
    • parameter of the Method for adding
    construction to the specified graph
    template
    • parameter of the Method for adding
    construction to the specified graph
    template
  )
⇒ method result class*:
    • graph template program object
⇒ class of exceptions*:
    • incorrect parameter of the Method for
    adding construction to the specified graph
    template
      ⊃ local identifier is not previously
      bound to the sc-address of the
      program object element of the
      three-element construct in the
      specified graph template program
      object
      ⊃ local identifier has already been
      used for another sc-address of the
      three-element construction
      program object element in the
      specified graph template program
      object
      ⊃ the same local identifier is
      simultaneously specified for the
      second and first (third) elements
      of the specified program object of
      the three-element construction
      element class of the created
      program object of the
      three-element construction in the
      specified program object of the
      graph template is the class of
      elements in the sc-memory
      corresponding to the sc-constants
      element with the sc-address of the
      element of the created program
      object of the three-element
      construction in the specified
      program object of the graph
      template does not exist in
      sc-memory
  }
}

```

To form the necessary *graph template program object*, do the following:

- If *graph template program object* has not been created before, then it must be created.
- For the created *graph template program object* apply several times *Method of adding a three-element structure for the specified graph template (Method*

of adding a five-element structure for the specified graph template), specifying as three (five) input parameters the **parameters of the Method for adding a construction for the specified graph template**, depending on the desired configuration of the *three-element (five-element) construction program object* to be added.

At the same time, *parameter of the Method for adding a construction for the specified graph template* differs significantly from the *parameter of the Method for creating sc-memory construction search iterator*. In addition to sc-addresses and classes of sc-memory elements, local identifiers of these addresses or classes can be specified in the created graph template. This greatly simplifies the process of creating the *graph template program object*, when it is necessary to specify *sc-address* or *sc-memory element class*[^] in the added construction, which was already specified earlier in another construction the specified *graph template*. Thus, using such a local identifier, it is possible to refer to the parameter of an already previously added construction in the specified graph template. In addition, this method allows you to get elements from the structures found by the specified *graph template* in the sc-memory using such local identifiers.

Regardless of what methods were applied to the created *graph template program object*, in the structure of the *graph template program object* itself, for the convenience of representing and processing data, only software objects of three-element constructions are created. Each element in *program object* of a *three-element construction*, except for the sc-address, class and local identifier, has its own position index within this construction, set in the range from zero to two, as well as the position index within the entire graph template, calculated as the sum of the product of the number of the *three-element construction* in the specified graph template and *number three* and *position index* of this element within the graph template. Adding *program object* of a *three-element construction* in the specified graph template program object to *graph template program object* is done as follows:

- If the argument specified as the second parameter has a local identifier in the specified graph template, and this local identifier is also specified for the first or second argument, then terminate the *Method for adding a three-element construct for the specified graph template* with the following exception: *the same local identifier is simultaneously specified for the second and first (third) elements of the specified program object of the three-element construction in the specified program object of the graph template.*
- If any of the parameters is specified as the class of the sc-memory element corresponding to the sc-constant, then terminate the *Method of adding a three-element construction for the specified graph template* with the following exception: *the element class of the three-*

element construct program object being created in the specified graph template program object is the class of the elements in the sc-memory corresponding to the sc-constants.

- *If any of the parameters is given as the sc-address of a non-existent element in sc-memory, then terminate the Method of adding a three-element construction for the specified graph template with the following exception: the element with the sc-address of the element of the created program object of the three-element construction in the specified program object of the graph template does not exist in the sc-memory.*
- *For all parameters that have local identifiers in the specified graph template and sc-addresses of elements in sc-memory, add all pairs with these local identifiers and the corresponding sc-addresses of elements in sc-memory to the relation local identifier of the sc-storage element and the sc-address of this element in the specified graph template*, otherwise, if the sc-addresses of the elements in the sc-storage for these local identifiers are known in relation to local identifier of the sc-storage element and sc- the address of this element in the specified graph template*, specify the known sc-addresses of the elements in the sc-memory for the specified parameters.*
- *For all parameters that have local identifiers in the specified graph template and element classes in sc-memory, add all pairs with these local identifiers and the corresponding element classes in sc-memory to the relation local element identifier sc-memory and the class of this element in the specified reference graph*.*
- *For all parameters for which only local identifiers are specified in the specified reference graph, as well as for the local identifier of the sc-memory element and the position numbers of this element in the specified reference graph*, if position indices are unknown by these local identifiers of the corresponding elements in the specified graph template, then add to this relation all pairs with local identifiers and position indices of the corresponding elements in the specified template graph.*
- *For the obtained program object of a three-element construct in the specified graph template program object, calculate the priority number required when executing the Method for finding sc-memory constructions isomorphic to the specified graph template:*
 - *If program object of a three-element construction contains sc-addresses of sc-memory elements for all elements, then the priority number of the specified construct is considered equal to zero (that is, it is considered the highest priority).*

- *If the software object of a three-element construction contains sc-address of the sc-memory element corresponding to the sc-connector for the second element, then the priority number of the specified construct is considered equal to one (i.e. it is the second by priority).*
- *If in the program object of a three-element construction for the first and third elements sc-addresses of the sc-memory element are specified, then the priority number of the specified construction is considered equal to two (that is, it is considered the third in priority).*
- *If the program object of a three-element construction contains sc-address of the sc-memory element only for the third element, then the priority number of the specified construct is considered equal to three.*
- *If in the program object of a three-element construction the first element is sc-address of the sc-memory element, and the third element is the class of the sc-memory element corresponding to sc-node^, then the priority number of the specified structure is considered equal to four.*
- *If in the program object of a three-element construction the first element is sc-address of the sc-memory element, and the third element is the class of the sc-memory element corresponding to sc-connector^, then the priority number of the specified structure is considered equal to five.*
- *If there are no elements in the software object of a three-element construction for which sc-addresses of sc-memory elements are specified, then the priority number of the specified construct is considered equal to six (that is, it is considered to be of the lowest priority).*

After determining the priority number of the specified program object of the three-element construct in the specified graph template program object, add this object to the set with the position equal to the calculated priority number of the set of sets of indices of program objects of three-element construction in the specified graph template, ordered by the search priority.

- *The obtained program object of a three-element construction of the specified graph template is added to tuple of program objects of three-element constructions in the specified graph template.*

Adding a five-element construction program object to graph template program object amounts to adding two three-element construction program object to this graph template program object. At the same time, in the second three-element construction program object only the local identifier of the second element of the first added three-element construct program object in the specified graph template program object is specified for the specified

graph template program object.

To find all *sc-memory constructions* isomorphic to a given *graph template program object*, do the following:

- For the generated *graph template program object* apply the *Method for finding sc-memory constructions isomorphic to the specified graph template*.
- The result of this method will be a *tuple of program objects of all constructions in sc-memory isomorphic to the specified graph template*, which, like the search iterator for three- and five-element constructions in *sc-memory*, has its own *program interface*.

Software interface for information retrieval methods of Implementation of sc-memory in the ostis-platform

$\supset=$
{

tuple of program objects of sc-memory constructions isomorphic to the specified graph template

$:=$ [ScTemplateSearchResult]
 \in C++
 \subset program object

Software interface of the tuple of program objects of sc-memory constructions isomorphic to the specified graph template

$\supset=$
{

\Leftarrow software interface*:
tuple of program objects of sc-memory constructions isomorphic to the specified graph template

Method of obtaining a program object of an sc-memory construction isomorphic to the specified graph template by its index in a tuple

\in method
 \Rightarrow method header in method representation language*:
[ScTemplateSearchResultItem operator[]](size_t index) const noexcept(false)]
 \in C++
 \Rightarrow input argument classes of a method*:
{

- 32-bit integer

}

\Rightarrow method result class*:

- program object of an *sc-memory construction isomorphic to the specified graph template*

\Rightarrow class of exceptions*:

- the element with the specified index does not exist in the tuple

Method of obtaining a program object of an sc-memory construction isomorphic to the specified graph template by its index in a tuple with a preliminary check for the specified index

\in method
 \in method without exceptions
 \Rightarrow method header in method representation language*:
[bool Get(size_t index, ScTemplateSearchResultItem & outItem) const noexcept]
 \in C++
 \Rightarrow input argument classes of a method*:
{

- 32-bit integer
- program object of an *sc-memory construction isomorphic to the specified graph template*

}

\Rightarrow method result class*:
 \in boolean
}

program object of an sc-memory construction isomorphic to the specified graph template

$:=$ [ScTemplateSearchResultItem]
 \in C++
 \subset program object

Software interface of the program object of an sc-memory construction isomorphic to the specified graph template

$\supset=$

{
 \Leftarrow programming interface*:
program object of an sc-memory construction isomorphic to the specified graph template

Method of obtaining the sc-address of an element of a program object of an sc-memory construction isomorphic to the specified graph template by its index in this program object

\in method
 \Rightarrow method header in method representation language*:
[ScAddr const & operator[]](size_t index) const noexcept(false)]
 \in C++
 \Rightarrow input argument classes of a method*:
{

- 32-bit integer

}

\Rightarrow method result class*:

- *sc-address of sc-memory element*

\Rightarrow class of exceptions*:

- there is no element at the specified index

in the program object of the sc-memory construction isomorphic to the specified graph template

Method of obtaining the sc-address of an element of a program object of an sc-memory construction isomorphic to the specified graph template by its index in this program object with a preliminary check of the specified index

∈ method
 ∈ method without exceptions
 ⇒ method header in method representation language*:
 [bool Get(size_t index, ScAddr & outAddr) const noexcept]
 ∈ C++
 ⇒ input argument classes of a method*:
 {
 • 32-bit integer
 • sc-address of sc-memory element
 }
 ⇒ method result class*:
 • boolean

Method of obtaining the sc-address of an element of a program object of an sc-memory construction isomorphic to the specified graph template by the local element identifier of the corresponding program object of the three-element construction in the specified program object of the graph template

∈ method
 ⇒ method header in method representation language*:
 [ScAddr const & operator[](std::string const & name) const noexcept(false)]
 ∈ C++
 ⇒ input argument classes of a method*:
 {
 • local identifier of an element of a three-element construction in the specified graph template
 }
 ⇒ method result class*:
 • sc-address of sc-memory element
 ⇒ class of exceptions*:
 • an element with the specified local identifier does not exist in the program object of an sc-memory construction isomorphic to the specified graph template

Method of obtaining the sc-address of an element of a program object of an sc-memory construction isomorphic to the specified graph template by the local identifier of the element of the corresponding program object of a three-element construction of the specified graph template with a preliminary check of the

specified local identifier

∈ method
 ∈ method without exceptions
 ⇒ method header in method representation language*:
 [bool Get(std::string const & name, ScAddr & outAddr) const noexcept]
 ∈ C++
 ⇒ input argument classes of a method*:
 {
 • local identifier of an element of a three-element construction in the specified graph template
 • sc-address of sc-memory element
 }
 ⇒ method result class*:
 • boolean
 }
 }

The current **Method for finding sc-memory constructions isomorphic to the specified graph template** consists of two stages: (1) *Stage of preprocessing of the graph template*, (2) *Stage of searching for sc-memory constructions isomorphic to the specified graph template*. At the same time, inside the *Method for finding sc-memory constructions isomorphic to the specified graph template*, a software iterator for finding sc-memory constructions isomorphic to the specified graph template is created, which performs the entire isomorphic search algorithm.

Software interface for information retrieval methods of Implementation of sc-memory in the ostis-platform

⊇=
 {

Software interface of graph template program object

⊇=
 {

iterator for finding sc-memory constructions isomorphic to the specified graph template

:= [ScTemplateSearch]
 ∈ C++
 ⊂ program object
 ⇒ concept specifying the specified entity*:
 {
 • local identifier of some element in some program object of a three-element construction of the specified graph template and the set of all indices of program objects of three-element constructions in the specified graph template with this element*
 • tuple of sets of indices of program objects

of three-element constructions of connectivity components in the specified graph template

- *set of indices of the highest-priority three-element constructions for the search for program objects of connectivity components in the specified graph template*
- *tuple of sets of sc-addresses of sc-memory elements corresponding to sc-connectors, such three-element constructions that are not isomorphic to the corresponding three-element constructions of the specified graph template, whose indices are equal to the position indices of sets in this oriented set*
- *tuple of sets of sc-addresses of sc-memory elements corresponding to sc-connectors of such three-element constructions that are isomorphic to the corresponding three-element constructions of the specified graph template, whose indices are equal to the position indices of the sets in this oriented set*
- *tuple of sets of sc-addresses of sc-memory elements corresponding to sc-connectors that are the second elements of the corresponding three-element constructions of found sc-memory constructions isomorphic to the specified graph template, whose indices are equal to the position indices of sets in this oriented set*
- *tuple of sets of indices of program objects of three-element constructions of the specified graph template for constructions isomorphic to it found from in sc-memory, whose indices are equal to the position indices of sets in this oriented set*
- *index of the last found sc-memory construction according to the specified graph template*
- *set of indices of all found and isomorphic constructions in sc-memory according to the specified graph template*

}
}
}

The graph template preprocessing step consists of the following intermediate processing steps:

- Addition to the relation *local identifier of some element in some program object of a three-element construction of the specified graph template and the set of all indices of program objects of three-element constructions in the specified graph template with this element** of all pairs with local element

identifiers in some program object of a three-element construction of the specified graph template and sets of indices of corresponding program objects of three-element constructions in the specified graph template. The sets of such program objects do not include those *program objects of three-element constructions* in the specified graph template, whose indices are included in the local identifier of the element itself. This element's local identifier is not the element's local identifier within the entire graph template. Such a local identifier is formed by the system itself, and not by the user of the *Method of adding a three-element construction for the specified graph template (Method of adding a five-element construction for the specified reference graph)* and consists of a local element identifier within the entire graph template and the number of the corresponding program object of the three-element structure in the specified graph template. Knowing such a local identifier of an element of some *program object of a three-element construction* of the graph template, one can quickly access other *program objects of three-element constructions* that contain this element.

- Removal from the sets of indices of *program objects of three-element constructions*, which are the second components of pairs of the relation *local identifier of some element in some program object of a three-element construction of the specified graph template and the set of all indices of program objects of three-element constructions in the specified graph template with this element** and in which there are elements that are the first components of these pairs, all such indices of *program objects of three-element constructions*, passing through which in the process of searching for constructions isomorphic according to the specified graph template can lead to a looping of the search algorithm. This pre-processing stage of the formed *graph template program object* makes it possible to eliminate transitions along such *program objects of three-element constructions* of the specified graph template as much as possible, which significantly complicating the process of isomorphic search for constructions according to the specified graph template. Since SC-code itself allows one to represent constructions of any possible configuration, it is impossible to say exactly which configurations of constructions can lead to cyclic situations when the isomorphic search algorithm for these structures is executed according to the specified graph template. A more universal algorithm for eliminating loops in the graph template can lead to significant additional time costs, since it may require a deeper syntactic analysis in the original graph template, so it is recommended to implement the conditions by which you can determine the program objects of three-

element constructions, the transition to which can lead to cyclic situations in the processing of the graph template. The elimination of loops in the graph template allows the algorithm of isomorphic search for structures on the specified graph template to more efficiently perform all the required operations on graphs, therefore it is a key step in preprocessing the original *graph template program object*. Also, this step cannot be performed together with the previous step, since in order to eliminate all loops in the graph template, it is necessary to know completely all possible transitions along this graph template.

- Search for all connectivity components in the specified graph template, that is, unrelated subgraphs in this graph, and add all *program objects of three-element constructions* corresponding to these connected components to the *tuples of sets of numbers of program objects of three-element constructions connectivity components in the specified graph template*. Thus, this makes it possible to find even such connected components that could be obtained after performing the second step of the algorithm for preprocessing the specified graph template, that is, eliminating cycles in the specified graph template. Dividing a graph template into connectivity components could be one of the solutions to the problem of eliminating a cycle in the specified graph template, however, the algorithm for isomorphic search for structures on the specified graph template is more advanced and allows one to find all three-element constructions for topics of three-element graph template constructions that have the same first or third element, so splitting the graph template into connected components is not used in the previous step.
- The last step of the *graph template preprocessing stage* is to select the connectivity components found at the previous *Pre-processing stage of graph template* of the highest-priority *program objects of three-element constructions* in the specified graph template. The highest priority program object of the three-element design is the object with a priority number equal to zero, the most non-priority object is the object with a priority number equal to six. In this case, if there are several *program objects of three-element constructions* that have the same priority number, then the object with the first (third) element, which is the sc-address of the sc-memory element, is considered to have the highest priority. has the least number of elements corresponding to outgoing (incoming) sc-connectors. As a result, a *set of numbers of the highest-priority three-element constructions for the search for program objects of the connectivity components in the specified graph template* is formed.

Thus, *graph template preprocessing stage* makes it possible to significantly simplify the processing of a graph template at the stage of searching for constructions isomorphic to it. The next ***stage of searching for sc-memory constructions isomorphic to the specified graph template*** includes the following steps:

- If the specified *tuple of program objects of sc-memory constructions isomorphic to the specified graph template* is not empty, then delete all program objects of constructions in sc-memory from it.
- If the *set of numbers of the highest-priority three-element constructions for searching program objects of connectivity components in the specified graph template* is empty, then this means that the specified *graph template* is empty. In this case, the result of the search is an empty *tuple of program objects of sc-memory constructions isomorphic to the specified graph template* and *The stage of searching for sc-memory constructions isomorphic to the specified graph template* ends with a successful result.
- Initialize *number of the last found sc-memory construction according to the specified graph template* with a value equal to zero. Set *Number of the current found sc-memory construction according to the specified graph template* equal to *number of the last found sc-memory construction according to the specified graph template*. Set *set of numbers "equivalent" program objects of constructions* to be equal to *set of numbers of the highest priority for searching program objects of three-element constructions of connectivity components in the specified graph template*. Set *set of numbers of current program objects of constructions* equal to *set of numbers of "equivalent" program objects of constructions*.
- Select the next number from *set of numbers "equivalent" program objects of constructions*. According to the received number from *tuple of program objects of three-element constructions in the specified graph template* take the corresponding *program object of three-element construction of the specified graph template* in this graph template.
- For the *selected program object of a three-element construction* in the specified graph template, find all such *program objects of three-element constructions*, (1) whose elements have the specified classes and sc-addresses the same as the classes and sc-addresses of the *elements of the selected program object of the three-element construction*, respectively, while either the first or third of their elements have the same local identifiers in the specified graph template or do not have them at all, (2) for which the corresponding replacements were not found, that is, the set located in the *tuple of sets of numbers of program objects of three-element constructions*

of the specified graph template for isomorphic constructions found using it in *sc-memory*, whose numbers are equal to the position numbers of sets in this oriented set by the number of the current found *sc-memory* construction according to the specified graph template, the numbers of the found program objects of three-element constructions do not belong, and also the numbers of which do not belong to the set of numbers of the current program objects of the constructions.

- If the received set of numbers of "equivalent" program objects of constructions is empty, then terminate this iteration of the algorithm.
- If the received set of numbers "equivalent" program objects of constructions is not empty, then choose a random number from the set of numbers "equivalent" program objects of constructions. According to the received number from tuple of program objects of three-element constructions in the specified graph template take the corresponding program object of three-element construction of the specified graph template in this graph template.
- Based on the obtained program object of a three-element construction, create an iterator for searching for three-element *sc-memory* construction. Creation of iterator for searching three-element *sc-memory* construction is done using the Method of creating iterator for searching three-element *sc-memory* construction. The parameters of the method are assigned to the elements of the specified program object of the three-element construction, while instead of classes of *sc-memory* elements corresponding to *sc-variables* (classes of *sc-memory* elements corresponding to *sc-metavariabes*), the corresponding them classes of *sc-memory* elements corresponding to *sc-constants* (classes of *sc-memory* elements corresponding to *sc-variables*). That is, for example, if some program object of a three-element construction element has a class of *sc-memory* elements corresponding to variable *sc-nodes*, then instead of it for the Method of creating a three-element *sc-memory* construction search iterator, the corresponding class of *sc-memory* elements corresponding to constant *sc-nodes* is used. Thus, when passing from program object of a three-element construction to program three-element *sc-memory* construction search iterator, the degree of variability of the elements of program object of a three-element construction decreases: classes of *sc-memory* elements corresponding to *sc-metavariabes* are converted to classes of *sc-memory* elements corresponding to *sc-variables*, and classes of *sc-memory* elements corresponding to *sc-variables* — to classes of *sc-memory* elements corresponding to *sc-constants*. If class of element of program object of

three-element construction is not a non-strict subset of class of *sc-memory* elements corresponding to *sc-metavariabes* or class of *sc-memory* elements corresponding to *sc-variables*, then the degree of variability is not reduced .

- Using the Method of moving to the next *sc-memory* construction "suitable" for the specified iterator go to the *sc-memory* construction isomorphic to the specified construction in the graph template.

Despite the wide range of tasks that can be performed using the current implementation of *isomorphic search*, there are a number of reasons why this and other implementations of *isomorphic search* should not be used:

- *isomorphic search* allows you to solve a wide range of problems if all knowledge isomorphic to the specified graph template is in the knowledge base or is missing. That is, the quality level of *isomorphic search* directly depends on the state of knowledge base. The more diverse knowledge base fragments are, the worse the performance of *isomorphic search* is;
- The cost of searching for large graph templates may be at odds with the desires of the developer or user. The larger the graph template, the more situations in which the *isomorphic search* algorithm can behave abnormally.
- Most of the problems solved with *isomorphic search* can and should be solved with three- and five-element search iterators. The simpler the method of solving problems, the fewer errors and emergency situations you can get.

VI. ADVANTAGES AND DISADVANTAGES OF THE IMPLEMENTATION OF THE INFORMATION RETRIEVAL SUBSYSTEM IN THE OSTIS-PLATFORM

The current Software interface of Implementation of *sc-memory* in the *ostis-platform* allows:

- Implementing platform-specific subsystems of the current software implementation of the *ostis-platform* to the extent necessary and sufficient, practically independently of the Implementation of *sc-memory* in the *ostis-platform*. That is, the current Software interface of Implementation of *sc-memory* in the *ostis-platform* is a way to unify access to the software Implementation of *sc-memory* in the *ostis-platform* and allows easily to replace various implementations of *sc-memory* with method representation language C++, while the Software interface of Implementation of *sc-memory* in the *ostis-platform* itself practically does not change or does not change at all.
- Implementing basic tools for designing platform-independent *ostis* systems, e.g. Implementation of *scp-interpreter*.
- Generating and expanding the Library of reusable components of Software implementation of the *ostis-*

platform with components that use the methods of *Implementation of the sc-memory in the ostis-platform* and are part of various plug-ins of the current *Software interface of Implementation of sc-memory in the ostis-platform*.

- Providing different levels of access to *Implementation of sc-memory in the ostis-platform*, including the levels of access for different users of the *Software implementation of the ostis-platform*.

It is worth noting that *Software interface of Implementation of sc-memory in ostis-platform* cannot exist separately from the current *Implementation of sc-memory in the ostis-platform*. In addition, it is part of the *Implementation of sc-memory in the ostis-platform*, that is, it is designed and developed in accordance with the implementation of the sc-memory itself. However, if necessary, it can be used for various modifications or versions of the current *Implementation of sc-memory in the ostis-platform*.

Despite the wide range of functionality of the current *Software interface of Implementation of sc-memory in the ostis-platform*, its disadvantages include the following:

- At the level of the *Software interface of Implementation of sc-memory in the ostis-platform*, there is no limit to the range of classes of sc-elements in sc-memory that can be set as arguments, for example, to the *Method of creating an sc-memory element of a given class, corresponding to an sc-node* and *Method of creating an sc-memory element of a given class, corresponding to some sc-connector*.
- Due to shortcomings in the current implementation of the agent architecture in the *software implementation of the ostis-platform* it is impossible for the *Software interface of Implementation of sc-memory in the ostis-platform* to use *Implementation of sc-memory in ostis-platform* stored as a compiled file. First of all, this is due to the fact that platform-specific agents are implemented by means that utilize creation of source files when building the entire platform. Thus compiled files remain dependent on the device where they were built.

In general, isomorphic search can be a useful tool in theoretical studies and some specialized applications, but in most cases there are better ways to work with graphs.

VII. CONCLUSION

Let us list the main ideas of this work:

- to solve information retrieval tasks in ostis-systems, the *Implementation of the information retrieval subsystem* of the current *Software implementation of the ostis-platform* is used;
- *Implementation of the information retrieval subsystem in the Software implementation of the ostis-platform* has a software interface that can be used in any platform-dependent component (subsystem);

- *Implementation of the information retrieval subsystem in the Software implementation of the ostis-platform* includes iterative methods for searching for sc-memory constructions and methods for searching for sc-memory constructions according to the specified graph template;
- to solve most information retrieval problems, it is sufficient to use iterative methods for searching for sc-memory constructions;
- the current implementation of isomorphic search is not universal and is limited to a certain set of graph templates, and also strongly depends on the state of the knowledge base.

When designing graph templates in one of the languages of the external representation of SC-code [20], one should:

- Minimize the number of cycles by splitting, for example, key constant sets into subsets that are not interconnected in this graph template. If the cycle in the graph cannot be eliminated, then leave it as it is, or reconsider the original problem for the possibility of simplifying its solution.
- Select among all those sc-constructions that can be selected by the search procedure as the first sc-construction, only the one that simplifies the work of the search procedure as much as possible for the specified one in the subject domain.
- Minimize the number of sc-constructions, the removal of which does not change the meaning of the found constructions and/or can be specified/checked later (for example, when the entity is already found and the class membership can be checked later) and/or the removal of which simplifies the choice of path search in a graph isomorphic to the specified graph template.

ACKNOWLEDGMENT

The author would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help in the work and valuable comments.

REFERENCES

- [1] D. Shunkevich, D. Koronchik, "Ontological approach to the development of a software model of a semantic computer based on the traditional computer architecture," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, pp. 75–92, 2021.
- [2] N. Zotov, "Software platform for next-generation intelligent computer systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, pp. 297—326, 2022.
- [3] A. Zagorskiy, "Factors that determine the level of intelligence of cybernetic systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, p. 13–26, 2022.

- [4] R. Reinanda, E. Meij, M. de Rijke *et al.*, “Knowledge graphs: An information retrieval perspective,” *Foundations and Trends® in Information Retrieval*, vol. 14, no. 4, pp. 289–444, 2020.
- [5] Y. WHAN KIM and J. H. Kim, “A model of knowledge based information retrieval with hierarchical concept graph,” *Journal of Documentation*, vol. 46, no. 2, pp. 113–136, 1990.
- [6] A. Parameswaran, A. D. Sarma, H. Garcia-Molina, N. Polyzotis, and J. Widom, “Human-assisted graph search: it’s okay to ask questions,” *arXiv preprint arXiv:1103.3102*, 2011.
- [7] S. Ma, J. Li, C. Hu, X. Lin, and J. Huai, “Big graph search: challenges and techniques,” *Frontiers of Computer Science*, vol. 10, pp. 387–398, 2016.
- [8] J.-A. Fernández-Madriral and J. González, “Multihierarchical graph search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 103–113, 2002.
- [9] G. J. Kowalski, *Information retrieval systems: theory and implementation*. Springer, 2007, vol. 1.
- [10] W.-C. Yeh, W. Zhu, C.-L. Huang, T.-Y. Hsu, Z. Liu, and S.-Y. Tan, “A new bat and pagerank algorithm for propagation probability in social networks,” *Applied Sciences*, vol. 12, no. 14, p. 6858, 2022.
- [11] H. Paulheim, “Knowledge graph refinement: A survey of approaches and evaluation methods,” *Semantic web*, vol. 8, no. 3, pp. 489–508, 2017.
- [12] I. Kivimäki, A. Panchenko, A. Dessy, D. Verdegem, P. Francq, H. Bersini, and M. Saerens, “A graph-based approach to skill extraction from text,” in *Proceedings of TextGraphs-8 graph-based methods for natural language processing*, 2013, pp. 79–87.
- [13] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen, and D. Wilkins, “A comparison of a graph database and a relational database: a data provenance perspective,” in *Proceedings of the 48th annual Southeast regional conference*, 2010, pp. 1–6.
- [14] Ian Robinson, Jim Webber and Emil Eifrem, *Graph databases*. O’Reilly Media, Inc., 2015.
- [15] V. Golenkov, N. Guliakina, V. Golovko, V. Krasnoproshin, “Methodological problems of the current state of works in the field of artificial intelligence,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 17–24, 2021.
- [16] N. Zotov, “Semantic theory of programs in next-generation intelligent computer systems,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 297—326, 2022.
- [17] D. Shunkevich, “Ontology-based design of hybrid problem solvers,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 101–131, 2022.
- [18] M. K. Orlov and A. P. Vasilevskaya, “Non-procedural problem-solving models in next-generation intelligent computer systems,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 161–172, 2022.
- [19] K. Bantsevich, “Metasystem of the ostis technology and the standard of the ostis technology,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 357–368, 2022.
- [20] V. Ivashenko, “General-purpose semantic representation language and semantic space,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 41–64, 2022.
- [21] S. Fortin, “The graph isomorphism problem,” 1996.
- [22] J. R. Ullmann, “An algorithm for subgraph isomorphism,” *Journal of the ACM (JACM)*, vol. 23, no. 1, pp. 31–42, 1976.
- [23] P. Foggia, C. Sansone, and M. Vento, “A performance comparison of five algorithms for graph isomorphism,” in *Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition*. Citeseer, 2001, pp. 188–199.
- [24] B. D. McKay, “Nauty user’s guide (version 2.4),” *Computer Science Dept., Australian National University*, pp. 225–239, 2007.
- [25] M. Orlov, “Comprehensive library of reusable semantically compatible components of next-generation intelligent computer systems,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, pp. 261–272, 2022.
- [26] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, “A (sub) graph isomorphism algorithm for matching large graphs,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 10, pp. 1367–1372, 2004.

Реализация информационно-поисковой подсистемы в Программной платформе ostis-систем

Зотов Н. В.

Описываются назначение и варианты реализации информационно-поисковых подсистем интеллектуальных компьютерных систем нового поколения. Данная работа является формальной спецификацией Реализации информационно-поисковой подсистемы в текущем Программном варианте реализации ostis-платформы, а также её программного интерфейса, и является продолжением серии работ по проектированию и реализации базового Программного варианта реализации ostis-платформы [1], [2]

Received 13.03.2023

Semantic Space Integration of Logical Knowledge Representation and Knowledge Processing Models

Valerian Ivashenko

Department of Intelligent Information Technologies
Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus
Email: ivashenko@bsuir.by

Abstract—This article deals with an approach to represent both logical structures and schemes, as well as logical knowledge processing models within semantic space in the form of semantic networks. Some properties of the semantic space and logical models such as topological, metric, and valuation (norm) properties are investigated. Concepts for ontological representation of classical and non-classical logical formulas, classes, and relations are proposed.

Keywords—semantic space, integration, problem-solving model integration, logical knowledge representation model, logical knowledge processing model, semantic topological space, semantic metric space, distensible set, unified semantic representation model, language key element, right proposition, uncorrupted proposition, finite set deduction

I. INTRODUCTION

The main quality of such intelligent systems as intelligent computer systems of a new generation is the ability to solve problems. Let us consider the indicators of quality which are necessary to ensure, maintain, and develop this quality. Intelligent computer systems of a new generation are classified as cybernetic systems (adaptive systems [1]). For such systems, some of the important quality indicators are [2]:

- self-learning, implemented (or automated) through self-improvement, provided by the universality of the intelligent system operating on a variety of knowledge types and (hybrid) problem-solving models, as well as their deep integration (degree of interosculation);
- interoperability that requires mutual understanding based on semantic compatibility;
- ability to coordinate their plans and intentions and coordinate them in a decentralized manner that requires the integration of plans or their parts;
- semantic compatibility (for types of knowledge and problem-solving models) that is matching between systems and concepts and requires formalization of semantic representation of information through unification, where the latter is the main indicator of

degree of convergence between intelligent computer systems and their components.

Thus, the requirement of deep integration (knowledge types and problem-solving models) is represented in the intelligent computer systems of a new generation. At the same time, intelligent computer systems of a new generation are characterized by the degree of convergence, unification, and standardization of intelligent computer systems and their components and the corresponding degree of integration (depth of integration) of intelligent computer systems and their components. Currently, there is a strong need for focusing on potentially universal (that is, capable of quickly acquiring any knowledge and skills), synergistic intelligent computer systems with “strong” intelligence, since the maximum level of cybernetic system processor quality in terms of the variety of problem-solving models interpreted by the processor of a cybernetic system is its universality, that is, its “principal ability to interpret any model for solving both intelligent and non-intelligent problems”. In order to ensure quality, the importance of moving to hybrid individual intelligent computer systems is highlighted, where convergence and integration of various problem-solving models and various knowledge types is carried out. Due to the fact that different intelligent computer systems may require different combinations of problem-solving models (models for representing and processing knowledge of various types), which have been developed by a large number at the present time, and in the development and implementation of various intelligent computer systems, the appropriate methods and tools must guarantee logical semantic compatibility of the developed components and, in particular, their ability to use common information resources, then for this, the need to unify these models is indicated. The creation of intelligent computer systems of a new generation involves the creation of an appropriate technology for integrated design and integrated support for the stages of the life cycle of these systems. It is noted that in order to create a technology for integrated design

and comprehensive support for the subsequent stages of the life cycle of intelligent computer systems of a new generation, in particular, it is necessary:

to unify the formalization of various models for representing various types of used information stored in the memory of intelligent computer systems and various models for solving intelligent problems to ensure semantic compatibility and simple automated integrability of various knowledge types and problem-solving models in intelligent computer systems. To do this, it is proposed to develop a basic universal abstract model for the representation and processing of knowledge, which ensures the implementation of various problem-solving models.

Let us consider the stages of solving these problems in the aspect of integrating logical models of representation and knowledge processing. The need for this consideration is caused by the need of applying logical problem-solving models in intelligent computer systems of a new generation (including knowledge-driven systems), while ensuring the quality of knowledge in accordance with the problems of knowledge management.

II. KNOWLEDGE INTEGRATION AND SEMANTIC SPACE MODELS

In order to solve the problem of unifying the formalization of various models for representing various types of information used, a model of a unified semantic representation of knowledge [3] has been developed, as well as models for representing data in the form of texts of generalized formal languages [4] and processing generalized strings (and lists) for knowledge-driven systems [5]. Based on and in accordance with the model of unified knowledge representation, a family of sc-languages [1], [3], [6] has been developed, to clarify the semantics of which a model of event (distensible) sets [4] has been developed and an ontological model of spatio-temporal relations of events and phenomena for knowledge processing operations has been proposed.

In order to ensure the integration of knowledge and quality assurance in the process of knowledge integration, models for the specification and integration of knowledge are proposed. Solving the problem of knowledge integration allows considering and studying by formal means the semantic neighborhoods of sc-language text elements, the key elements of sc-languages, and studying the similarity of structures that are formed as a result of integration. Based on the proposed models, a meta-model of the semantic space was developed [4], within which it is possible to study the semantic space [7]–[14] and consider semantic subspaces of various types.

The system of transitions from texts of sc-languages to topological space is studied. Below is a fragment of the ontology that describes the types of topological spaces and inclusion relations of topological spaces of various types [?], [15].

generalized sc-tuple
 $:=$ [non-empty sc-set]

generalized sc-relation
 $:=$ [sc-set of non-empty sc-sets]
 \Rightarrow *explanation**:
 [A generalized sc-relation is a sc-set of generalized sc-tuples.]

binary sc-relation
 \Rightarrow *explanation**:
 [A binary sc-relation is an sc-set of sc-pairs (or generalized sc-tuples to which there are two different memberships of sc-elements or the same sc-element).]

nodal sc-pair
 \Rightarrow *explanation**:
 [A nodal sc-pair is an sc-pair that cannot be denoted by a membership sc-arc (positive, negative, or fuzzy).]

slot sc-relation
 \Rightarrow *explanation**:
 [A slot sc-relation is a binary sc-relation (an sc-set of (oriented) sc-pairs) whose elements are not nodal sc-pairs.]

membership phenomenon
 \Rightarrow *explanation**:
 [A membership phenomenon is a set of phenomena each of which is a slot sc-relation, while any sc-arc of permanent non-membership does not belong permanently to each of them.]

becoming*
 \Rightarrow *explanation**:
 [becoming* is a binary sc-relation between events (states) or phenomena.]

immediately before'
 \Rightarrow *first domain**:
*becoming**
 \Rightarrow *second domain**:
established event or phenomenon

immediately after'
 \Rightarrow *first domain**:
*becoming**
 \Rightarrow *second domain**:
constitutive event or phenomenon

continuance*
 \Rightarrow *explanation**:
 [The continuance* is the transitive closure of the

sc-relation of becoming.]

earlier'

- ⇒ *first domain**:
*continuance**
- ⇒ *second domain**:
early event or phenomenon

later'

- ⇒ *first domain**:
*continuance**
- ⇒ *second domain**:
later event or phenomenon

sc-structure*

- ⇒ *explanation**:
[A sc-structure* is an sc-set that contains a non-empty support sc-subset (the set of primary elements of the sc-structure*.)]

sc-structure'

- ⇒ *first domain**:
*sc-structure**
- ⇒ *second domain**:
non-empty sc-set

support of sc-structure'

- ⇒ *first domain**:
*sc-structure**
- ⇒ *second domain**:
non-empty sc-set

elementarily represented sc-set'

- := [elementarily represented element']
- ⇒ *explanation**:
[An elementarily represented element' is an element of an sc-structure* and an sc-set all of whose elements are elements of an sc-structure*.]

full-connectively represented sc-set sc-set'

- := [full-connectively represented element']
- ⇒ *explanation**:
[A full-connectively represented sc-set element' is an element of an sc-structure* an sc-set all of whose elements and all its memberships are elements of an sc-structure* or an sc-arc that is an elementarily-represented element' of this sc-structure*.]

fully represented sc-set'

- := [fully represented element']
- ⇒ *explanation**:
[A fully represented element' is a full-connectively represented element' of an sc-structure* with any its element that is not an sc-arc outgoing

from it connected by a membership sc-arc or a non-membership sc-arc belonging to this sc-structure*.]

sc-tuple'

- ⇒ *explanation**:
[A sc-tuple' is a sc-tuple' is a full-connectively represented element' of sc-structure* that is an sc-tuple and belongs to the sc-relation' of this sc-structure*.]

sc-relation'

- ⇒ *explanation**:
[A sc-relation' is a full-connectively represented element' of sc-structure* being a sc-relation whose elements are all sc-tuples' of this sc-structure*.]

sc-class'

- ⇒ *explanation**:
[A sc-class' is a full-connectively represented element' of sc-structure* all of whose elements are members of an sc-structure* that is neither an sc-relation' nor an sc-tuple' of that sc-struct*.]

entitive closure*

- ⇒ *explanation**:
[An entitive closure* is the smallest superset* (structure*) in which each element is elementarily represented'.]

entitive closure'

- ⇒ *first domain**:
*entitive closure**
- ⇒ *second domain**:
entitive closure

support of entitive closure'

- ⇒ *first domain**:
*entitive closure**
- ⇒ *second domain**:
non-empty sc-set

substantial closure*

- ⇒ *explanation**:
[A substantial closure* is the smallest superset* (structure*) in which each element is a full-connectively represented element']

substantial closure'

- ⇒ *first domain**:
*substantial closure**
- ⇒ *second domain**:
substantial closure

support of substantial closure [']

- ⇒ *first domain**:
*substantial closure**
- ⇒ *second domain**:
non-empty sc-set

sc-relation of similarity by slot relations*

- ⇒ *explanation**:
[A similarity sc-relation by slot sc-relations* is a sc-relation that is reflexive by these slot relations, i.e. for any element included in the tuple of this sc-relation under one of the slot sc-relations, there is a tuple of this sc-relation in which it enters under each of these slot sc-relations.]

sc-relation of similarity by slot relations [']

- ⇒ *first domain**:
*sc-relation of similarity by slot relations**
- ⇒ *second domain**:
sc-relation of similarity by slot relations

slot relations of similarity sc-relation [']

- ⇒ *first domain**:
*sc-relation of similarity by slot relations**
- ⇒ *second domain**:
slot relations of similarity sc-relation

sc-relation of semantic similarity by slot relations*

- ⇒ *explanation**:
[A semantic similarity sc-relation by slot relations* is a similarity sc-relation by slot relations* si and sj, in which each element under the slot sc-relation si can be converted to an element of the syntactic type of the element under the slot sc-relation sj; two incident sc-elements under the slot sc-relation si, within this sc-relation of semantic similarity correspond to the incident elements, respectively, under the slot sc-relation sj.]

sc-relation of semantic similarity by slot relations [']

- ⇒ *first domain**:
*sc-relation of semantic similarity by slot relations**
- ⇒ *second domain**:
sc-relation of semantic similarity by slot relations

slot relations of semantic similarity sc-relation [']

- ⇒ *first domain**:
*sc-relation of semantic similarity by slot relations**
- ⇒ *second domain**:
slot relations of semantic similarity sc-relation

connected sc-structure*

- ⇒ *explanation**:
[A connected sc-structure* is a sc-structure* that is connected.]

connected sc-structure [']

- ⇒ *first domain**:
connected sc-structure
- ⇒ *second domain**:
connected non-empty sc-set

support of connected sc-structure [']

- ⇒ *first domain**:
*connected sc-structure**
- ⇒ *second domain**:
non-empty sc-set

semantic similarity of sc-structures*

- ⇒ *explanation**:
[A semantic similarity of sc-structures* connects the sc-set of sc-structures* with the sc-structure* sc-relation of semantic similarity by slot sc-relations si, sj so that for each sc-structure* from the sc-set there is its an element and a tuple of this sc-relation of similarity, in which it is included under the slot sc-relation si, and under the slot sc-relation sj there is an element of the sc-structure *, also for each element of the sc-structure there is a tuple of this sc-relation of similarity, in which it enters under the slot sc-relation sj, and under the slot sc-relation si enters an element of the sc-structure* from the sc-set.]

sc-relation of semantic similarity of sc-structures [']

- ⇒ *first domain**:
*semantic similarity of sc-structures**
- ⇒ *second domain**:
*sc-relation of semantic similarity by slot relations**

semantic similarity of sc-structures [']

- ⇒ *first domain**:
*semantic similarity of sc-structures**
- ⇒ *second domain**:
*sc-structure of semantic similarity of sc-structures**

sc-structure of semantic similarity of sc-structures [']

- ⇒ *first domain**:
*sc-structure of semantic similarity of sc-structures**
- ⇒ *second domain**:
sc-structure of semantic similarity of sc-structures

set of semantically similar sc-structures'
 \Rightarrow first domain*:
sc-structure of semantic similarity of sc-structures*
 \Rightarrow second domain*:
set of semantically similar sc-structures

semantic continuous similarity of sc-structures*
 \Rightarrow explanation*:
[A semantic continuous similarity of sc-structures* connects an sc-set of sc-structures* with a connected sc-structure* sc-relation of semantic similarity by slot sc-relations s_i, s_j so that for each sc-structure* from sc-set find its element and the tuple of this similarity sc-relation, in which it enters under the slot sc-relation s_i , and under the slot sc-relation s_j there is an element of the connected sc-structure*, also for each element of the connected sc-structure there is a tuple of this sc- a similarity relation in which it enters under the slot sc-relation s_j , and under the slot sc-relation s_i the element of the sc-structure* from the sc-set enters.]

sc-relation of semantic continuous similarity of sc-structures'
 \Rightarrow first domain*:
semantic continuous similarity of sc-structures*
 \Rightarrow second domain*:
sc-relation of semantic continuous similarity by slot relations*

semantic continuous similarity of sc-structures*'
 \Rightarrow first domain*:
semantic continuous similarity of sc-structures*
 \Rightarrow second domain*:
sc-structure of semantic continuous similarity of sc-structures*

sc-structure of semantic continuous similarity of sc-structures'
 \Rightarrow first domain*:
sc-structure of semantic continuous similarity of sc-structures*
 \Rightarrow second domain*:
sc-structure of semantic continuous similarity of sc-structures

set of semantically continuously similar sc-structures'
 \Rightarrow first domain*:
sc-structure of semantic continuous similarity of sc-structures*
 \Rightarrow second domain*:
set of semantically continuously similar sc-structures

key query'
 \Rightarrow first domain*:
key query*
 \Rightarrow second domain*:
key query
 \Rightarrow explanation*:
[A key query' is a search-verify query (from one known element) that is executed from at least one element and is not executed from at least one element.]

element of key query'
 \Rightarrow first domain*:
key query*
 \Rightarrow second domain*:
element of key query

minimal key query'
 \subset key query'
 \Rightarrow explanation*:
[A minimal key query is a key query that finds sc-subsets of the element sets found by all other key queries that have the same domains of known satisfiability and non-satisfiability elements.]

element of minimal key query'
 \Rightarrow first domain*:
minimal key query*
 \Rightarrow second domain*:
element of minimal key query

full semantic neighborhood of an element*
 \Rightarrow explanation*:
[A full semantic neighborhood of an element* is all elements found by minimal key queries from this element (taking into account disjunctive search and negation of search).]

full semantic neighborhood of an element'
 \Rightarrow first domain*:
full semantic neighborhood of an element*
 \Rightarrow second domain*:
full semantic neighborhood of an element

element of full semantic neighborhood'
 \Rightarrow first domain*:
full semantic neighborhood of an element*
 \Rightarrow second domain*:
element of full semantic neighborhood

introspective key element'
 \Rightarrow explanation*:
[An introspective (basic) key element is an element of a set (from the class of the smallest such sets) of elements such that any full semantic

neighborhood of any element is an sc-subset of the union of their full semantic neighborhoods.]

topological space

⇒ *explanation**:
 [A topological space is a set with a set (family) of (open) subsets defined over it, including the set itself and the empty set. For any subset of the family, the result of the union belongs to the family, and for any finite subset of the family, the result of the intersection also belongs to the family. The complements of the sets of a family to the largest of the sets are called closed sets.]

topological space of connector incidence closure

⇒ *explanation**:
 [A topological space of connector incidence closure on a set of sc-elements is a topological space, all closed sets of which contain all sc-elements of this set, to which there is a route along oriented connectives of the incidence relation of connectors.]

⇒ *note**:
 [In the general case, it does not satisfy the Tikhonov separability axiom. The pragmatics of considering such spaces is determined by the operations of removing sc-elements and connectors with which they are incident. Deleting an sc element requires deleting all connectors whose closure of any open neighborhood it belongs to.]

topological subspace of connector incidence closure'

⇒ *first domain**:
*inclusion of topological spaces of connector incidence closure**
 ⇒ *second domain**:
topological space of connector incidence closure

topological superspace of connector incidence closure'

⇒ *first domain**:
*inclusion of topological spaces of connector incidence closure**
 ⇒ *second domain**:
topological space of connector incidence closure

topological space of syntactic closure

⇒ *explanation**:
 [A topological space of syntactic closure on a set of sc-elements is a topological space, all closed sets of which contain all sc-elements of this set, to which there is a route along oriented connectives of the incidence relation.]

⇒ *note**:
 [In the general case, it does not satisfy the Kolmogorov separability axiom. Syntactic closure

can be singled out as the basis of closed sets of a topological space, however, due to the possibility of drawing arcs from any sc-node to any, in the final case (as a result of the process of eliminating non-factors), such a space is a trivial (antidiscrete) space. The union relation of topological spaces of syntactic closure is not algebraically closed on the set of topological spaces of syntactic closure. For the same reason, for any incomplete topological space of syntactic closure, one can consider a topological space of syntactic closure whose support is a superset of the support of the former and which does not preserve closed sets. In this sense, the topology based on syntactic closure is not stable with respect to the processes of knowledge formation and its consideration is not pragmatically justified. The topology of the complete topological space of syntactic closure is antidiscrete (trivial). Thus, for a complete topological space of syntactic closure, all topological subspaces of syntactic closure have antidiscrete (trivial) topology.]

topological space of entitive closure

⇒ *explanation**:
 [The topological space of an entitive closure on the set of sc-elements is a topological space all of whose closed sets are entitive closures.]

⇒ *note**:
 [In the general case, it does not satisfy the Tikhonov separability axiom. An entitive closure can be considered as a support of a topological (sub)space of entitive closure. The topological space of entitive closure preserves the closed sets of any topological spaces of entitive closure whose support is a subset of its support and an entitive closure. Such spaces form a structure of topological subspaces – topological superspaces of entitive closure. The topology of spaces in this structure is diverse.]

topological subspace of entitive closure'

⇒ *first domain**:
*inclusion of topological spaces of entitive closure**
 ⇒ *second domain**:
topological space of entitive closure

topological superspace of entitive closure'

⇒ *first domain**:
*inclusion of topological spaces of entitive closure**
 ⇒ *second domain**:
topological space of entitive closure

topological space of substantial closure

- ⇒ *explanation**:
[The topological space of a substantial closure on the set of sc-elements is a topological space all of whose closed sets are substantial closures.]
- ⇒ *note**:
[In the general case, it does not fit the axiom of separability according to Tikhonov. An substantial closure can be considered as a support of a topological (sub)space of substantial closure. The topological space of a substantial closure preserves closed any topological spaces of a substantial closure whose support is a subset of its support and a substantial closure. Such spatial formations of topological subspaces are topological superspaces of substantial closure. Topology of spaces in this vast area.]

topological subspace of substantial closure'

- ⇒ *first domain**:
*inclusion of topological spaces of substantial closure**
- ⇒ *second domain**:
topological space of substantial closure

topological superspace of substantial closure'

- ⇒ *first domain**:
*inclusion of topological spaces of substantial closure**
- ⇒ *second domain**:
topological space of substantial closure

There is a possible transition from sc-structures to manifolds and topological spaces by reducing sc-structures to graph structures. The issues of reducing sc-structures to graph structures and further towards manifolds and topological spaces are considered in detail in [4] (see Fig. 1).

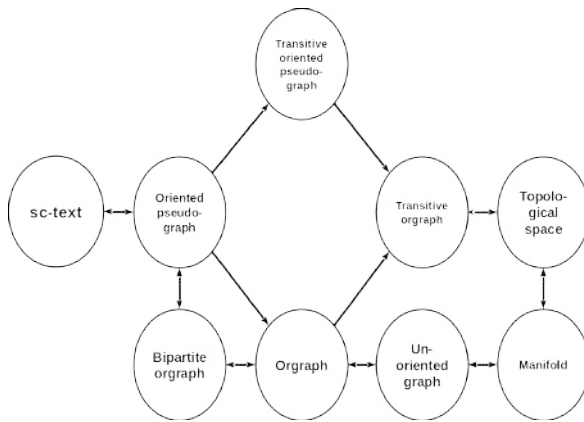


Figure 1. Transformations for sc-texts, graphs, manifolds and topological spaces.

Types of metric and pseudometric semantic subspaces

and normalized semantic subspaces can also be represented. Further, we will consider such metric spaces as finite subspaces with full-connectively represented sc-elements, in particular.

metric

- ⇒ *explanation**:
[A metric is a function of two arguments that takes values on a (linearly) ordered support of the group, is non-negative, is equal to the neutral element (zero) only when the arguments are equal, is symmetric, satisfies the triangle inequality.]

metric space

- ⇒ *explanation**:
[A metric space is a set with a function of two arguments defined on it, which is a metric that takes values on the ordered support of the group.]

metric finite syntactic space

- ⇒ *explanation**:
[A metric finite syntactic space of the SC-code is a metric space with a finite support, whose elements are designations (sc-elements), and the value of the metric can be determined through the incidence relations of the elements without taking into account their semantic type.]

metric finite syntactic subspace'

- ⇒ *first domain**:
*inclusion of metric finite syntactic spaces**
- ⇒ *second domain**:
metric finite syntactic space

metric finite syntactic superspace'

- ⇒ *first domain**:
*inclusion of metric finite syntactic spaces**
- ⇒ *second domain**:
metric finite syntactic space

metric finite semantic space

- ⇒ *explanation**:
[A metric finite semantic space of the SC-code is a metric space with a finite support, whose elements are designations (sc-elements), and the value of the metric cannot be determined through the incidence relations of elements without taking into account their semantic type.]

metric finite semantic subspace'

- ⇒ *first domain**:
*inclusion of metric finite semantic spaces***

⇒ *second domain**:
metric finite semantic space

metric finite semantic superspace'

⇒ *first domain**:
inclusion of metric finite semantic spaces*
⇒ *second domain**:
metric finite semantic space

A metric finite syntactic space can be constructed by [4] according to the string processing model and metric definitions given in [5].

pseudometric

⇒ *explanation**:
[A pseudometric is a function of two arguments that takes values on a (linearly) ordered group support, is non-negative, symmetric, and satisfies the triangle inequality.]

pseudometric space

⇒ *explanation**:
[A pseudometric space is a set with a function of two arguments defined on it, which is a pseudometric [16] taking values on the ordered support of the group.]

pseudometric finite semantic space

⇒ *explanation**:
[A pseudometric finite semantic space of the SC-code is a pseudometric space with a finite support whose elements are designations (sc-elements), and the value of the pseudometric cannot be determined through the incidence relations of elements without taking into account their semantic type.]

Some models of more complex structures that take into account non-factors [17] associated with space-time have been successfully proposed in [4]. The proposed models rely on a representation capable of expressing the semantics of variable notation and operational semantics by extended means of the alphabet. To build such models, in addition to the extended alphabet tools, it is proposed to rely on models that describe the processes of integration and formation of knowledge [18], on knowledge specification tools [3], [4], focused on consideration of finite structures, which allow proceeding with consideration of complex metric relationships within the semantic space meta-model (see Fig. 2).

The possibility of considering the metric in the semantic space allows speaking about the semantic metric, which, along with activity, scaling, interpretability, and the presence of a complex structure and coherence, is a hallmark of knowledge.

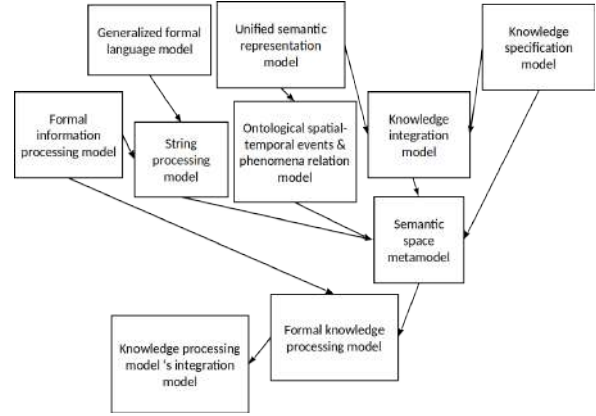


Figure 2. Models providing integration.

semantic metric

:= [semantic similarity]

⇒ *explanation**:

[Semantic metric is a metric defined on signs and quantitatively expressing the proximity of their meanings.]

In addition to factual knowledge (facts), rules are used in knowledge bases. Within logical models of knowledge processing, rules are represented as logical formulas. Thus, the transition to the integration of such types of knowledge as (logical) rules allows talking about the integration of knowledge processing models (problem-solving models).

III. INTEGRATION OF LOGICAL PROCESSING MODELS AS PROBLEM SOLVING MODELS

In order to solve the problem of integrating problem-solving models, the concept of a formal model for knowledge processing is proposed, which is a development for the concept of a formal model of information processing. The approach is used in the works of V. Kuzmitsky [19] and A. Kalinichenko [20]. A meta-model for the integration of formal models of knowledge processing is proposed.

The integration of knowledge processing models boils down to the following steps:

- For each state of the integrating model, its one-to-one (i) representation is constructed in the model of the unified semantic representation of knowledge.
- Next, a mapping π of this representation to a set of sc-texts immersed in a metric semantic subspace is constructed, and a one-to-one mapping i of operations $i(\rho)$ of this model to operations ρ on sc-texts from this set is constructed, so that:

$$i \circ i^{-1} \subseteq I = \{ \langle x, x \rangle \mid \exists y \langle x, y \rangle \in i^{-1} \cup i \}$$

$$i^{-1} \circ i \subseteq I$$

$$\forall \rho (i^{-1} \circ i \circ \rho \circ i \circ i^{-1} = \rho)$$

$$\forall \rho \exists i (\rho) (\pi^{-1} \circ i^{-1} \circ \rho \subseteq i \circ \pi \circ i (\rho))$$

$$\forall \rho \exists i (\rho) (i \circ \pi \circ i (\rho) \subseteq \rho \circ i \circ \pi)$$

$$\forall \rho \exists i (\rho) (\rho = i \circ \pi \circ i (\rho) \circ \pi^{-1} \circ i^{-1})$$

$$\forall \rho \exists i (\rho) (i (\rho) = \pi^{-1} \circ i^{-1} \circ \rho \circ i \circ \pi)$$

- Syntactic relations are distinguished on the elements of sc-texts.
- Interpretation functions are built on the states of the original model (in projective semantics) or on their representation in sc-texts (in reflexive semantics).
- The metric is set in accordance with the metric of the metric semantic subspace.
- In addition to the specified requirements, additional requirements τ and σ can be specified in accordance with a given scale on the set of states of the integrating information processing model: bijection (trivial order), out-degree, in-degree, etc.

$$\forall \rho \exists i (\rho) (\rho \circ \tau = i \circ \pi \circ i (\rho) \circ \pi^{-1} \circ i^{-1})$$

$$\forall \rho \exists i (\rho) (i (\rho) \circ \sigma = \pi^{-1} \circ i^{-1} \circ \rho \circ i \circ \pi)$$

It should be noted that in the previous article [21], the mapping requirements were considered to be quite strong ($\tau = I$ and $\sigma = I$):

$$\forall \rho \exists i (\rho) (\rho \subseteq i \circ \pi \circ i (\rho) \circ \pi^{-1} \circ i^{-1})$$

$$\forall \rho \exists i (\rho) (i (\rho) \subseteq \pi^{-1} \circ i^{-1} \circ \rho \circ i \circ \pi)$$

The current text contains proposal for weakening these requirements. Other additional requirements (including the quantitative properties of the information) may also be taken into account.

Let us consider some examples (Fig. 3–25).

From the point of view of topological properties, for each state of the model, there is its topological closure with respect to the set of operations. Moreover, these topological properties are preserved during integration. Thus, integration is a continuous mapping. However, for classical logical models of information processing, it is known that the closure with respect to deducibility is not topological closure (not additive):

$$[S] \cup [T] \neq [S \cup T].$$

The seeming contradiction can be resolved if we notice that in the first case, the elements of the closure are

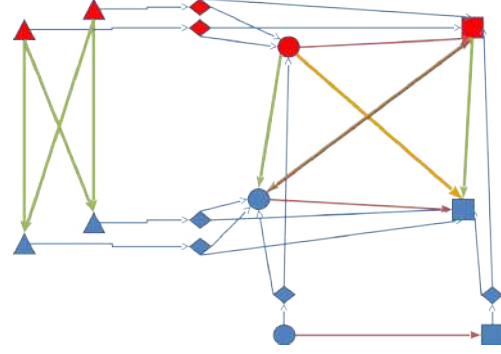


Figure 3. The reconvergent integration of non-deterministic knowledge processing operation as non-deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as non-deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

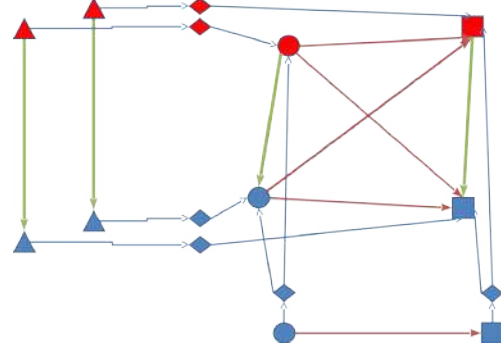


Figure 4. The convergent integration of deterministic knowledge processing operation as deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as non-deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

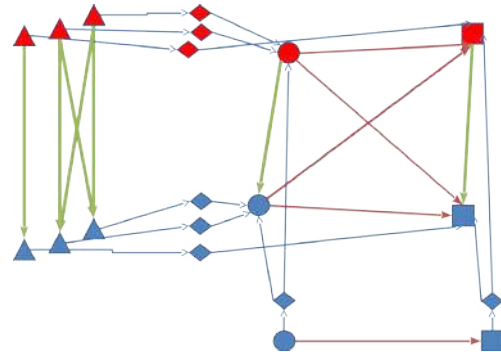


Figure 5. The reconvergent integration of non-deterministic knowledge processing operation as deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

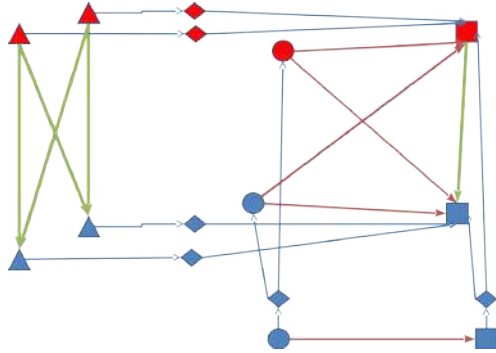


Figure 6. The convergent integration of non-deterministic knowledge processing operation as deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

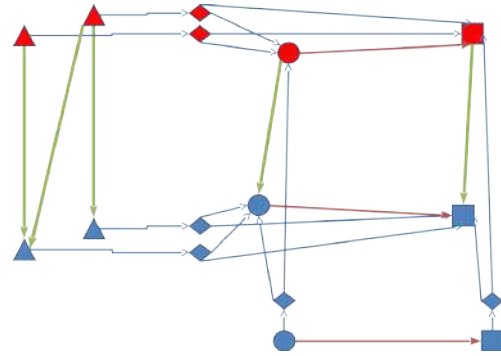


Figure 9. The asymmetrical reconvergent integration of non-deterministic knowledge processing operation as deterministic one (green lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one (red lines).

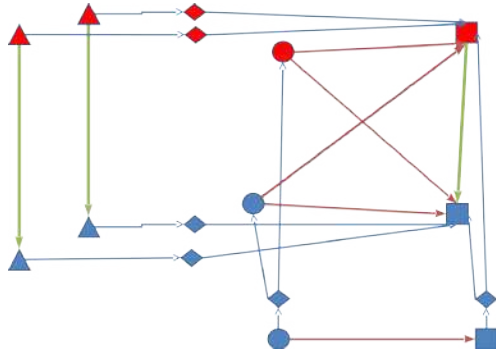


Figure 7. The convergent integration of non-deterministic knowledge processing operation as deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

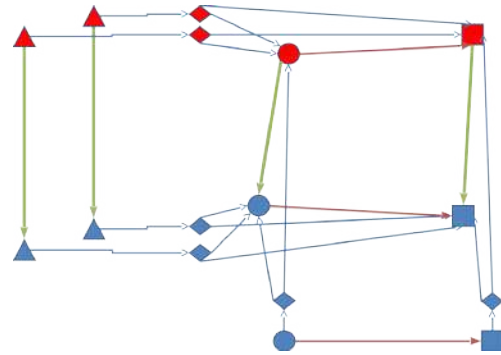


Figure 10. The reconvergent integration of deterministic knowledge processing operation as deterministic one (green lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one (red lines).

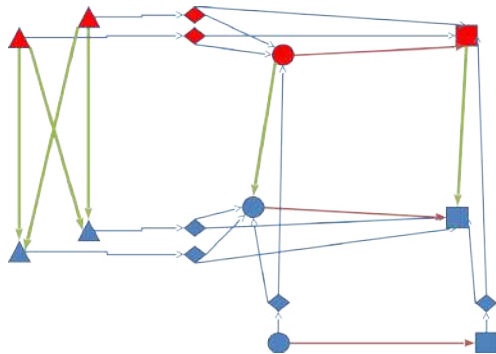


Figure 8. The reconvergent integration of non-deterministic knowledge processing operation as deterministic one ((green) vertical lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one ((red) horizontal lines). Rhombuses are subtext (substates). Triangles and the bottom blue disk and square are states of integrating models. Others disks and squares are the states (text) of the integrated model.

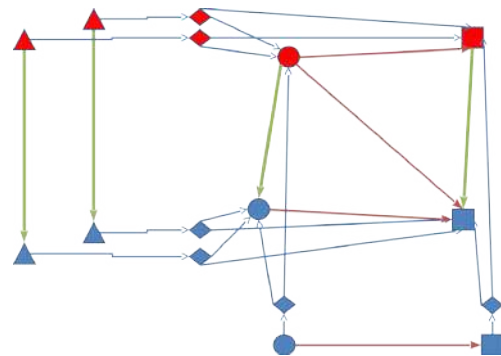


Figure 11. The reconvergent integration of deterministic knowledge processing operation as deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

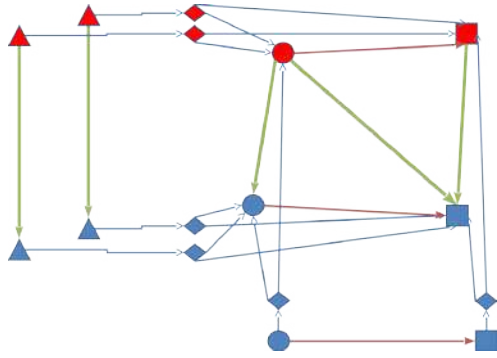


Figure 12. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the divergent integration of deterministic knowledge processing operation as deterministic operation one (red lines).

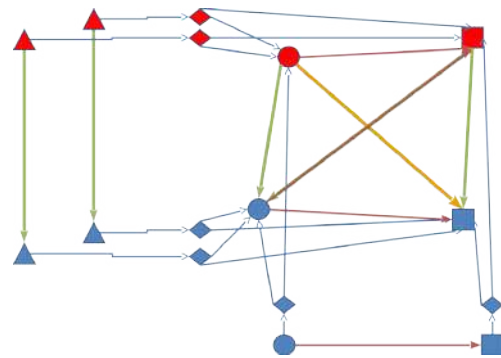


Figure 15. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as non-deterministic one (yellow and green lines) with the (symmetrical) divergent integration of deterministic knowledge processing operation as non-deterministic operation one (yellow and red lines).

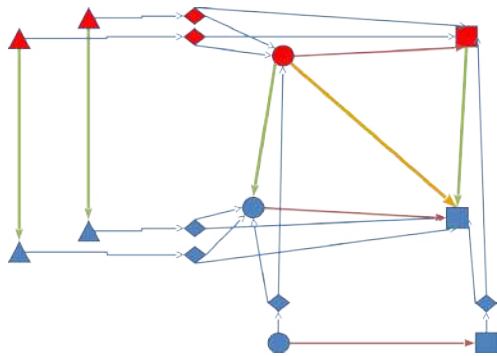


Figure 13. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (yellow (diagonal) and green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (yellow (diagonal) and red lines).

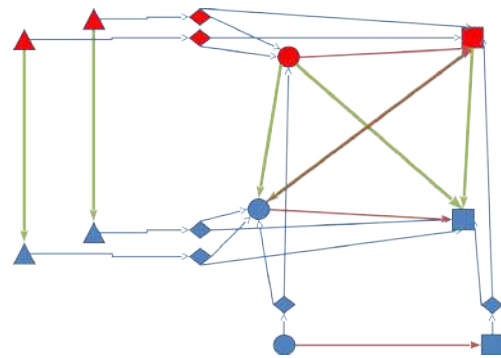


Figure 16. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

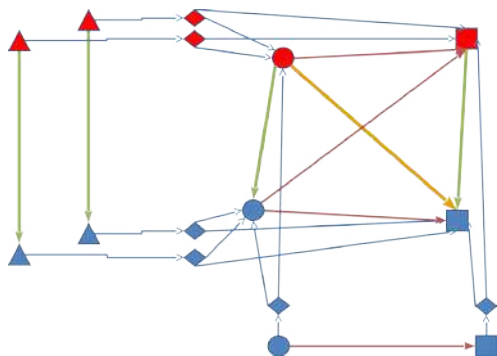


Figure 14. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as non-deterministic one (yellow and green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (yellow and red lines).

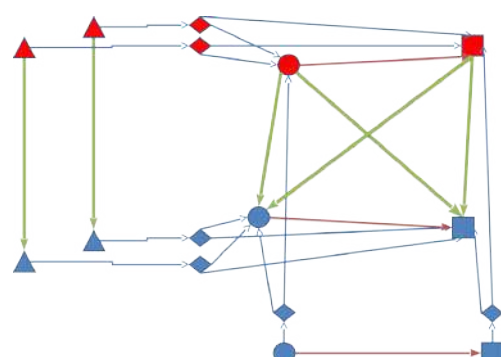


Figure 17. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the (symmetrical) divergent integration of deterministic knowledge processing operation as deterministic operation one (red lines).

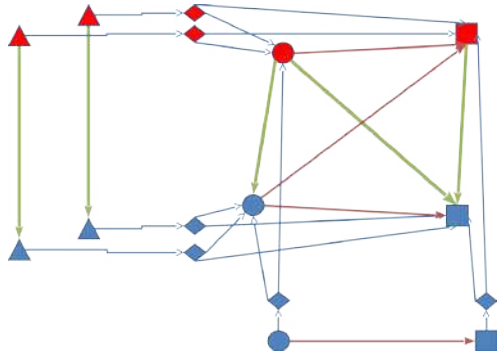


Figure 18. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

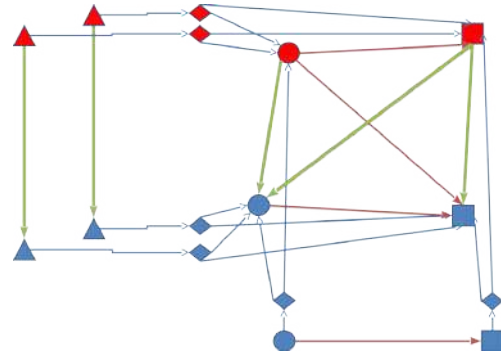


Figure 21. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

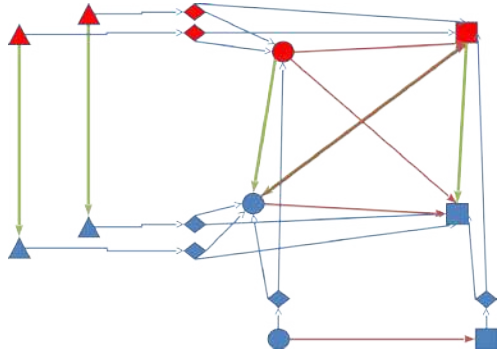


Figure 19. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the (symmetrical) divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

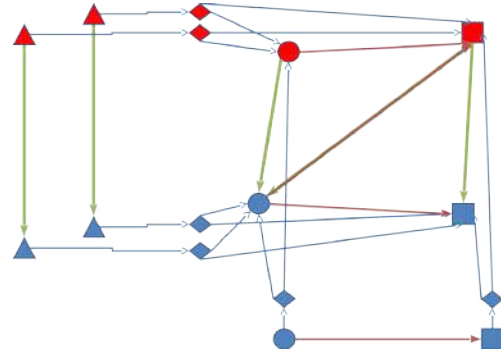


Figure 22. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

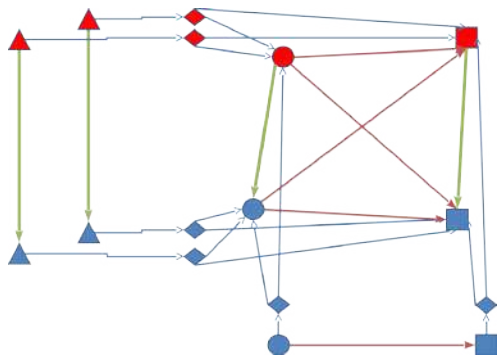


Figure 20. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as deterministic one (green lines) with the (symmetrical) divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

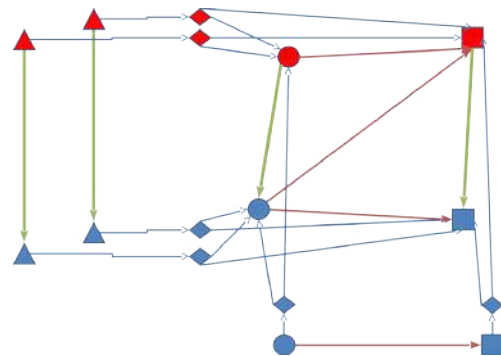


Figure 23. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as deterministic one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

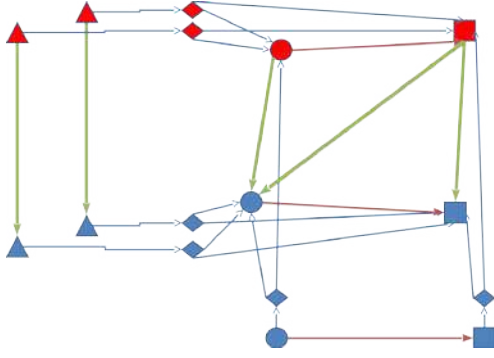


Figure 24. The asymmetrical reconvergent integration of deterministic knowledge processing operation as non-deterministic one (green lines) with the (symmetrical) divergent integration of deterministic knowledge processing operation as deterministic operation one (red lines).

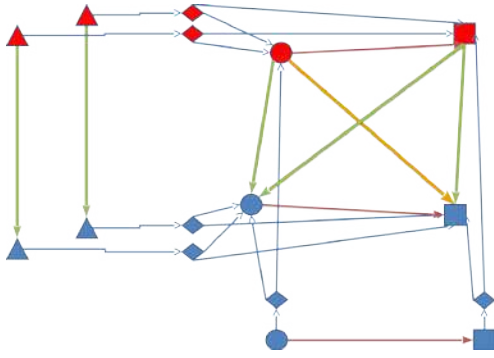


Figure 25. The (symmetrical) reconvergent integration of deterministic knowledge processing operation as non-deterministic operation one (green lines) with the asymmetrical divergent integration of deterministic knowledge processing operation as non-deterministic operation one (red lines).

states (sets of formulas), and in the second case, they are formulas.

$$\begin{aligned} & \{(A \rightarrow B), (B \rightarrow C)\} \vdash \{(A \rightarrow B), (B \rightarrow C), (A \rightarrow C)\} \\ & \{(D \rightarrow B), (B \rightarrow E)\} \vdash \{(D \rightarrow B), (B \rightarrow E), (D \rightarrow E)\} \\ & \{(A \rightarrow B), (B \rightarrow C)\} \cup \{(D \rightarrow B), (B \rightarrow E)\} \vdash \\ & \{(A \rightarrow B), (B \rightarrow C), (A \rightarrow C), (D \rightarrow C), \\ & (D \rightarrow B), (B \rightarrow E), (D \rightarrow E), (A \rightarrow E)\} \end{aligned}$$

For classical logics and logical models of knowledge processing, it is possible to naturally introduce a metric on sets of literal conjuncts (of a given length), if we take a finite subject domain and accept the assumption (hypothesis) of a closed world, then the metric is introduced as the sum of exclusive-OR from each pairs of matching literals.

$$\mu(\langle x, y \rangle) = \sum_{i=1}^n x_i \vee y_i$$

Moreover, if the conjuncts define a set-ring algebra, then we can speak of a normed space and a norm (valuation) over a vector space, where the field is GF(2). Any (meaningful) proposition over this domain, except for the identically false one, can be represented as (PDFN). Then for PDFN we get a metric vector. The proposition

is true if and only if the metric vector contains 0. For PCNF, there is no 0.

$$\mu(\langle x, y \rangle) = \min_j \sum_{i=1}^n x_{ij} \vee y_{ij}$$

In addition to (null-local predicates) of literals (constants), unary (semantic) predicates for the type of an element of an sc-text and unary and binary (syntactic) predicates for element incidence tuples (variables) can be considered (by analogy of relation algebra [22], [23]). In the case of search by pattern (homomorphism), the metric is calculated on multisets. Possible task is to minimize (maximize) the metric from the pragmatic view of logics. Associated with the search for a relevant structure, this task is of practical importance in reference and testing (checking) (dialog) systems [21], [24]. Also, metrics with other quantor elimination approaches ([23], [25]) can be used for logical inference and theorem proving purposes. The system of natural inference and sequent calculus consider finite sets of formulas. One of the algorithms for solving inference problems is a conflict-driven (contradiction-driven) clause learning (CDCL) [26]. These techniques seemed to be promising. To express complex patterns and regularities, it is possible to construct a metatheory using metastructures. For this, meta-relations and modal operators are introduced. Such a formalism allows describing the complex introspective reasoning characteristic of modal logics (see Fig. 26–28 for the sages hat puzzle solution [27]).

Applied logics [27]–[32] consider applications of classical logic to abstract and subject domains describing reality: logical theories about equality and order relations [28], [29], logical theories of arithmetic [28], logical theories of time [27], [31], logical proof theories [28], [30], graph and geometric theories [32], theories of natural and social systems [29].

Classification of logical theories corresponds to the classification of subject domains. Let us consider some concepts and examples that are considered within applied logics.

slot binary relation

\Rightarrow note*:
[slot binary relation is a slot sc-relation that is a set.]

non-slot binary relation

\Rightarrow note*:
[non-slot binary relation is a binary sc-relation that is a set but is not a slot sc-relation.]

irreflexive slot binary relation

\subset irreflexive binary relation
 \Rightarrow note*:

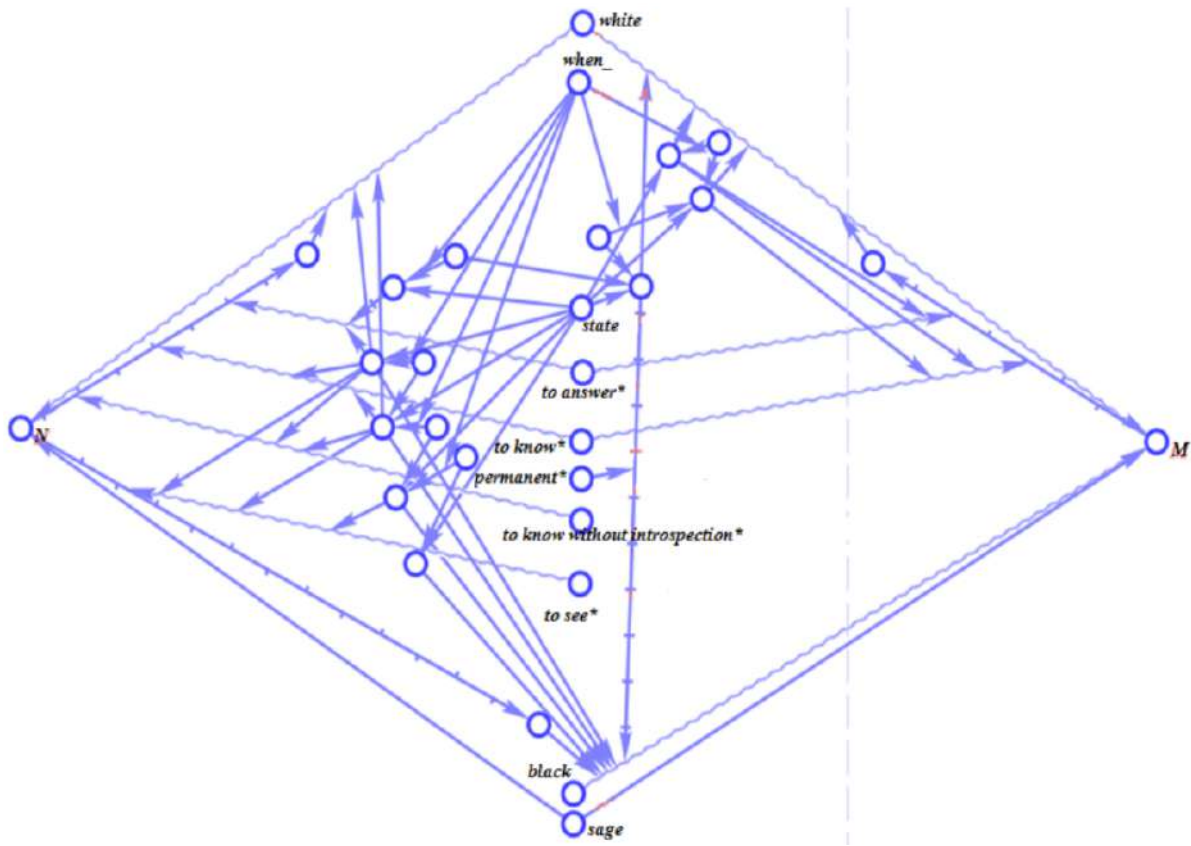


Figure 26. Example of solving the puzzle with sages hats for one of two sages.

[An irreflexive slot binary relation is a slot (binary) relation. Any tuple of which is not denoted by a loop arc (an arc with the same beginning and end).]

[A reflexive non-slot binary relation is a non-slot binary sc-relation. For any tuple's element of which, there is a tuple with two different memberships of this element.]

irreflexive non-slot binary relation

\subset irreflexive binary relation
 \Rightarrow note*:

[An irreflexive non-slot binary relation is a non-slot binary sc-relation. Different memberships of any tuple of which are the memberships of different elements.]

transitive slot binary relation

\subset transitive binary relation
 \Rightarrow note*:

[A transitive slot binary relation is a slot binary relation. For any two tuples of which, the end of one of them is the beginning of the second, there is a link whose beginning is the beginning of the first link, and the end is the end of the second link.]

reflexive slot binary relation

\subset reflexive binary relation
 \Rightarrow note*:

[A reflexive slot binary relation is a slot binary sc-relation. For any tuple's element of which, there is a its tuple denoted by a loop arc (an arc with the same beginning and end).]

transitive non-slot binary relation

\subset transitive binary relation
 \Rightarrow note*:

[A transitive non-slot binary relation is a non-slot binary relation. For which, there is a role relation, the first domain of which is this binary relation such that for any two tuples of this binary relation, if any element's membership to one of them does not belong to this role relation and the

reflexive non-slot binary relation

\subset reflexive binary relation
 \Rightarrow note*:

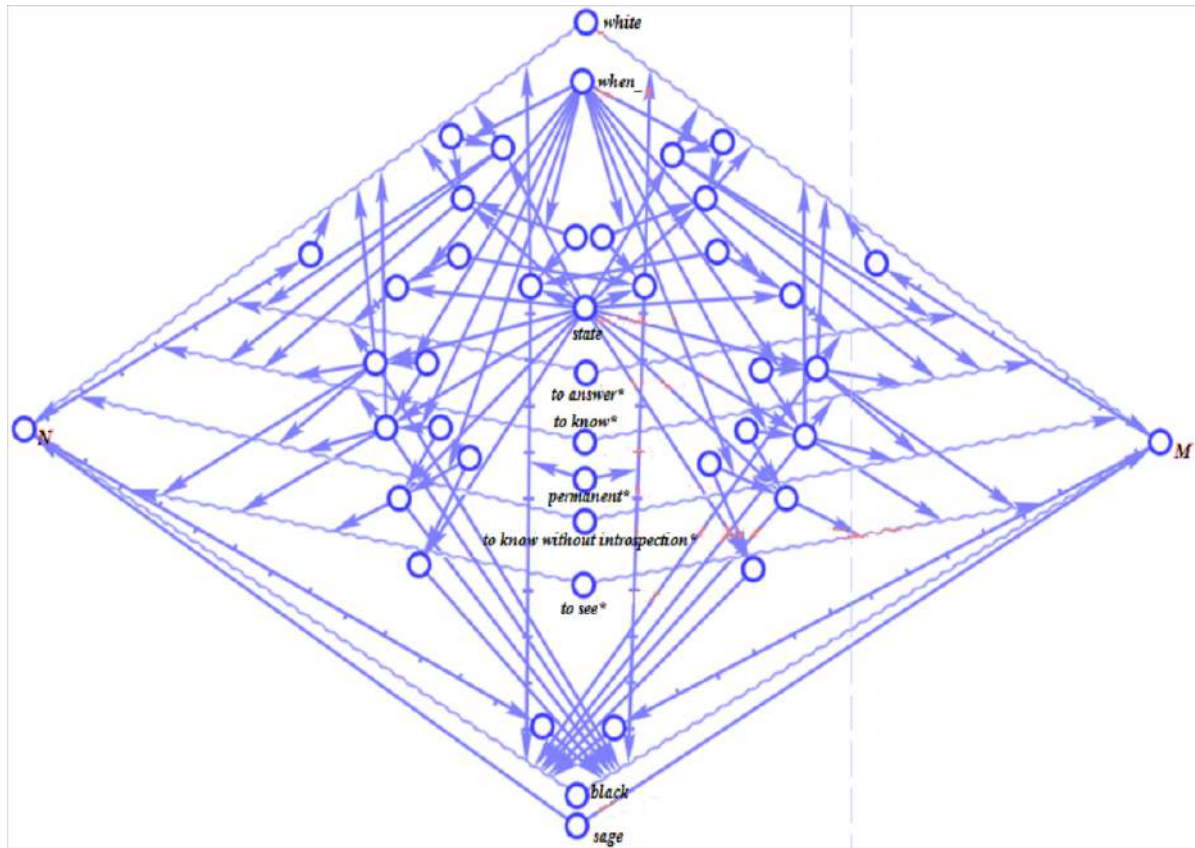


Figure 27. Example of solving the puzzle with sages hats for both of two sages.

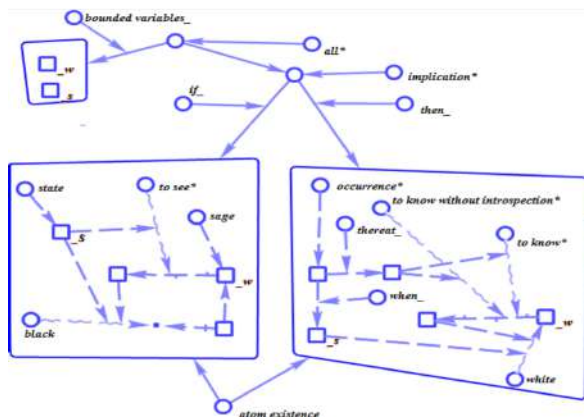


Figure 28. Rule for solving the sages hat puzzle.

membership of the same element to the second tuple belongs to this role relation then there is a tuple with element's membership belonging to the role relation, whose membership to the first tuple belongs to the role relation, and with element's membership that does not belong to the role relation, whose membership to the second tuple does not belong to this role relation.]

symmetric slot binary relation

\subset symmetric binary relation
 \Rightarrow note*:

[A symmetric slot binary relation is a slot binary relation. For any tuple of which, there is a tuple the end of the last is the beginning of the first tuple and the beginning of the last is end of the first (i.e. these tuples are denoted by opposite arcs).]

symmetric non-slot binary relation

\subset symmetric binary relation
 \Rightarrow note*:

[A symmetric non-slot binary relation is a non-slot binary relation. For which, there is a role relation the first domain of which is this binary relation. Also, for any tuple of which, there is a tuple with an element whose membership in the first tuple belongs to the role relation while the membership of that element in the last tuple does not belong to this role relation and with an element whose first tuple membership does not belong to the role relation while its membership in the last tuple belongs to this role relation.]

antisymmetric slot binary relation \subset *antisymmetric binary relation* \Rightarrow *note**:

[An antisymmetric slot binary relation is a slot binary relation. For any tuple of which having a different beginning and end, there is no tuple the end of which is the beginning of the first tuple and the beginning of the last is the end of the first one (i.e. these tuples are denoted by opposite arcs) .]

antisymmetric non-slot binary relation \subset *antisymmetric binary relation* \Rightarrow *note**:

[An antisymmetric non-slot binary relation is a non-slot binary relation. For which, there is a role relation the first domain of which is this binary relation. For any tuple of which, there is no tuple with an element whose membership in the first link belongs to the role relation while its membership to the second link does not belong to this role relation and with an another element whose first tuple membership does not belong to the role relation but its membership to the second tuple belongs to this role relation.]

monotonic slot binary relation* \subset *monotonic binary relation** \Rightarrow *note**:

[Monotone slot binary relation * is slot binary relation with respect to an order relation. Thus, if there is a tuple of this binary relation then for any of its two tuples, the beginning of the second is connected by the tuple of this order relation with the beginning of the first one, there is a third tuple of the binary relation the beginning of which coincides with the beginning of the second tuple and the end coincides with the end of the first.]

monotonic relation order relation' \Rightarrow *first domain**:*monotonic binary relation** \Rightarrow *second domain**:*order relation***monotonic binary relation'** \Rightarrow *first domain**:*monotonic binary relation** \Rightarrow *second domain**:*monotonic binary relation***monotonic non-slot binary relation*** \subset *monotonic binary relation** \Rightarrow *note**:

[Monotonic non-slot binary relation* is non-slot binary relation with respect to an order relation. For which, there is a role relation, the first domain of which is this binary relation, such that if there is a tuple of this binary relation, then for any of its two tuple, if the element belongs to the last of them under this role relation unlike the other one and also it is connected by the tuple of this order relation with the element belonging to the first tuple under the same role relation in contrast to another element of the first tuple then there is a third tuple of this binary relation such that the element belonging to it under the role relation belongs to the second tuple under the same role relation and the element belonging not under this role relation to the third tuple belongs without this role relationship to the first tuple.]

slot equivalence relation \subset *equivalence sc-relation* \Rightarrow *note**:

[A slot equivalence relation is a slot transitive binary relation which is a slot reflexive and symmetric relation.]

non-slot equivalence relation \subset *equivalence sc-relation* \Rightarrow *note**:

[A non-slot equivalence relation is a non-slot transitive binary relation that is a non-slot reflexive and symmetric relation (over the respective domains).]

slot non-strict order relation \subset *sc-relation of non-strict order* \Rightarrow *note**:

[A slot relation of nonstrict order is a transitive binary relation that is reflexive and antisymmetric.]

non-slot non-strict order relation \subset *sc-relation of non-strict order* \Rightarrow *note**:

[A non-slot non-strict order relation is a transitive binary relation that is reflexive and antisymmetric (with respect to the respective domains).]

Inference relation \supset *Inference relation on finite sets* \supset *Inference relation on finite sets of full-connectively represented sets* \in *reflexive binary relation* \in *transitive binary relation* \in *monotonic binary relation* \Rightarrow *note**:

[The inference relation is a reflexive, transitive, monotone binary relation on sets of premises (judgments (propositions), logical formulas). The properties of the inference relation are the rules of inference by Gentzen.]

sequent

⇒ *note**:
 [A sequent is a tuple (of an implicative form) between a conjunctive set of logical formulas (conjunction) and a disjunctive set of logical formulas (disjunction). An example of a sequent is an expression (judgement) like: $A_1 \wedge A_2 \wedge \dots \wedge A_n \Rightarrow C_1 \vee C_2 \vee \dots \vee C_m$.]

antecedent'

⇒ *first domain**:
sequent
 ⇒ *second domain**:
conjunction

consequent'

⇒ *first domain**:
sequent
 ⇒ *second domain**:
disjunction

Inference relation on sequents

⇒ *note**:
 [The inference relation on sequents satisfies the rules of inference of the sequent calculus.]

metastructure

⇒ *note**:
 [A metastructure is a structure whose full-connectively represented element is another structure.]

modal operator

⇒ *note**:
 [A modal operator is a logical connectives that links a logical formula to a structure (and sometimes other elements) in a metastructure. An example of a modal operator is the knowledge operator: Δ .]

modal inference rule

⇒ *note**:
 [A modal inference rule the modal operator binding of a formula is true (has a true interpretation) in a structure if and only if the formula is true (has a true interpretation) in the structure that precedes it. An example of an inference rule is the knowledge operator: $\Gamma \cup \{\alpha\} \vdash \Gamma \cup \{\Delta\alpha\}$.]

relation of becoming of structures

⇒ *note**:
 [The relation of becoming of structures is a binary relation on the set of structures having a non-empty common support. The roles in the tuples of the relation of becoming are the role relations of the previous structure (preceding structure) and the subsequent structure.]

thinking sequence

:= [fate of thinking]
 := [thought]
 ⇒ *note**:
 [A sequence of thinking is a sequence of sc-sets of propositions (logical formulas).]
 ⇒ *subdividing**:
 {
 • *irrational thinking sequence*
 • *rational thinking sequence*
 }

sequence of rational thinking of classical logic

:= [fate of rational thinking of classical logic]
 \subset *rational thinking sequence*
 ⇒ *note**:
 [A sequence of rational thinking is a sequence (given by the relation of becoming of structures) of (classically) satisfiable sc-sets (sc-subsets or sc-supersets) of propositions.]

sequence of classical rational deductive thinking

\subset *sequence of rational thinking of classical logic*
 \supset *sequence of rational deductive thinking of classical logic on finite sc-sets*
 ⇒ *note**:
 [A sequence of classical rational deductive thinking is a sequence of rational thinking, the sequence (of becoming) of satisfiable sc-sets of propositions deductively logically following (according to classical rules) one after another.]

sequence of classical rational deductive cognition

:= [will]
 ⇒ *note**:
 [A sequence of classical rational deductive cognition is a sequence (given by the relation of becoming of structures) of non-contradictory sc-supersets of propositions logically following from one to another.]

Non-classical logics [27], [29]–[31], [33], [34] consider (1) non-classical inference whose inference relation having unusual properties [27], [29]–[31], following which, it is possible or impossible to deduce results which is deduced in classical logic, as well as (2) other scales of truth constants for logical formulas, their interpretations, and [33], [34] values that are different from false and

(ground) truth.

The event-like nature of event (distensible) sets [3], [4], used by the languages of the unified knowledge representation model, allows naturally not only representing transitions between deducible sets of facts under the conditions of the open world assumption (hypothesis) but also the non-monotonic modifiable reasoning corresponding to them under the conditions of the closed world assumption (hypothesis). To implement non-monotone inference, special relations of non-monotone inference are introduced. Modifiable reasoning considers four kinds of predicates:

$$\begin{aligned} P1_i &= P1_j \\ P2_i &\subseteq P2_j \\ P3_i &\supseteq P3_j \\ P4_i &? P4_j \end{aligned}$$

An efficient solution of the problem is possible by adding additional restrictions on the class of considered formulas and functions (Horn clauses, monotone predicates). The transition to fuzzy logic allows reducing a discrete problem to a continuous problem (embed it in a continuous model which is close to quantor elimination methods). The used formalism of event (distensible) sets [3], [4] allows representing fuzzy logic expressions in a natural way. Let us consider some definitions of fuzzy connectives that link and express causality and exclusivity in Lukasiewicz's fuzzy logic [34].

$$\begin{aligned} \varphi \tilde{\rightarrow} \psi & \\ \tilde{1} &\stackrel{\text{def}}{=} \tilde{0} \tilde{\rightarrow} \tilde{0} \\ \sim \varphi &\stackrel{\text{def}}{=} \varphi \tilde{\rightarrow} \tilde{0} \\ \varphi \tilde{\wedge} \psi &\stackrel{\text{def}}{=} (\varphi \tilde{\rightarrow} \psi) \tilde{\rightarrow} \psi \\ \varphi \tilde{\vee} \psi &\stackrel{\text{def}}{=} \sim (\sim \varphi \tilde{\wedge} \sim \psi) \\ \varphi \tilde{\oplus} \psi &\stackrel{\text{def}}{=} (\sim \varphi) \tilde{\rightarrow} \psi \\ \varphi \tilde{\otimes} \psi &\stackrel{\text{def}}{=} \sim (\sim \varphi \tilde{\oplus} \sim \psi) \\ \varphi \tilde{\ominus} \psi &\stackrel{\text{def}}{=} (\varphi \tilde{\rightarrow} \psi) \\ \varphi \tilde{\leftrightarrow} \psi &\stackrel{\text{def}}{=} (\varphi \tilde{\rightarrow} \psi) \vee (\psi \tilde{\rightarrow} \varphi) \end{aligned}$$

To implement the inference in other non-classical logics, including those dealing with subject domains in which non-deterministic sets and structures are considered, additional concepts are also considered.

nonmonotonic inference on finite sc-set of premises

⇒ *note**:

[Nonmonotonic inference on a finite sc-set of premises is a relation between (finite) sc-sets of true logical statements (premises). If there is no embedding of the structure of an atomic logical formula in the relational structure (sc-subset of the subject area) of the sc-set of true (consistent) premises and the negation of this atomic formula is true with respect to them,

then there is an sc-set with a relational structure belonging to it, including all elements of the previously mentioned relational structure and the constants of this atomic formula, to which all premises of the previously mentioned sc-set of true (consistent) premises and the mentioned atomic logical formula belong.]

inferencing set

⇒ *note**:

[An inferencing set is an event sc-set, the (temporary) belonging of logical formulas to which is established in the order of formation of the process of deriving these logical formulas.]

fuzzy truth*

⇒ *note**:

[Fuzzy truth connects a finite sc-set with temporal belongings on a finite set of finite phenomena of belonging to a statement. On the membership phenomena, a finite sc-subset of the sc-relation of becoming (immediately before, immediately after) is given, which defines the structure of the corresponding sc-subsets. This structure is a directed tree. Fuzzy truth is a binary relation between the (fuzzy) membership of a link of a formal theory statement and a finite sc-set and a real number from 0.0 to 1.0. The fuzzy truth of the negation of the statement is equal to the difference 1.0 and the fuzzy truth of the statement belonging to the negation. The fuzzy truth of the conjunction of propositions does not exceed the minimum of the fuzzy truth of the elements of this conjunction and is not lower than the (boundary or drastic) product of the fuzzy truth of the same elements of the conjunction. The fuzzy truth of a disjunction does not exceed the (boundary or drastically) sum of the fuzzy truth of the elements of this conjunction and not less than the maximum fuzzy truth of the same elements of the conjunction. The fuzzy truth of atomic propositions is equal to the arithmetic mean of the isomorphic embedding of the proposition structure in each of the sc-subsets of the finite sc-set that are included in the structure given by the becoming sc-relation.]

constructively true proposition*

⇒ *note**:

[A constructively true proposition* is a subset of a true proposition*. True atomic logical formulas or their true interpretations are constructively true if and only if they have an isomorphic embedding in the domain where all elements of the embedding

are full-connectively represented. A conjunction of constructively true logical formulas (or having corresponding constructively true interpretations) is constructively true (or has a constructively true corresponding interpretation). The disjunction of at least one constructively true logical formula (or having a corresponding complete constructively true interpretation) is constructively true (or has a constructively true corresponding interpretation). The negation of a false logical formula (or having a false corresponding interpretation) is constructively true (or has a constructively true interpretation). If all the logical formulas in the disjunction are false (have corresponding false interpretations), then the disjunction is also false (has a corresponding false interpretation). The negation of a false logical formula (or having a false corresponding interpretation) is constructively true (or has a constructively true interpretation). An implication with a false premise (or having a corresponding false interpretation) is constructively true (or has a constructively true interpretation). An implication with a constructively true consequence (or having a corresponding constructively true interpretation) is constructively true (or has a constructively true interpretation). A constructively true implication (or having a constructively true interpretation) with a constructively true premise (or having a corresponding constructively true interpretation) has a constructively true consequence (or having a corresponding constructively true interpretation). A constructively true implication (or having a constructively true interpretation) with a false consequence (or having a corresponding false interpretation) has a false premise (or having a corresponding false interpretation). The existence of variable values for a logical formula is constructively true (or has a corresponding constructive true interpretation) if the universality of variable values for that logical formula is constructively true (or has a corresponding constructive true interpretation). If a logical formula has only constructively true corresponding interpretations, then the universality of the values of the variable for this logical formula is constructively true (or has a corresponding constructively true interpretation).]

right proposition*

⇒ note*:

[A right proposition is a proposition that is true or uncorrupted.]

uncorrupted proposition*

⇒ note*:

[An uncorrupted proposition is a proposition whose truth or falsity (untruth) does not lead to a contradiction.]

The approach based on the semantic space [4], [8], in addition to considering useful metric properties, allows imposing additional requirements on the topological properties of the corresponding logics, when all substructures of the relational structure of the subject domain, on which the logical formulas are interpreted, are meaningfully closed.

IV. CONCLUSION

In reference and testing (checking) systems, there is a problem of analyzing complex answers characteristic of modified reasoning. The representation of such answers as logical constructions in the semantic space makes it possible to quantify the correctness of such answers, as well as the quality of the knowledge that the system is able to acquire in the process of dialog within the OSTIS Ecosystem.

REFERENCES

- [1] V. Golenkov, N. Gulyakina, and D. Shunkevich, *Standart otkrytoi tekhnologii ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheskii sovmestimyykh gibridnykh intellektual'nykh komp'yuternyykh sistem [Open technology standard for ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint, 2021, (In Russ.).
- [2] V. V. Golenkov and N. A. Gulyakina, "Next-generation intelligent computer systems and technology of complex support of their life cycle," *Otkrytye semanticheskii tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 27–40, 2022.
- [3] V. P. Ivashenko, "Modeli i algoritmy integratsii znaniia na osnove odnorodnykh semanticheskikh setei (diss. na soiskanie stepeni kand. tekhn. nauk: 05.13.17) [models and algorithms for knowledge integration based on homogeneous semantic networks (diss. for the degree of candidate of technical sciences: 05.13.17)]," dis. ... kand. tekhn. nauk : 05.13.17, Belarus. gos. un-t informatiki i radioelektroniki [Belarusian State University for Informatics and Radioelectronics], Minsk, 2014, (In Russ.).
- [4] V. Ivashenko, "General-purpose semantic representation language and semantic space," *Otkrytye semanticheskii tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 41–64, 2022.
- [5] —, "String processing model for knowledge-driven systems," *Doklady BGUIR*, vol. 18, pp. 33–40, 10 2020.
- [6] V. Golenkov, V. Golovko, N. Guliakina, and V. Krasnoprosin, "The standardization of intelligent computer systems as a key challenge of the current stage of development of artificial intelligence technologies," pp. 73–88, 2020.
- [7] W. Lowe, "Towards a theory of semantic space," 01 2001, pp. 576–581.
- [8] Y. I. Manin and M. Marcolli, "Semantic spaces," *Mathematics in Computer Science*, vol. 10, pp. 459–477, 2016.
- [9] I. Mel'cuk, *Language: from Meaning to Text*, D. Beck, Ed. Moscow & Boston: Academic Studies Press, 04 2016.
- [10] Y. M. A. Kostrikin, *Linear Algebra and Geometry*. Gordon and Breach Science Publishers, 1997.
- [11] D. Bohm and B. Hiley, *The Undivided Universe: An Ontological Interpretation of Quantum Theory*. London: Routledge, 1993.

- [12] D. Bohm, *Wholeness and the Implicate Order*. London: Routledge, 2002.
- [13] A. N. Gordei. Theory for automatic generation of knowledge architecture: Tapaz-2. Available at: <http://tapaz.by> (accessed 18.04.2021).
- [14] A. Hardzei, M. Sviatoschchik, L. Bobyor, and S. Nikiforov, "Universal semantic markup and top-level ontology representation," in *Open Semantic Technologies for Intelligent Systems*. Springer International Publishing, 2022, pp. 192–221. [Online]. Available: https://doi.org/10.1007/978-3-031-15882-7_11
- [15] M. A. Armstrong, *Basic Topology*. M.: Springer, 1983.
- [16] L. Collatz, *Functional Analysis and Numerical Mathematics*. New York, San Francisco, London: Academic Press, 1966.
- [17] A. Narin'yani, "Ne-factory: kratkoe vvedenie [non-factors: a brief introduction]," *Novosti iskusstvennogo intellekta [AI news]*, no. 2, pp. 52–63, 2004.
- [18] V. P. Ivashenko, "Ontologicheskaya model' prostranstvenno-vremennykh otoshenii sobytii i yavlenii v protsessakh obrabotki znaniy [ontological model of the space-time relations of the event and the phenomenon in the processes of knowledge processing]," *Vestnik Brestskogo gosudarstvennogo tekhnicheskogo universiteta [Bulletin of Brest State Technical University]*, vol. 107, no. 5, pp. 13–17, 2017, (In Russ.).
- [19] V. M. Kuz'mitskii, "Printsipy postroeniya grafodinamicheskogo parallel'nogo komp'yutera, orientirovannogo na reshenie zadach iskusstvennogo intellekta [principles of constructing a graph-dynamic parallel computer focused on solving artificial intelligence problems]," *dis. ... kand. tekhn. nauk : 05.13.11 [dis. ... Cand. of tech. sciences: 05.13.11]*, Minsk, 2000, 236 p.
- [20] L. Kalinichenko, *Metody i sredstva integratsii neodnorodnykh baz dannykh [Methods and tools for integrating heterogeneous databases]*. M.: Nauka, 1983, (In Russ.).
- [21] V. Ivashenko, "Application of an integration platform for ontological model-based problem solving using an unified semantic knowledge representation," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 179–186, 2022.
- [22] V. P. Ivashenko, *Modeli resheniya zadach v intellektual'nykh sistemakh. V 2 ch. Ch. 1 : Formal'nye modeli obrabotki informatsii i parallel'nye modeli resheniya zadach : ucheb.-metod. posobie [Models for solving problems in intelligent systems. In 2 parts, Part 1: Formal models of information processing and parallel models for solving problems: a tutorial]*. Minsk : BSUIR, 2020, (In Russ.).
- [23] A. Tarski, "On the calculus of relations," *J. Symbolic Logic*, vol. 6, 09 2014.
- [24] W. Li and L. Qian, "Semantic approach to user answer verification in intelligent tutoring systems," in *Open Semantic Technologies for Intelligent Systems*, V. Golenkov, V. Krasnoproshin, V. Golovko, and D. Shunkevich, Eds. Springer International Publishing, 2022, pp. 242–266. [Online]. Available: https://doi.org/10.1007/978-3-031-15882-7_13
- [25] T. Imielinski and W. Lipski, "The relational model of data and cylindric algebras," *J. Comput. Syst. Sci.*, vol. 28, pp. 80–102, 1984.
- [26] Z. Lintao, C. Madigan, M. Moskewicz, and S. Malik, "Efficient conflict driven learning in a boolean satisfiability solver," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers*, 02 2001, pp. 279–285.
- [27] A. Teiz, P. Gribomon, G. Yulen, A. Pirot, D. Rolan, D. Snaiers, M. Vokler, P. Goshe, P. Vol'per, E. Greguar, and F. Del'sart, *Logicheskii podkhod k iskusstvennomu intellektu: ot modal'noi logiki k logike baz dannykh [Logic approach to Artificial Intelligence: from modal logic to database logic]*, G. Gavrilov, Ed. M.: Mir, 1998, (In Russ.).
- [28] S. K. Klini, *Matematicheskaya logika [Mathematical logic]*. Moskva : Mir, 1973, (In Russ.).
- [29] A. Teiz, P. Gribomon, and Z. L. i dr., *Logicheskii podkhod k iskusstvennomu intellektu: ot klassicheskoi logiki k logicheskoi programirovaniyu [Logic Approach to Artificial Intelligence: From Classical Logic to Logic Programming]*, G. Gavrilov, Ed. M.: Mir, 1990, (In Russ.).
- [30] A.G.Dragalin, *Konstruktivnaya teoriya dokazatel'stv i nestandardnyy analiz [Constructive evidence theory and non-standard analysis]*, G. E. Mints, M. K. Valiev, E. G. Dragalina-Chernaya, N. M. Nagorny, N. N. Nepeivoda, V. N. Sadvoskii, and E. D. Smirnova, Eds. Editorial URSS, 2003, (In Russ.).
- [31] V. N. Vagin, E. Y. Golovina, A. A. Zagoryanskaya, M. V. Fomina, and D. A. Pospelov, *Dostovernyye i pravdopodobnyy vyvod v intellektual'nykh sistemakh [Reliable and plausible inference in intelligent systems]*, 2nd ed., pod red. V. N. Vagina and D. A. Pospelova, Eds. M. : Fizmatlit, 2008, (In Russ.).
- [32] V. Golenkov, V. Tarasov, O. Eliseeva, N. Gulyakina, T. Lemesheva, N. Bezzubenok, V. Ivashenko, T. Gavrilova, E. Vasil'eva, V. Emel'yanov, R. Serdyukov, S. Akulich, O. Lubnevskii, and Y. Tel'nov, *Intellektual'nye obuchayushchie sistemy i virtual'nye uchebnye organizatsii: Monografiya [Intelligent Learning Systems and Virtual Learning Organizations: Monograph]*, V. Golenkov and V. Tarasov, Eds. Mn.: BGUIR, 2001, (In Russ.).
- [33] I. Z. Batyrshin, A. O. Nedosekin, A. A. Stetsko, V. B. Tarasov, A. V. Yazenin, and N. G. Yarushkina, *Nechetkie gibridnye sistemy. Teoriya i praktika [Fuzzy hybrid systems. Theory and practice]*, pod red. N. G. Yarushinoi, Ed. M. : Fizmatlit, 2007, (In Russ.).
- [34] L. Behounek, P. Cintula, and P. Hájek, *Introduction to mathematical fuzzy logic*, 01 2011, vol. 37, pp. 1–101.

Интеграция логических моделей представления и обработки знаний в смысловом пространстве

Ивашенко В. П.

В данной статье рассматривается подход к представлению логических структур и схем, как моделей логической обработки знаний, в семантическом пространстве в виде семантических сетей. Исследуются некоторые свойства семантического пространства и логических моделей, такие как топологические, метрические и валюационные (нормовые) свойства. Предложены понятия для онтологического представления классических и неклассических логических формул, классов и отношений.

Received 13.03.2023

Principles of Problem Solving in Distributed Teams of Intelligent Computer Systems of a New Generation

Daniil Shunkevich

*Belarusian State University of
Informatics and Radioelectronics*

Minsk, Belarus

Email: shunkevich@bsuir.by

Abstract—The article considers an approach to the organization of problem solving within the distributed collective of intelligent computer systems that are part of the OSTIS Ecosystem (ostis-systems). The classification of agents within such a system and the principles of their interaction are considered.

Keywords—OSTIS Technology, OSTIS Ecosystem, problem solver, multi-agent approach

I. INTRODUCTION

One of the key components of the *intelligent system*, which provides the ability to solve a wide range of *tasks*, is *problem solver*. Their peculiarity in comparison with other modern *software systems* is the necessity to solve *tasks* in conditions when the needed information is not explicitly localized in *knowledge base of intelligent system* and must be found in the process of solving *tasks* based on any criteria.

The development of intelligent systems problem solvers at this moment is usually considered in the context of individual (independent) intelligent systems operating in some environment (the user is part of this environment, if there is one). At the same time, there is an obvious tendency of modern information technologies to move from individual systems to collectives of distributed interacting computer systems, in particular, to distributed data storage and distributed computing. In the case of intelligent computer systems, the most important property of the systems involved in such collectives becomes *interoperability*, which is the ability of the system to interact in a coordinated manner with other similar systems in order to solve any problems. Therefore, the transition from the development of problem solvers of individual intelligent systems to problem solvers of interacting interoperable intelligent systems is especially relevant, including the development of principles for solving problems in such distributed teams, in light of the solution of all the problems outlined above.

The expansion of the application areas of *intelligent systems* requires them to be able to solve the so-called

complex problems, the solution of each of which requires combining several models of problem solving, while it is not known a priori in what order and how many times a particular model will be used. *problem solvers*, in which several *problem solving models* are combined, are called *hybrid problem solvers*, and intelligent systems, in which various *types of knowledge* and various *problem solving models* are combined – *hybrid intelligent systems* [1].

Improving the efficiency of the development and use of *hybrid intelligent systems* requires the unification of models for the representation of various *types of knowledge* and *models of knowledge processing*, which would make it easy to integrate components based on it correspond to different models of problem solving. Such models based on the unified semantic representation of information are proposed within the framework of the OSTIS Technology [2]–[4]. The systems developed using this technology are called *ostis-systems*. The encoding of information in the memory of *ostis-systems* (*sc-memory*) is based on the language of unified semantic networks, called *SC-code*. The elements of such a semantic network (*sc-text*) are called *sc-elements* (*sc-nodes*, *sc-arcs*, *sc-edges*).

Within the frame of this article, it is proposed to clarify the principles of solving problems by a distributed team of interacting intelligent computer systems on the basis of the previously proposed principles of solving problems within the framework of individual intelligent computer systems (see [3]). To solve this problem, it is proposed to consider such a system of interacting intelligent computer systems as a multi-agent system and clarify the principle of agent behavior in such a system.

II. PRINCIPLES OF DEVELOPING PROBLEM SOLVERS FOR INDIVIDUAL COMPUTER SYSTEMS

The proposed approach to problem solving is based on a number of ideas related to the concept of situational management proposed in the work of D. Pospelov [5] and developed in works in the field of multi-agent knowledge processing [6].

Let's consider the principles underlying the proposed approach to the development of *hybrid problem solvers* [3], [4]:

- as a foundation for creating a hybrid *problem solver* model, it is proposed to use *multi-agent approach*. This approach allows to provide a foundation for the construction of parallel asynchronous systems with a distributed architecture, to increase the modifiability and performance of the developed *problem solvers*;
- *problem solver* is proposed to be considered as a hierarchical system consisting of several interconnected levels;
- it is proposed to record all information about the solver and the problems solved by them using *SC-code* in the same *knowledge base* as the actual subject of systems *knowledge*. This will allow, firstly, to ensure the independence of the developed *problem solvers* from the platform for interpreting semantic models of ostis-systems (*ostis-platforms*, see [4]), secondly, to enable the system to analyze the processes occurring in it, optimize and synchronize their execution, that is, to ensure *reflexivity* of the designed *intelligent systems*.

The focus on *multi-agent approach* as a foundation for building *hybrid problem solvers* is owing a number of advantages of such an approach [3], [4], [7]–[9].

In accordance with these principles, the *ostis-system problem solver* is proposed to be divided into components corresponding to classes of logically atomic actions in semantic memory, called *sc-agents*.

Logical atomicity of performed *sc-agent actions* assumes that each *sc-agent* reacts to its corresponding class of *situations* and/or *events* occurring in *sc-memory* and performs a certain transformation of *sc-text* located in the semantic neighborhood of the processed *situations* and/or *events*. At the same time, each *sc-agent* generally does not have information about which other *sc-agents* are currently present in the system and interacts with other *sc-agents* exclusively through the formation of some constructs (usual— action specifications) in the common *sc-memory*. Such a message can be, for example, a question addressed to other *sc-agents* in the system (it is not known in advance which one specifically), or an answer to a question posed by other *sc-agents* (it is not known in advance which one specifically). Thus, each *sc-agent* at any given time controls only a fragment of *knowledge base* in the context of the task being solved by this agent *task*, the state of the rest of the *knowledge base* is generally unpredictable for *sc-agent*.

A certain *method* can be assigned to an action class, that is, a description of how any or almost any (with explicit exceptions) action belonging to this *action class* can be performed. Since a specific *class of actions* corresponds to some specific *class of tasks*, we can say that the method describes a way to solve any tasks belonging to a given

class. The concept of a method can be considered a generalization of the concept of "program", and therefore within the framework of *OSTIS Technology* the terms "method" and "program" are synonymous, and the term "method representation language" is synonymous with the term "programming language".

Since it is assumed that copies of the same *sc-agent* or functionally equivalent *sc-agents* can work on different *ostis-systems*, while being physically different *sc-agents*, then it is expedient to consider the properties and classification not of *sc-agents*, but of classes of functionally equivalent *sc-agents*, which we will call **abstract sc-agents**. **abstract sc-agent** is understood as a certain class of functionally equivalent *sc-agents*, different instances (that is, representatives) of which can be implemented in different ways.

III. HIERARCHY OF SC-AGENTS FROM THE POINT OF VIEW OF THE METHOD REPRESENTATION LANGUAGE LEVEL

Sc-agents can be classified according to various criteria. Since we can talk about the hierarchy of *methods* (methods of interpretation of other methods) and, accordingly, the hierarchy of skills, then there is a need to talk about the hierarchy of *sc-agents* that provide interpretation of a particular method. In this context, we can talk about the hierarchy of *sc-agents* in two aspects:

- *abstract sc-agent* (and, accordingly, *sc-agent*) can uniquely correspond to *method (sc-agent program)* describing the activity of this *sc-agent*. Such agents will be called *atomic abstract sc-agents*;
- *abstract sc-agents* sometimes it is advisable to combine such agents into collectives, which can be considered as one integral *abstract sc-agent*, from a logical point of view, working on the same principles as *atomic abstract sc-agents*, that is, reacting to events in *sc-memory* and describing its activities within this memory. Such a *abstract sc-agent* will not correspond to any specific *method* stored in *sc-memory*, but the rest of the specification of *abstract sc-agent* (initiation condition, description of the initial situation and the result of *sc-agent* and so on) remains the same as that of *atomic abstract sc-agent*. Therefore, we can say that the concept of atomicity/non-atomicity of *abstract sc-agent* indicates how the implementation of this *abstract sc-agent* is specified – by specifying a specific method (*program sc-agent*) or by decomposition of *abstract sc-agent* to simpler ones. It is important to note that *non-atomic abstract sc-agents* can also be part of other, more complex *non-atomic abstract sc-agents*. Thus, a hierarchical system of *abstract sc-agents* is formed, generally having a random number of levels.
- In turn, the corresponding *sc-agent* method should be interpreted by some other *sc-agent* of a lower

level, and most often by a team of such agents, each of which is assigned its own *method* describing the behavior of this agent, but already at a lower level. Therefore, we can say that the concept of atomicity/non-atomicity of *abstract sc-agents* is applicable within the framework of one *method description language*. In turn, we can talk about the hierarchy of *abstract sc-agents* from the point of view of the level of the language of description of the methods corresponding to such agents. In general, such a hierarchy can also have an unlimited number of levels, however, it is obvious that when lowering the level of the method description language, sooner or later we must approach the method description language, which will be interpreted by agents implemented at the level of *ostis-platform*, and descending even lower - to the level of the method description language, interpreted at the hardware level. Thus, in order to ensure the platform independence of *ostis-systems*, it is advisable to allocate a method description language that would be interpreted at the level of *ostis-platform* and be the basis for the development of interpreters of higher-level languages. *Language SCP* is suggested as such a language (Semantic Code Programming), which is considered as an assembler for *associative semantic computer* [4].

With that stated, we will distinguish two variants of the classification of *abstract sc-agents*. Classification of *abstract sc-agents* on the basis of atomicity:

abstract sc-agent

⇒ *subdividing**:

- { • *non-atomic abstract sc-agent*
- *atomic abstract sc-agent*
- }

Classification of *abstract sc-agents* based on the possibility of their implementation at the platform-independent level:

abstract sc-agent

⇒ *subdividing**:

- { • *abstract sc-agent, not implemented in the SCP language*
- *abstract sc-agent, implemented in the SCP language*
- }

The classification of *abstract sc-agents* is discussed in more detail in [3].

IV. PRINCIPLES OF ORGANIZATION OF PROBLEM SOLVING IN A DISTRIBUTED TEAM OF OSTIS-SYSTEMS

Above, the principles of organizing the process of solving a problem by a team of agents within an individual

ostis-system were considered. Given the necessity of solving problems in a distributed team of ostis-systems, it is advisable to talk about two types of multi-agent systems within the framework of OSTIS Technology:

- internal system of sc-agents over common sc-memory within some ostis-system;
- a distributed system of ostis-systems within the OSTIS Ecosystem [4].

In both cases, we can talk about hierarchy of agents:

- within the internal system of sc-agents, *atomic abstract sc-agents* and *non-atomic abstract sc-agents* are distinguished, in addition, there is a hierarchy of sc-agents from the point of view of the method interpretation language;
- Within the OSTIS Ecosystem, both *individual ostis-systems* and *collective ostis-systems* are distinguished, which in turn can consist of both *individual ostis-systems* and *collective ostis-systems*.

The key difference between the distributed system of ostis-systems and the internal system of -agents within the framework of *individual ostis-system* is the absence of a common memory that stores a common knowledge base for all sc-agents and acts as a medium for communication of sc-agents. In general, as a means of communication between agents within the framework of dedicated agent systems, it can be used:

- Shared unallocated (monolithic) memory, as in the case of sc-agents over sc-memory;
- Shared distributed memory. In this case, from a logical point of view, agents can assume that they are still working on a shared memory, within which the entire available knowledge base is stored, but in reality the knowledge base will be distributed among several ostis-systems and the transformations performed will have to be synchronized between these ostis-systems;
- Specialized communication channels. Obviously, when solving a problem in a distributed team of ostis-systems, there must be language and technical means that allow for the transmission of messages from one ostis-system to another.

All the listed methods of communication, depending on the class of the problem being solved, the *knowledge* and *skills* required for its solution, as well as the currently existing (available) set of ostis-systems, can be combined.

The idea of the maximum possible unification and convergence of the principles of problem solving within the framework of an individual ostis-system and a distributed team of ostis-systems is proposed as the basis for solving problems within a distributed team of ostis-systems. This approach has the following important advantage: if the general principles of problem solving do not depend on which specific set of ostis-systems is involved in solving a particular problem, then it becomes possible to easily switch from *individual ostis-system* to

a distributed team of ostis-systems with its complication without the need to significantly revise the team of agents, which are part of such an ostis-system and rethink the approach used to solve problems of a particular class. To switch from *individual ostis-system* to *collective ostis-system*, it is enough to do the following steps:

- Divide the set of classes of problems solved by this ostis-system into a family of subsets, each of which has some logical integrity, the criteria of which are generally determined by the developer. At the same time, these subsets may intersect, but when combined they must give the original set, so it is necessary to construct one of the possible *coverings** for the set of classes of problems solved by this ostis-system;
- For each of the selected subsets, it is necessary to form a set of *knowledge* and *skills* necessary to solve the problems of this set of classes. At the same time, in the general case, it may be necessary to revise the hierarchy of skills and their corresponding sc-agents, in particular, the transformation of some atomic sc-agents into non-atomic ones. Theoretically, it is impossible to avoid such a situation, but such situations can be practically eliminated at the stage of designing problem solvers of individual ostis-systems, making the hierarchy of agents sufficiently deep and matching atomic sc-agents with such classes of tasks, the division of which into subclasses from a practical point of view does not make sense. A similar situation may arise during the allocation of fragments of the knowledge base. In this case, it may be necessary to revise the hierarchy of *subject areas* and *ontologies* and, possibly, the allocation of new subject areas. As in the case of problem solvers, it is possible to avoid such a situation in practice if the hierarchy of subject areas is deep enough so that the allocation of more specific subject areas is practically impractical;
- Each set of knowledge and skills formed in this way becomes, respectively, the knowledge base and the problem solver of the new ostis-system, which will be able to implement only part of the functionality of the original ostis-system.

Such separation can be performed iteratively and for the resulting ostis-systems, in general, an unlimited number of times, creating at each iteration a new "generation" of ostis-systems obtained by decomposition of the original ostis-system.

therefore, the proposed idea of unifying the principles of problem solving in ostis-systems of any kind allows:

- from a practical point of view, remove the restriction on the expansion of functionality (training) not only of the *individual ostis-system*, but also of the *collective ostis-system*, therefore allowing us to constantly increase the functionality of the OSTIS Ecosystem as a whole.

- from a theoretical (architectural) point of view, we can talk about the fractal nature of not only the internal organization of ostis-systems, but also the collectives of ostis-systems, which, in turn, makes it possible to inherit other principles of building individual ostis systems in distributed collectives of ostis-systems, including, for example, the design methodology ostis-systems and their components and the corresponding means, as well as the principles of synchronization of parallel information processes corresponding to sc-agents.

The interaction of sc-agents within the framework of an individual ostis-system is based on the refined principle of the "bulletin board" in which agents interact through a common sc-memory for them . To implement the same idea in the case of a distributed collective ostis-system, it is necessary to select some sc-memory to perform this role. When solving problems in a distributed team of ostis-systems, two options for organizing agent interaction are possible (which are the ostis-systems themselves):

- If the task being solved is quite complex and requires frequent access to several separate ostis-systems, then it is advisable to create a *temporary ostis-system* by combining separate ostis-systems, where all sc-agents that were part of the original ostis-systems become internal, and the principles of organizing their interaction are known. In this case, the costs of solving the problem are significantly reduced, but there are overhead costs for creating such *temporary ostis-systems*. Thus, it is necessary to separately develop criteria on the basis of which a decision will be made on the expediency of such an association. Note that in order to be able to save the result and the progress of solving the problem for subsequent use, it is advisable to combine ostis-systems based on one of the ostis-systems included in such an association, and not create a completely new ostis-system. At the same time, knowledge and skills from the combined systems will be copied into such a system, and these combined systems themselves may not change at all. Then, after solving the problem, it will be necessary to exclude from the original ostis-system those skills and knowledge that were needed only to solve this problem.

It is important to note that the described integration of ostis-systems, due to the peculiarities of their architecture, is much easier than in other computer systems, since the principles of building both knowledge bases and problem solvers of ostis-systems initially assume the possibility of unlimited expansion of the knowledge and skills available in the system without the need to make changes to the already existing knowledge base and solver. Thus, the integration of two ostis-systems, subject to their semantic compatibility, is reduced to the usual set-

theoretic unification of their knowledge bases and problem solvers and the subsequent exclusion of duplicated components. Due to this, the creation of such temporary ostis-systems can be performed automatically, which makes the application of this approach to the organization of problem solving expedient in many cases.

- Another possible option assumes that the sc-memory of one of the ostis-systems that are part of the ostis-systems team is selected as the medium for the interaction of sc-agents (both external and internal, the external ostis-system is also considered as a sc-agent from the point of view of the problem solving process). The following criteria for choosing this sc-memory are proposed:
 - If the task is solved repeatedly within the framework of some ostis-community (community of ostis-systems and their users, then, to coordinate the actions of sc-agents, the sc-memory of the corporate ostis-system for this ostis-community is selected;
 - If a team of ostis-systems is formed temporarily (on a one-time basis) to solve this problem, then the sc-memory of the ostis-system that initiated the solution of this problem is selected to coordinate the actions of sc-agents.

The disadvantage of this option is the cost of communication between ostis-systems. If for some reason these costs are high (for example, due to the poor quality of the connection between the systems), then it is more appropriate to use the first of the proposed options.

In any of the proposed options, some specific sc-memory is eventually determined, which becomes an environment for the interaction of agents performing the task solution, according to the principles outlined in the . Then it is possible to clarify the concept of a *sc-agent* as a component of a problem solver in the context of distributed problem solving by a team of ostis-systems and consider as a sc-agent not only a component of the solver of an individual ostis-system, but also any ostis-system that is part of a permanent or temporary team of ostis-systems that solve any problems, since the principles of interaction ostis-systems in such a team completely coincide with the principles of interaction of sc-agents as part of the solver of an *individual ostis-system*.

Thus, we can talk about a fractal hierarchical structure (see [10]) of a distributed hybrid problem solver, within which two variants of the hierarchy of sc-agents are distinguished:

- Hierarchy of sc-agents from the point of view of the level of *method representation languages* in which the methods corresponding to these sc-agents are presented. Within this hierarchy, in turn, three levels can be distinguished that have important differences:

- The ostis-platform sc-agent level, which provides interpretation of platform-independent level methods within the framework of an *individual ostis-system*, within which a hierarchy of presentation languages of ostis-platform level methods and corresponding means of their interpretation can be distinguished;
- The level of platform-independent sc-agents within an *individual ostis-system*, within which a hierarchy of platform-independent method representation languages can be distinguished;
- The level of distributed collectives of ostis-systems, at which it is also possible to talk about the *method representation languages* and their hierarchy, but in general, even individual methods can be physically stored distributed in different ostis-systems. For example, we can talk about the *method representation language* for the financial activities of large enterprises, but it is advisable to allocate sublanguages to describe the activities of departments of various categories and have separate ostis-systems for servicing each of the departments.
- Hierarchy of sc-agents in terms of atomicity/non-atomicity within a single *method representation language*. The formation of such a hierarchy may be appropriate at any level of the language of the *method representation language* and leads to the allocation of:
 - *atomic platform-dependent sc-agents* and *non-atomic platform-dependent sc-agents* at the ostis-platform level;
 - *atomic platform-independent sc-agents* and *non-atomic platform-independent sc-agents* at the platform-independent level within the framework of an individual ostis-system;
 - *individual ostis-systems* and *collective ostis-systems* at the level of problem solving within the OSTIS Ecosystem.

The presented hierarchy of *abstract sc-agents* and *methods* corresponding to *atomic abstract sc-agents* is illustrated in Figure 1. The label “M” in the figure conventionally denotes *methods*, the label “AA” and “NA” – *atomic abstract sc-agents* and *non-atomic abstract sc-agents*, respectively, solid arrows show the decomposition of *non-atomic sc-agents* into simpler ones, and dotted arrows – the relationship between *methods* and their operational semantics, that is, *abstract sc-agents* that provide interpretation of these *methods*. As shown in the figure above, there should be a clear boundary between methods that are described at the *ostis-platform* level and methods that can be described at the platform-independent level. In addition, the upper part of the figure shows ostis-systems that are agents within the OSTIS Ecosystem, and together with the ostis-system considered in more

detail in the lower part of the figure, performing the interpretation of methods within the OSTIS Ecosystem (shown by larger rectangles with the label “M”).

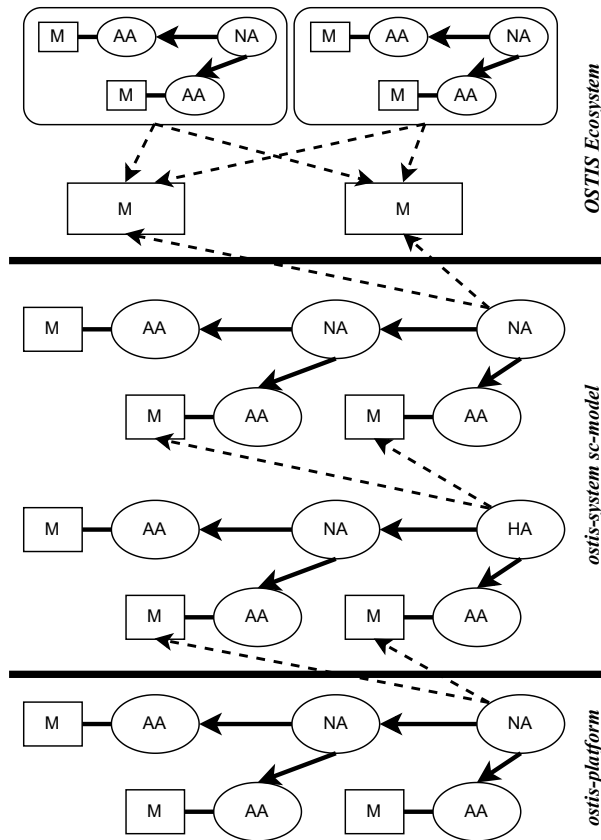


Figure 1. Figure. Hierarchy of sc-agents.

V. CONCLUSION

The paper considers an approach to the organization of problem solving within a distributed team of intelligent computer systems that are part of the OSTIS Ecosystem (ostis-systems).

Further development of the presented principles of problem solving by distributed teams of ostis-systems involves:

- Development of formal criteria for assessing the feasibility or inexpediency of the formation of temporary individual ostis-systems;
- Development of the language and principles of messaging between ostis-systems that are part of the ostis-systems team that solves any task. Despite the fact that from a logical point of view, each ostis-system is treated as a sc-agent and the principles of their interaction remain the same, the implementation, for example, of the ability to respond to events in the knowledge base and make changes to this knowledge base for internal sc-agents and

external ostis-systems will be different and requires clarification.

ACKNOWLEDGMENT

The authors would like to thank the scientific teams of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help in the work and valuable comments.

REFERENCES

- [1] A. Kolesnikov, *Gibridnye intellektual'nye sistemy: Teoriya i tekhnologiya razrabotki [Hybrid intelligent systems: theory and technology of development]*, A. M. Yashin, Ed. SPb.: Izd-vo SpbGTU, 2001.
- [2] I. Davydenko, "Semantic models, method and tools of knowledge bases coordinated development based on reusable components," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2018, pp. 99–118.
- [3] D. Shunkevich, "Hybrid problem solvers of intelligent computer systems of a new generation," *Open semantic technologies for intelligent systems*, no. 6, pp. 119–144, 2022.
- [4] V. Golenkov, N. Guliakina, and D. Shunkevich, *Otkrytaja tehnologija ontologicheskogo proektirovaniya, proizvodstva i jekspluatatsii semanticheski sovmestimyh gibridnykh intellektual'nykh komp'yuternykh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint [Bestprint], 2021.
- [5] D. Pospelov, *Situacionnoe upravlenie. Teoriya i praktika [Situational management. Theory and practice]*. M.: Nauka, 1986.
- [6] P. Skobelev, "Situacionnoe upravlenie i mul'tiagentnye tekhnologii: kollektivnyj poisk soglasovannykh reshenij v dialoge [Situational management and multi-agent technologies: collective search for agreed solutions in a dialogue]," *Ontologija proektirovaniya [Ontology of designing]*, no. 2, pp. 26–48, 2013.
- [7] M. Wooldridge, *An Introduction to MultiAgent Systems — Second Edition*. Wiley, 2009.
- [8] V. Gorodetskii, V. Samoilov, and D. Trotskii, "Bazovaya ontologiya kollektivnogo povedeniya avtonomnykh agentov i ee rasshireniya [Basic ontology of autonomous agents collective behavior and its extension]," *Izvestiya RAN. Teoriya i sistemy upravleniya [Proceedings of the RAS. Theory and control systems]*, no. 5, pp. 102–121, 2015, (in Russian).
- [9] V. Tarasov, *Ot mnogoagentnykh sistem k intellektual'nykh organizatsiyam [From multi-agent systems to intelligent organizations]*. M.: Editorial URSS, 2002, (in Russian).
- [10] A. N. Kopajgorodskij and L. V. Massel, "Fraktal'nyj podhod k proektirovaniyu arhitektury informacionnykh sistem [Fractal Approach to Information Systems Architecture Design]," *Vestnik IrGTU [Bulletin of ISTU]*, no. 6(46), pp. 8–12, 2010.

Принципы решения задач в распределенных коллективах интеллектуальных компьютерных систем нового поколения

Шункевич Д. В.

В работе рассмотрен подход к организации решения задач в рамках распределенного коллектива интеллектуальных компьютерных систем, входящих в состав Экосистемы OSTIS (ostis-систем). Рассмотрена классификация агентов в рамках такой системы, а также принципы их взаимодействия.

Received 13.03.2023

Versioning Model of Neural Network Problem-Solving Methods in Intelligent Systems

Mikhail Kovalev
*Belarusian State University of
Informatics and Radioelectronics*
Minsk, Belarus
michail.kovalev7@gmail.com

Abstract—In the article, the design process of neural network problem-solving methods in the knowledge bases of intelligent systems is considered. The versioning model of neural network problem-solving methods, described in a specialized language for representation of neural network problem-solving methods, is proposed.

Keywords—problem-solving method, ontological approach, neuro-symbolic AI, artificial neural network

I. INTRODUCTION

The modern development of all directions of *Artificial intelligence* is aimed at building *intelligent systems* that automate more and more complex human activities. The current state in the field of developing *intelligent systems of a new generation*[1] shows that such systems should provide:

- unification of representation and coherence of different knowledge types and problem-solving methods;
- integration and convergence of different problem-solving methods in a single knowledge base to ensure consistency in the semantics of that set of methods;
- representation and interpretation of as many classes of problem-solving methods (programs) as possible.

Integration of different problem-solving methods in a single knowledge base guarantees consistency of semantics of this set of methods. When solving problems using such methods, the system does not communicate with the external environment by transferring input and output data. Instead, a single knowledge base allows the system to track changes in input knowledge in real time using a wide range of methods, which provides the ability to introspect and explain the decisions made by the system. A single *knowledge base for problem-solving methods and knowledge* used to solve them allows the system to reflect on the process of problem solving, to explain the reasons for its solutions, and to find mistakes there.

The actively developing *class of problem-solving methods* is *artificial neural networks* (ANNs). This is conditioned, on the one hand, by the rapid development of the theoretical foundations of artificial neural networks and on the other hand, by the increasing computing power of the machines used to train them.

Impressive results have been obtained in problem solving with artificial neural networks [2]. Among the positive characteristics of ANNs are their ability to effectively solve problems in the absence of known regularities, as well as their ability to solve problems without necessarily developing problem-oriented *problem-solving methods*.

However, there are serious problems with neural network problem-solving methods:

- Heuristic nature of the design process of *neural network problem-solving methods*. The process of selecting ANNs architectures and their training parameters places high demands on the knowledge level of ANNs engineers.
- Lack of explicit allocation of semantic connections between knowledge in the process of problem solving. They are highlighted implicitly, statistically, based on the data that was used for training. Lack of explicit allocation of meaning leads to the problem of the “*black box*” [3]. An entire field of *Explainable AI* has emerged, in which researchers attempt to explain the ANNs solutions [4], [5].

Formalization of ANNs in the *knowledge base* of the *intelligent system* together with other *problem-solving methods* allows negating the listed ANNs problems, since in such systems, the design problem of ANNs and the explanation problem of solutions for these ANNs are represented in one form for the whole knowledge base and can be solved using any of the represented *problem-solving method* from this *knowledge base*.

Frequently, the ANN is actively changed during the design and interpretation process (configuration of connections, number of layers, synapse weights, activation functions, etc.). To solve the problem of ANNs design, the system must be able to analyze the solutions of the same problem on different versions of the same *neural network problem-solving method* in order to evaluate the success of certain solutions in the design of this method, for example, the success of selection of activation functions, training sample, training algorithm, configuration of connections in layers, etc.

The purpose of this article is to develop an approach

to *versioning of neural network problem-solving methods* in the *knowledge base* of the *intelligent system*.

II. PROPOSED APPROACH

In order to solve the above problems, the OSTIS Technology is proposed. Intelligent systems developed using the OSTIS Technology are called ostis-systems. Any ostis-system consists of a knowledge base, a problem solver, and a user interface.

The problem solver performs the processing of fragments of the knowledge base. At the operational level, processing means adding, searching, editing, and deleting sc-nodes and sc-connectors of the knowledge base. On the semantic level, such an operation is an *action performed in the memory of an action subject*, where, in the general case, the subject is an ostis-system and the knowledge base is its memory. An *action* is defined as the influence of one entity (or some set of entities) to another entity (or some set of other entities) according to some purpose.

Actions are performed according to the set problems. A *problem* is a formal specification of some action, sufficient to perform this action by some subject. Depending on a particular class of problems, it is possible to describe both the internal state of the intelligent system itself and the required state of the external environment [6].

For classes of problems, classes of methods for their solution are formulated. A *problem-solving method* is defined as a problem-solving program of the corresponding class, which can be either procedural or declarative. In turn, a *class of problem-solving methods* is defined as a set of all possible problem-solving methods having a common language for representing these methods. The method representation language allows describing the syntactic, denotational, and operational semantics of this method.

Approaches to *integration of ANNs with knowledge bases* in *ostis-systems* are considered in [7]. Input-output integration approaches have been tested and described in [8], [9]. Full *integration of ANNs with knowledge bases*, i.e., using neural network problem-solving methods by formalizing them in an ostis-system knowledge base, are described in [10], which describes the *Denotational and Operational semantics of the Language for representation of neural network problem-solving methods (Neuro-SCP)*. The present work is a development of this language.

The *Language for representation of neural network problem-solving methods* allows representing and interpreting neural network methods in the ostis-system memory. This language is a sublanguage of the *SCP Language* [11]. Any method represented in the *SCP Language* is a fragment of the knowledge base, so the problem of *versioning of neural network problem-solving methods* in the *knowledge base* of the *intelligent system* is reduced to the problem of *versioning any knowledge base fragments*.

During the existence of the *intelligent system*, the state of fragments of its *knowledge base* (that is, the configuration of connections between signs in this knowledge base), as well as the neural network problem-solving method represented in the knowledge base, can change over time [12]. The need to account for the dynamics of knowledge changing over time in *knowledge bases* of *intelligent systems* is conditioned by the qualities inherent in intelligent systems, as well as the range of problems they solve. *Intelligent systems* must:

- maintain the relevance, adequacy, and accuracy of the knowledge stored in it at any point in time;
- remember the history of user and developer actions performed on *fragments* of the knowledge base in order to analyze them and support decision-making in other problems;
- allow performing reverse actions in case of abnormal situations;
- allow verifying the sources of unreliable knowledge and warn about inconsistencies in *knowledge bases*;
- adapt to the characteristics of its users and other *intelligent systems*;
- plan and initiate different kinds of problem solving.

Thus, to provide a higher level of intelligence of the system and support its life cycle, it is necessary to specify a set of methods and tools that allow solving these problems quickly and efficiently. One of such means for solving these problems is a ***Subsystem for versioning of knowledge base fragments***, embedded in any *intelligent system*, developed on the principles of the *OSTIS Technology* (i.e., in ostis-system), providing continuous versioning for various knowledge base fragments and analysis of their states.

A *knowledge base fragment* means a formal specification of any entity or concept sign represented in the *knowledge base* of a given system. That is, a *knowledge base fragment* is nothing but some semantic neighborhood or structure that includes knowledge about an object in the *subject domain of the knowledge base* of this system. The process of *versioning of a knowledge base fragment* implies a complete representation and description of its states, as well as providing capabilities and tools for processing and analyzing the states of this *knowledge base fragment*.

The *state of the knowledge base fragment* means the integration of the results of actions performed on this *knowledge base fragment* since its existence in the *knowledge base* (that is, since the initial state of this knowledge base fragment).

The *versioning of the knowledge base fragment* requires strict identification of all states from the beginning of the existence of this *knowledge base fragment*, that is, it requires the construction of the *bijective correspondence* between the *state of the knowledge base fragment* and its unique identifier in the whole knowledge base of the

system [13]. In the process of *versioning of the knowledge base fragment*, a tree-like linear structure of states of this knowledge base fragment (*tree of states of knowledge base fragments*) and a tree-like linear structure of state identifiers of this knowledge base fragment (*tree of state identifiers of knowledge base fragments*) are built. Both structures are represented in the SC-code.

The Subsystem for versioning of knowledge base fragments consists of the following components:

- *specification for the versioning model of knowledge base fragments* (i.e., documentation (knowledge bases) describing methods, tools, and algorithms of *versioning of knowledge base fragments*);
- *problem solver* [14], including:
 - an *Abstract sc-agent for generating the initial state of a knowledge base fragment in its state stack*;
 - an *Abstract sc-agent for integrating changes with the current state of a knowledge base fragment and adding the resulting state of the knowledge base fragment to its state stack*;
 - an *Abstract sc-agent for identifying the state of the knowledge base fragment in the stack of state identifiers of a given knowledge base fragment*;
 - an *Abstract sc-agent for getting the state of a knowledge base fragment by its identifier*;
 - an *Abstract sc-agent for getting the version of a knowledge base fragment by its identifier*.
- *program interface and user interface* to use the *versioning of knowledge base fragments*.

The *state of knowledge base fragments* is pairs of two sets. The elements of the first set are the membership arcs of the certain substructures for decomposing the more general structure, which in the version of this *state of the knowledge base fragment* have not been changed. The elements of the second set are the membership arcs of the certain substructures for decomposing the more general structure, which have been changed in this version. Both sets can be represented as a role relation (attribute set), denoting which and in what way the certain structures of the general structure of states are represented. These two sets play an important role in the operational semantics used when implementing agents to work with the history of structure changes.

Reconstructing a version by the *state of the knowledge base fragment* takes only one traversal of the subtree, i.e., a traversal of this state. Structures located in the hierarchy lower than the structure, which includes a membership arc of the composition of a more general structure, which in turn is an element in the attribute set of membership arcs of changed (unchanged) structures, are considered to be changed (unchanged) until an arc from the second attribute set is encountered, and those above the arc are considered to be unchanged (changed) until an arc from the first attribute set is encountered. This approach is easy to implement and does not require a large number

of copies for bindings of decomposition and membership arcs for changed structures, compared to the previous cases.

Let us consider an example of versioning a particular neural network method represented in *Neuro-SCP* and described in [10]. This method solves the classical “EXCLUSIVE OR” problem [15].

In *Neuro-SCP*, the implementation of this ANN is reduced to a single layer interpretation operator, for which the *matrix of synapse weights*, *threshold*, and *activation function* are defined. In Figure 1, the *neuro-SCP* interpretation operator of the *layer of a single-layer perceptron*, solving the “EXCLUSIVE OR” problem, is shown.

In Figure 2, the *single-layer perceptron* scheme, which solves this problem and from which the *Neuro-SCP* operator has been described, is demonstrated.

Any of these elements can be changed in the process of training and reconfiguration of the ANN. Within the proposed approach, such ANN will correspond to its own *tree of states*. Let us suppose that a given ANN has been trained and its *synaptic weight matrix* of a single layer has changed.

In Figure 3, a *tree of states* that stores two versions of the same ANN — before training and after training — is shown. The root of this tree is an instance of the history class. There are two states associated with the root, each of which is an oriented pair of sets.

For each state, the time of the beginning of the state existence is set. All elements describing the states are *temporal*, since they exist temporarily and each state can be replaced by another.

For the first state, the first set was empty, since it is the first state in the tree and it cannot refer to previous states. The second set considered above describes all elements that were added in this version. Since it is the first state, this set describes all membership arcs involved in the description of the operator.

The first set of the second state — the set of elements that have not changed from the previous state — is described using the constructions involved in describing the second set of the first state. The second set of the second state, on the other hand, describes only the changes or, to be more precise, only the membership arcs that have been added in the new state. Thus, it is possible to understand that the weight matrix has changed in the new state. It can be assumed that the ANN has been retrained.

The use of ANNs as a problem-solving method implies the use of an already designed and trained ANNs. However, the presence of a neural network method description language in ostis-system memory opens the way for automation of the processes of designing and training ANNs. Such automation is represented by separate classes of problems and the corresponding skills for their solution.

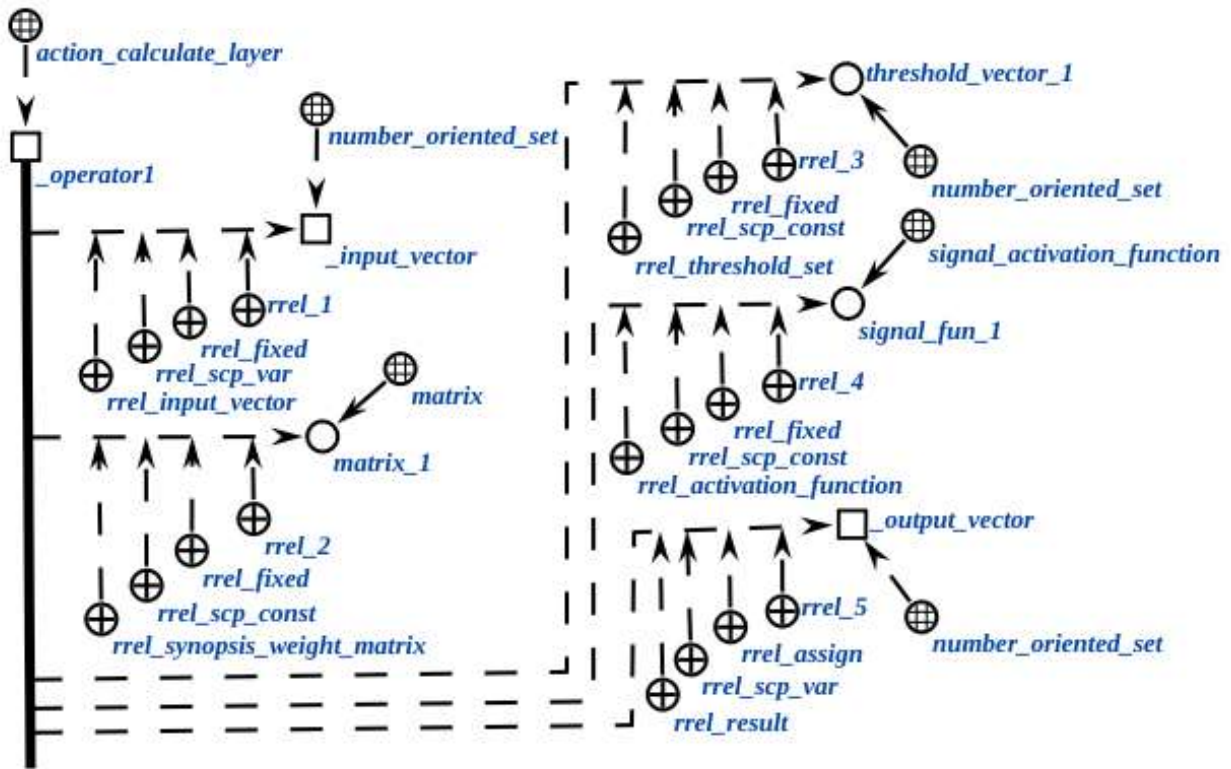


Figure 1. Operator for interpreting the ANN layer on Neuro-SCP

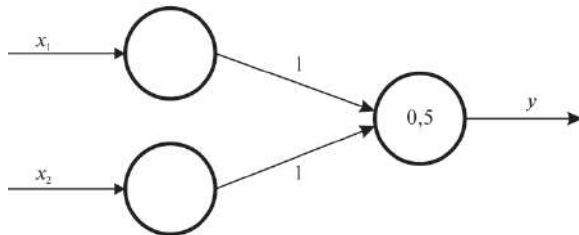


Figure 2. Scheme of a single-layer perceptron solving the "EXCLUSIVE OR" problem [15]

III. CONCLUSION

The described versioning model allows storing and restoring any state not only of the *neural network method* described with *Neuro-SCP* in the *knowledge base* but also any *knowledge base fragment* in principle with minimal effort.

The model can be used to describe not only what has been changed but also at what time and under what influences. If necessary, each state can be described from various angles — efficiency, validity, operational history, etc.

In the case of *convergence and integration of ANNs with knowledge bases of intelligent systems*, such a model will allow considering *ANNs* not only as a

problem-solving method but also as a design object. The automation of such design will become a task that can be posed by the same intelligent system, in which the neural network is described and interpreted.

The availability of unification of the knowledge representation describing the problem and the methods of solving these problems allows the whole arsenal of *problem-solving methods of the intelligent system* to be used for design. Such methods can achieve good results in design automation by accessing the knowledge describing the problem to be solved by the designed *ANNs*, the context of the problem to be solved, the history of the intelligent system solving similar problems, etc.

A further development of this work is the design and implementation of *intelligent design framework of ANNs*, which will automate various activities for *ANNs* designers.

ACKNOWLEDGMENT

The author would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help in the work and valuable comments, in particular, Vladimir Golenkov, Vladimir Golovko, Aliaksandr Kroschchanka, and Nikita Zotov.

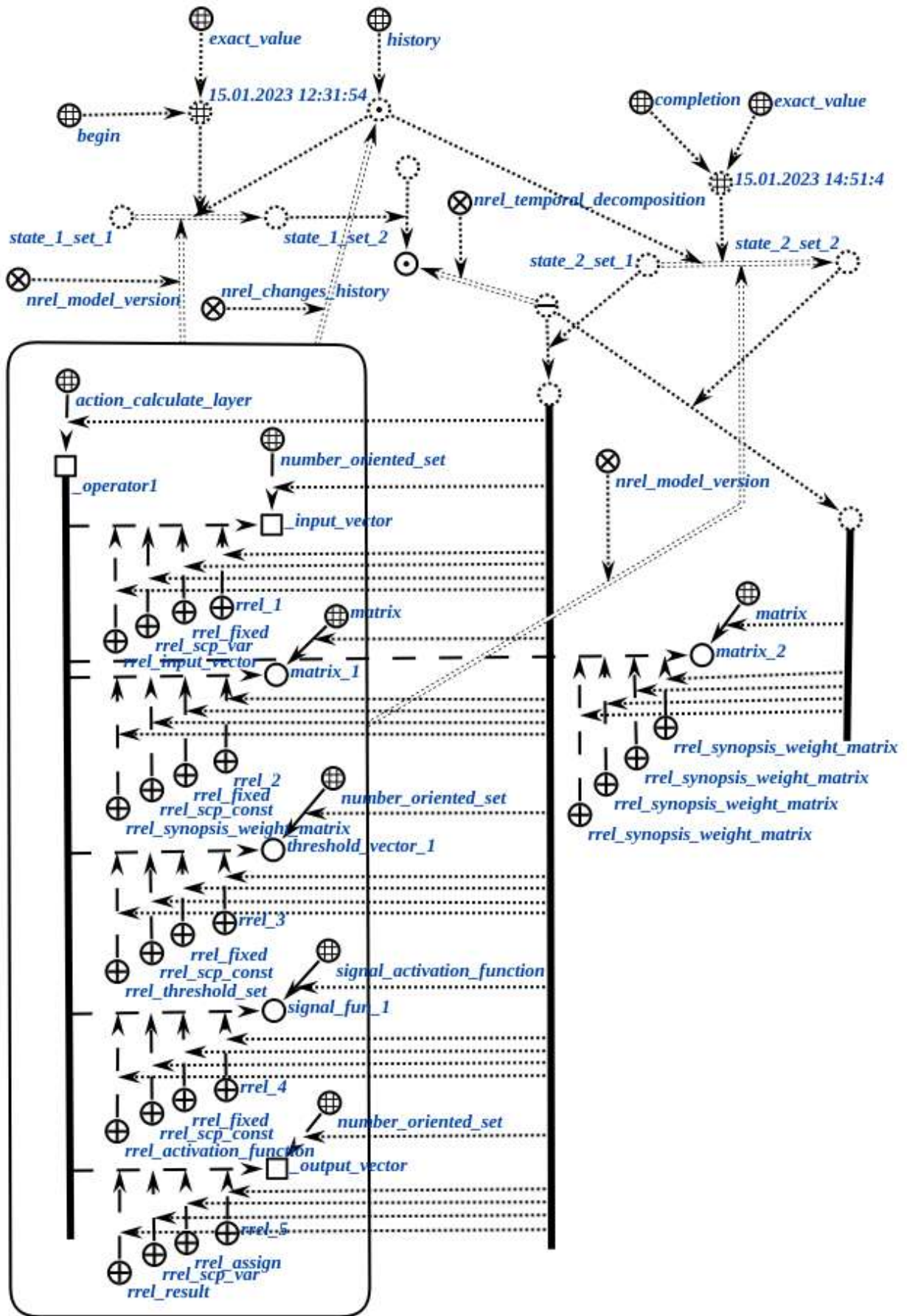


Figure 3. Example of an ANN tree of states

REFERENCES

- [1] V. V. Golenkov and N. A. Gulyakina, "Next-generation intelligent computer systems and technology of complex support of their life cycle," *Open semantic technologies for intelligent systems*, pp. 27–40, 2022.
- [2] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Q. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, 2021.
- [3] D. Castellevecchi, "Can we open the black box of AI?" *Nature News*, vol. 538, no. 7623, Oct 2016.
- [4] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier," 2016. [Online]. Available: <https://arxiv.org/abs/1602.04938>
- [5] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf>
- [6] D. Shunkevich, "Ontology-based design of hybrid problem solvers," *Open Semantic Technologies for Intelligent Systems*, pp. 101–131, 2022.
- [7] V. A. Golovko, V. V. Golenkov, V. P. Ivashenko, V. V. Taberko, D. S. Ivaniuk, A. A. Kroshchanka, and M. V. Kovalev, "Integration of artificial neural networks and knowledge bases," *Open semantic technologies for intelligent systems*, pp. 133–145, 2018.
- [8] V. Golovko, A. Kroshchanka, M. Kovalev, V. Taberko, and D. Ivaniuk, "Neuro-symbolic artificial intelligence: Application for control the quality of product labeling," *Open Semantic Technologies for Intelligent System*, pp. 81–101, 10 2020.
- [9] A. Kroshchanka, V. Golovko, E. Mikhno, M. Kovalev, V. Zahariev, and A. Zagorskij, "A Neural-Symbolic Approach to Computer Vision," *Open Semantic Technologies for Intelligent Systems*, pp. 282–309, 2022.
- [10] M. Kovalev, "Convergence and integration of artificial neural networks with knowledge bases in next-generation intelligent computer systems," *Open semantic technologies for intelligent systems*, pp. 173–186, 2022.
- [11] V. Golenkov, N. Gulyakina, and D. Shunkevich, *Otkrytaya tekhnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheskii sovmestimyykh gibridnykh intellektual'nykh komp'yuternyykh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint, 2021.
- [12] V. Ivashenko, "Semanticheskoe protokolirovanie processov obrabotki znaniy [Semantic logging of knowledge processing processes]," in *Information Technologies and Systems 2017 (ITS 2017) : Republic of Belarus, Minsk, 25 october 2017 year*, L. Y. Shilin, Ed. Minsk: BSUIR, 2017. [Online]. Available: <https://libeldoc.bsuir.by/handle/123456789/27633>
- [13] —, "Identifikaciya binarno porozhdaemykh sobytij processov obrabotki znaniy dlya semanticheskogo protokolirovaniya [Identification of binary generated events of knowledge processing processes for semantic logging]," in *Information Technologies and Systems 2018 (ITS 2018) : Republic of Belarus, Minsk, 25 october 2018 year*, L. Y. Shilin, Ed. Minsk: BSUIR, 2017. [Online]. Available: <https://libeldoc.bsuir.by/handle/123456789/33352>
- [14] D. Shunkevich, "Agentno-orientirovannye reshateli zadach intellektual'nykh sistem [Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems]," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 119–132, 2018.
- [15] V. A. Golovko and V. V. Krasnoproshein, *Nejrosetevye tekhnologii obrabotki dannykh [Neural network data processing technologies]*. Minsk : Publishing House of the BSU, 2017, (In Russ.).

Модель версионирования нейросетевых методов решения задач в интеллектуальных системах

Ковалёв М. В.

В статье рассматривается процесс проектирования нейросетевых методов решения задач в базах знаний интеллектуальных систем. Предложена модель версионирования нейросетевых методов решения задач, описанных на специализированном языке представления нейросетевых методов решения задач.

Received 29.03.2023

Reduction of Neural Network Models in Intelligent Computer Systems of a New Generation

Aliaksandr Kroschanka
Brest State Technical University
Brest, Belarus
Email: kroschenko@gmail.com

Abstract—In the article, an approach to reducing the tuning parameters of deep neural network models, which is based on the use of the pre-training method, is proposed. Examples of using this approach for model reduction are given for MNIST, CIFAR10, CIFAR100 datasets. Recommendations are given on the use of the proposed method in the context of integrating massive neural network models into the intelligent computer systems of a new generation based on the use of the OSTIS Technology.

Keywords—deep neural networks, model reduction, RBM, CRBM, pre-training, transfer learning, hybrid intelligent systems

I. INTRODUCTION

The development of hybrid intelligent systems that combine the use of various models and approaches is associated with integration difficulties. This problem can be successfully solved by using the OSTIS Technology [1], which makes it possible to develop such systems taking into account their semantic compatibility. An equally important factor is the quality of the particular components of the system.

Deep neural network models have recently become one of the most actively used components of hybrid systems. These models show impressive results in solving a wide variety of problems — recognition, detection, and segmentation of objects in photo and video images (for example, [2], [3]), generating annotations for photos, and generating images from a text description [4], generating texts of varying complexity ([5], [6]). Neural networks used to solve such problems contain millions and billions of adjustable parameters, as well as tens and hundreds of layers of neural network elements (Fig. 1).

Although artificial neural networks have recently been constantly expanding the boundaries of their application in various fields, it is the fact of the complexity of these models that gives rise to some conceptual problems and issues that hinder the process of widespread use of truly useful and effective neural networks.

Training remains the only possible way to specialize the model, since no pre-trained neural network can be used to effectively solve a specific problem. The only

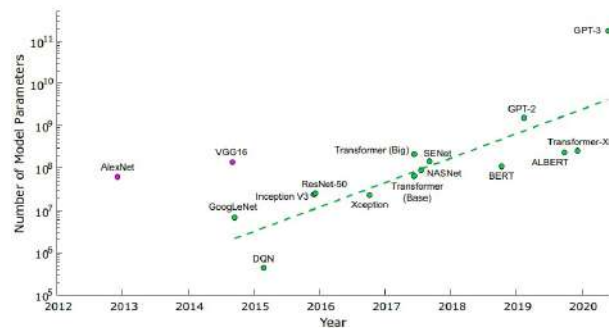


Figure 1. Evolution of parameters amount in deep neural networks [7]

possibility in this case is additional training of this model on specific data.

However, training such “heavy” models is not a trivial problem. It is often associated with the risk of overfitting, which results in excellent fit of the model to the training dataset but poor generalization ability. Most often, overfitting occurs when using a small training dataset. Another problem is the size of the models used, which makes the learning process slow and resource intensive even with the use of modern technical means.

Thus, the most lightweight modification of the widely known and used Llama model requires 4–8 video accelerators like NVIDIA A100 with 80 GB of video memory for additional training on user data ([8], [9]). And this, unfortunately, is far from the limit for such models. Thus, the OPT model with 175 billion adjustable parameters already requires 992 video accelerators of the same type for training [10]. Thus, the problem of additional training becomes unattainable for ordinary users and researchers, becoming the prerogative of large laboratories and companies.

Thus, it becomes critical to reduce the number of parameters used without losing the quality of the neural network model. Ideally, it is required to achieve the ability to run models on portable devices with limited computing capabilities.

As possible solutions to both the problems of overfitting

and unlimited growth in the number of parameters, the authors see the use of the pre-training procedure.

The following sections are organized as follows: in Section II, the problem of reducing deep neural networks in the context of intelligent computer systems of a new generation is described; in Section III, the proposed approach to reduce the number of parameters of neural network models is considered; in Section IV, the main practical results obtained are shown; finally, in Section V, the main conclusions on the proposed approach are represented and possible options for its development, including in the context of intelligent computer systems of a new generation, are described.

II. PROBLEM FORMULATION

It is known that neural networks have a certain degree of redundancy. Most often, this is conditioned by the use of an inconsistent number of model parameters and the size of the training dataset. If the number of adjustable parameters is greater than the volume of the training dataset, then the problems with the efficiency of training the model, or rather, with its generalizing ability, occur.

In addition, fully connected layers, in comparison with convolutional ones, contain a larger number of adjustable parameters, however, in computer vision problems, convolutional neural networks show significantly better results in terms of generalizing ability than fully connected ones. Thus, it is obvious that in fully connected networks with a larger number of adjustable parameters, they are used less optimally. It can be assumed that the specified “redundant” parameters can be discarded without a significant loss in the efficiency of the network.

Thus, the problem of reducing the neural network model is in identifying parameters that have small effect on the final result of the model and removing them. Moreover, such removal can be performed both logically (zeroing the corresponding weight coefficients and thresholds [11]) and architecturally (complete removal of a neural element if its weight coefficients and a threshold are equal to zero).

Therefore, **performing the reduction of the neural network model leads to its architectural change, reducing the number of neurons used on each layer.**

An important question that arises when performing the reduction concerns the very algorithm for discarding uninformative parameters. For current moment, several works have been proposed in which the authors reduce the dimension of neural network architectures (for example, [12], [13]).

Let us list the main advantages that the reduction of neural network models brings as separate components integrated into intelligent computer ostis-systems of a new generation.

Firstly, it becomes possible to automate the selection of the optimal architecture of the neural network, which prevents the core of the ostis-system from the formation

of redundant and confusing rules, most of which are formulated not on the basis of theoretical studies but purely empirically. In the case of reduction, it is sufficient to determine the maximum upper value for the number of neurons of each layer, without the need to select these parameters during a series of experiments.

Secondly, the work of neural network models as components of the ostis-system is accelerated by reducing the number of parameters used. Frequently, deep models can be slow and resource-intensive, which affects the overall system uptime, resulting in serious delays, especially in the absence of powerful computing tools.

Thirdly, the process of additional training of neural network models on user datasets is automated, which could be insufficient in size when training a neural network from scratch, which would lead to the effect of overfitting.

Thus, model reduction expands the possibilities of integrating neural network models as components of ostis-systems, making the models themselves more adaptive, faster, and more efficient, and solving this problem is an urgent problem.

III. PROPOSED APPROACH

Currently, there are two main approaches to pre-training deep neural networks: **I type** is based on unsupervised training of individual layers of the network, represented, for example, by restricted Boltzmann machines (RBM) [14], **II type** — on the use of special types of activation functions (ReLU), a large available training set, some special regularization techniques (for example, dropout), and special initialization of model parameters.

The selection of one or another approach to pre-train deep neural networks depends on the size of the training dataset. So, if the dataset is large, type II of pre-training is applied. Otherwise, type I of pre-training is used [15].

After performing the pre-training, an acceptable initial initialization of the parameters of the neural network model is achieved, which allows speeding up the additional training process, starting it with a smaller total error. In some cases, the nature of the data for additional training may change, differing from that used for pre-training (transfer learning). The goal of both the first and second types of pre-training is to achieve the ability to additionally train the model on small datasets without the appearance of the overfitting effect [16].

Thus, the process of training a deep neural network model with type I of pre-training consists of the following steps:

- 1) pre-training of the neural network by greedy layer-wise training, starting from the first layer (Fig. 2). Such training is carried out uncontrollably;
- 2) fine-tuning the parameters of the entire network using the backpropagation algorithm or the “wake-sleep” algorithm [17].

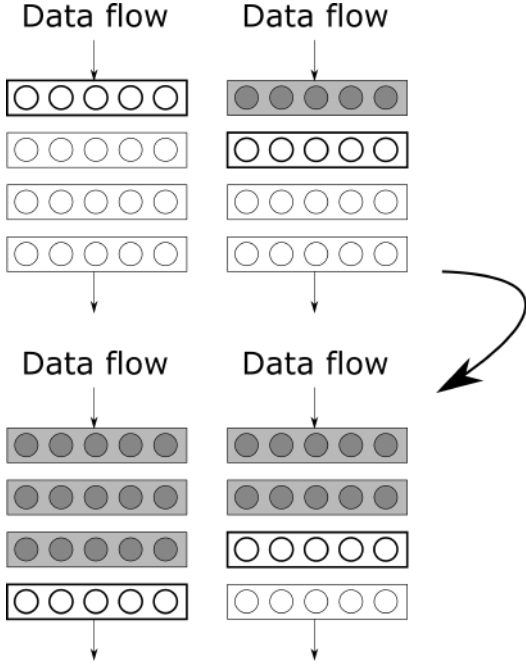


Figure 2. Greedy layer-wise algorithm

The proposed parameter reduction approach is based on the use of type I of pre-training proposed by G. Hinton (hereinafter referred to as the classical method) [18].

Let us give a brief description of this method. To do this, we consider the model of a restricted Boltzmann machine.

This model consists of two layers of stochastic binary neurons, which are interconnected by bidirectional symmetrical connections (Fig. 3). The input layer of neurons is called visible (layer X), and the output layer is called hidden (layer Y). The restricted Boltzmann machine can generate any discrete distribution if enough hidden layer neurons are used [19]. Let the visible layer contain n and the hidden layer contain m neurons.

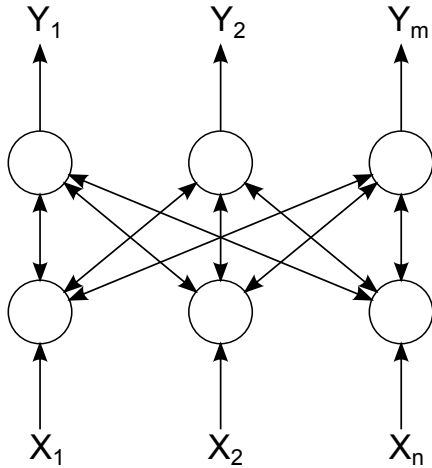


Figure 3. Restricted Boltzmann machine

The rules for online training of a restricted Boltzmann machine proposed in the classical method are as follows:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(k)y_j(k))$$

$$T_i(t+1) = T_i(t) + \alpha(x_i(0) - x_i(k))$$

$$T_j(t+1) = T_j(t) + \alpha(y_j(0) - y_j(k))$$

where $x_i(0), x_i(k)$ — the original input data of the visible layer that are restored by the neural network, $y_i(0), y_i(k)$ — the original output data of the hidden layer that are restored by the neural network. Data recovery is performed using the Contrastive Divergence (CD- k) algorithm.

In practice, this algorithm is most often used for $k=1$.

The above rules are relevant for the case of a deep fully connected neural network, however, they can be easily reformulated for the case of a deep convolutional network. In this case, the individual layers of the deep model are treated as convolutional restricted Boltzmann machines (CRBMs) [20].

In this case, the rules will look as follows:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(x_i(0) \otimes y_j(0) - x_i(k) \otimes y_j(k))$$

$$T_i(t+1) = T_i(t) + \alpha(x_i(0) - x_i(k))$$

$$T_j(t+1) = T_j(t) + \alpha(y_j(0) - y_j(k))$$

where \otimes denotes the convolution operation.

Thus, for a deep convolutional neural network, it is possible to combine several training options — with a representation in the form of CRBM (for the first convolutional layers) and in the form of RBM (for finalizing fully connected ones).

Previously, the authors proposed an approach that generalizes the classical approach and demonstrated its effectiveness for some problems (for example, [21]).

In the context of this approach, the learning rules are given, for the derivation of which the authors were guided by the idea of minimizing the total squared error of the network (the case of using CD- k):

$$E_s(k) = \frac{1}{2L} \left(\sum_{l=1}^L \sum_{j=1}^m (\Delta y_j^l(k))^2 + \sum_{l=1}^L \sum_{i=1}^n (\Delta x_i^l(k))^2 \right)$$

where $\Delta y_j^l(k) = y_j^l(k) - y_j^l(0)$, $\Delta x_i^l(k) = x_i^l(k) - x_i^l(0)$, L — a dimension of the training dataset.

The RBM online training rules in accordance with the proposed approach for CD- k are as follows:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha((y_j(k) - y_j(0))F'(S_j(k))x_i(k) + (x_i(k) - x_i(0))F'(S_i(k))y_j(0)),$$

$$T_i(t+1) = T_i(t) - \alpha(x_i(k) - x_i(0))F'(S_i(k)),$$

$$T_j(t+1) = T_j(t) - \alpha(y_j(k) - y_j(0))F'(S_j(k)).$$

For the CRBM case, the rules will take the form:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha((y_j(k) - y_j(0))F'(S_j(k)) \otimes x_i(k) + (x_i(k) - x_i(0))F'(S_i(k)) \otimes y_j(0)),$$

$$T_i(t+1) = T_i(t) - \alpha(x_i(k) - x_i(0))F'(S_i(k)),$$

$$T_j(t+1) = T_j(t) - \alpha(y_j(k) - y_j(0))F'(S_j(k)).$$

It is possible to prove the identity of these learning rules to the classical ones by using neurons with a linear activation function.

Let us consider an approach for reducing the parameters of a fully connected neural network based on the use of pre-training procedure. The first and fourth stages of this procedure are equal to the stages of performing the first type of pre-training. During the execution of additional stages 2–3, sparse connections are formed between the input and output neurons of the layer and its dimension is reduced by zeroing some of the parameters that are not used in fine-tuning and further using of the neural network model (Fig. 4):

- 1) Pre-training of a neural network represented as a sequence of restricted Boltzmann machines according to greedy layer-wise algorithm.
- 2) Zeroing parameters of the neural network that do not exceed some specified threshold $t > 0$. In other words, the parameters falling within the interval $[-t, t]$ are excluded and are not used in further training.
- 3) Architectural reconfiguration of the neural network, during which the neurons that are not involved in the formation of the output activity of the network (neurons with completely zero weight coefficients) are removed.
- 4) Fine-tuning of the resulting simplified architecture, for example, by backpropagation algorithm.

At stage 3, zero columns and rows of the layer weight matrix are removed. At the same time, the corresponding elements of the layer threshold vector are deleted and consistent deletion of rows and columns of the next or previous layers is ensured (this is done to avoid violation of consistency between the dimensions of the matrices and vectors of neighboring layers).

IV. RESULTS

Let us demonstrate the effectiveness of the proposed approach using the example of reducing various architectures of fully connected neural networks used to classify images from MNIST [22], CIFAR10 and CIFAR100 [23] datasets. These datasets are classic for testing the performance of machine learning models.

We conducted a series of experiments, including various datasets, architectures, and pre-training options used. Within the same dataset and NN architecture, the current initialization of the parameters was saved to be able to

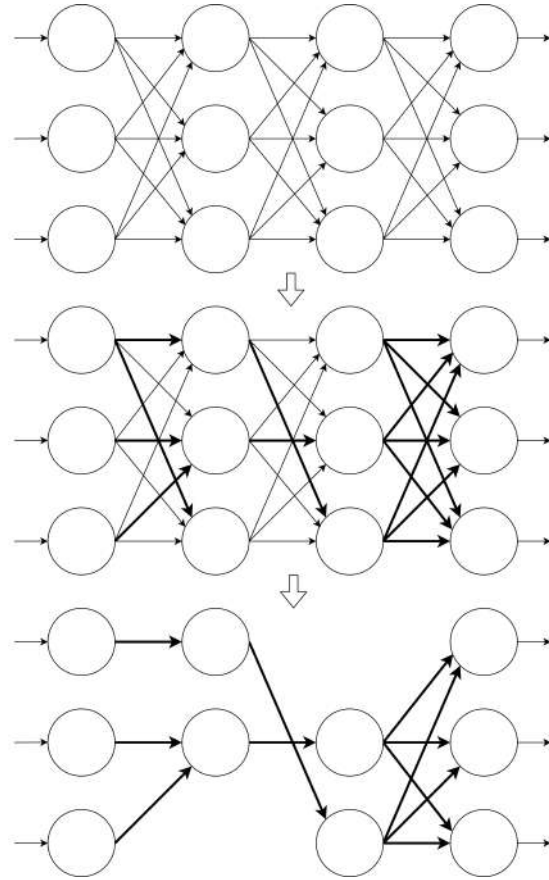


Figure 4. Method of reducing parameters on the example of fully connected layers

compare the effectiveness of different variants of the pre-training procedure.

Below, for the considered datasets, the main parameters are given, including the learning rate, mini-batch size, momentum parameter, and the number of epochs for pre-training and fine-tuning of models (Table I).

Table I
MAIN TRAINING PARAMETERS

Stage	Parameter	Value
Training	Learning Rate	0.05-0.1
	Mini-batch size	100
	Momentum parameter	0.9
	Number of training epochs	50-100
Pre-training	Learning Rate	0.05-0.2
	Mini-batch size	32-100
	Momentum parameter	[0.5, 0.9]
	Number of training epochs	10

As a result of the computational experiment, results were obtained for various datasets, NN architectures, and values of reduction parameter t (Table II-VI).

As can be seen from the above results, the considered

Table II
RESULTS OF TRAINING – MNIST, 784-800-800-10

Type	Efficiency, %, Classic / REBA	Tunable parameters, Classic / REBA	Reduced parameters, %, Classic / REBA
wr	98.63 / 98.33	1276810 / 1276810	0/0
t=0.2	98.61 / 98.27	233760 / 279635	81.69 / 78.1
t=0.5	98.03 / 98.05	32524 / 32817	97.45 / 97.43
t=0.8	97.1 / 96.48	17061 / 12217	98.66 / 99.04

Table III
RESULTS OF TRAINING – MNIST, 784-1600-1600-800-800-10

Type	Efficiency, %, Classic / REBA	Tunable parameters, Classic / REBA	Reduced parameters, %, Classic / REBA
wr	98.76 / 98.37	5747210 / 5747210	0/0
t=0.2	98.51 / 98.55	710734 / 781103	87.63 / 86.41
t=0.5	98.01 / 98.03	54709 / 43867	99.05 / 99.24
t=0.8	96.9 / 93.08	25385 / 14914	99.56 / 99.74

Table IV
RESULTS OF TRAINING – CIFAR10,
3072-1024-512-256-128-64-10

Type	Efficiency, %, Classic / REBA	Tunable parameters, Classic / REBA	Reduced parameters, %, Classic / REBA
wr	58.56 / 55.85	3844682 / 3844682	0/0
t=0.2	58.69 / 54.37	409211 / 227072	89.36 / 94.09
t=0.5	42.08 / 41.2	29033 / 11320	99.24 / 99.71
t=0.8	23.02 / 10.0	10058 / 4886	99.74 / 99.87

Table V
RESULTS OF TRAINING – CIFAR10, 3072-512-256-128-64-10

Type	Efficiency, %, Classic / REBA	Tunable parameters, Classic / REBA	Reduced parameters, %, Classic / REBA
wr	57.28 / 53.69	1746506 / 1746506	0/0
t=0.2	56.83 / 41.72	220037 / 126846	87.40 / 92.73
t=0.5	45.29 / 44.93	20431 / 11383	98.83 / 99.35
t=0.8	10.0 / 10.0	8599 / 3797	99.51 / 99.78

architectures generally retain their generalizing properties, being reduced by more than 80 percent, and even with a greater degree of reduction, they demonstrate good generalizing ability.

It is also possible to notice that the more adjustable parameters in the model, the more efficient the reduction is. However, with an increase in the reduction parameter, the efficiency of the original network gradually decreases, since the reduction begins to concern the parameters that

Table VI
RESULTS OF TRAINING – CIFAR100,
3072-3072-1024-512-256-128-64-100

Type	Efficiency, %, Classic / REBA	Tunable parameters, Classic / REBA	Reduced parameters, %, Classic / REBA
wr	20.84 / 21.63	13290788 / 13290788	0/0
t=0.2	20.77 / 21.01	1304525 / 703319	90.18 / 94.71
t=0.5	13.4 / 1.0	49847 / 24636	99.62 / 99.81
t=0.8	2.67 / 1.0	21329 / 16977	99.84 / 99.87

affect the final result.

The obtained results substantiate the possibility of pre-training a deep neural network using an uncontrolled procedure without obtaining the effect of overfitting and reducing the efficiency of the model, since in the process of pre-training, the influence of certain model parameters on the final output activity of the network is actually reduced. Such parameters are “parasitic” in nature and are actually a factor of the model overfitting. At the model additional training stage, they are not modified and can be removed after the pre-training stage.

V. CONCLUSION

In the article, the problem field generated by the use of modern deep neural networks is defined. The main advantages for integrated computer systems that appear when using reduction as a way to reduce the dimensionality of neural networks are determined.

An approach to the implementation of the method of reducing the parameters of deep neural networks based on the use of pre-training is proposed. A review of the rules for performing unsupervised learning of restricted Boltzmann machines and convolutional restricted Boltzmann machines is given. The obtained theoretical results are used for pre-training of deep neural networks.

Experimental studies of the proposed reduction method were carried out, which confirmed its effectiveness.

The authors see the further development of the proposed approach in obtaining practical results for known deep architectures of models used to solve problems of computer vision and natural language processing.

REFERENCES

- [1] V. V. Golenkov, N. A. Gulyakina, and D. V. Shunkevich, *Otkrytaya tekhnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheskikh sovremennykh gibridnykh intellektual'nykh komp'yuternykh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*. Minsk, Bestprint, 2021, (In Russ.).
- [2] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” 2022.
- [3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” 2018.

Редуцирование нейросетевых моделей в интеллектуальных компьютерных системах нового поколения

Крощенко А. А.

Статья посвящена разработке метода редуцирования глубоких нейронных сетей в контексте интеграции подобных моделей в ostis-системы. Предлагается альтернативный подход к обучению глубоких нейронных сетей, базирующийся на использовании RBM и CRBM. Предлагается метод для снижения размерности “тяжелых” моделей. Полученные теоретические результаты подтверждаются вычислительными экспериментами, демонстрирующими эффективность предложенного подхода к редуцированию.

Received 12.03.2023

- [4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” 2022.
- [5] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, “Llama: Open and efficient foundation language models,” 2023.
- [6] OpenAI, “GPT-4 Technical Report,” 2023.
- [7] L. Bernstein, A. Sludds, R. Hamerly, V. Sze, J. Emer, and D. Englund, “Freely scalable and reconfigurable optical hardware for deep learning,” *Scientific Reports*, vol. 11, 02 2021.
- [8] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, “Alpaca: A strong, replicable instruction-following model,” 2023. [Online]. Available: <https://crfm.stanford.edu/2023/03/13/alpaca.html>
- [9] —, “Stanford alpaca: An instruction-following llama model,” https://github.com/tatsu-lab/stanford_alpaca, 2023.
- [10] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, “Opt: Open pre-trained transformer language models,” 2022.
- [11] A. Kroshchanka and V. Golovko, “The reduction of fully connected neural network parameters using the pre-training technique,” in *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 2, 2021, pp. 937–941.
- [12] H. Mostafa and X. Wang, “Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization,” 2019.
- [13] S. Han, J. Pool, J. Tran, and W. J. Dally, “Learning both weights and connections for efficient neural networks,” 2015.
- [14] P. Smolensky, “Information processing in dynamical systems: Foundations of harmony theory,” *Parallel Distributed Process*, vol. 1, 01 1986.
- [15] V. Golovko and V. Krasnoprosin, *Neural network data processing technologies*. BSU, 2017. [Online]. Available: <https://elib.bsu.by/handle/123456789/193558>
- [16] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, no. 521 (7553), pp. 436–444, 2015.
- [17] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, “The “wake-sleep” algorithm for unsupervised neural networks,” *Science*, vol. 268, no. 5214, pp. 1158–1161, 1995. [Online]. Available: <https://www.science.org/doi/abs/10.1126/science.7761831>
- [18] G. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, pp. 1527–54, 08 2006.
- [19] Y. Bengio, “Learning deep architectures for AI,” *Foundations and Trends in Machine Learning*, no. 2(1), pp. 1–127, 2009.
- [20] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML ’09. New York, NY, USA: Association for Computing Machinery, 2009, p. 609–616. [Online]. Available: <https://doi.org/10.1145/1553374.1553453>
- [21] V. Golovko, A. Kroshchanka, and E. Mikhno, “Deep Neural Networks: Selected Aspects of Learning and Application,” in *Pattern Recognition and Image Analysis*. Cham: Springer International Publishing, 2021, pp. 132–143.
- [22] Y. LeCun and C. Cortes, “MNIST handwritten digit database.” [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [23] A. Krizhevsky, “Learning multiple layers of features from tiny images,” pp. 32–33, 2009. [Online]. Available: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>

Neural Network Software Technology Trainable on the Random Search Principles

Victor Krasnoproshin and Vadim Matskevich

Belarusian State University

Belarus, Minsk, Nezavisimosti av. 4, 220030

Email: krasnoproshin@bsu.by, matskevich1997@gmail.com

Abstract—The paper deals with a state-of-art neural technology programmed implementation problem in which the training process is based on random search algorithms.

Training neural networks is a typical optimization problem. At the initial stage of neural network technologies development, various variants of gradient methods were traditionally used to solve such problems. Such methods, as a rule, met the requirements for the problem in terms of quality and speed of training. However, with the appearing of a new class of applied problems, the situation has changed. The traditional approach to training using gradient methods did not always meet the requirements of the applied problem in terms of the resulting solution quality.

The paper proposes one of the options for the software implementation of neural network technology (in the form of a framework) according to the ostis 2021 standard, in which random search algorithms are used to train neural networks.

Keywords—framework, neural network, training, random search algorithms, annealing method

I. INTRODUCTION

In modern society, digital data processing technologies based on artificial intelligence methods are rapidly developing. In particular, neural network technologies based on various artificial neural networks architectures have become widespread.

Due to their high flexibility and the ability to tune to the subject area, they are actively used to solve a wide class of applied problems. However, neural network tuning for the problem being solved (training) is a time-consuming process.

Automation of the training process is effectively solved within the existing frameworks [1]. They allow us to simplify the neural networks training process by using already implemented training algorithms. The use of gradient optimizers inside such frameworks is quite justified. Gradient methods have a high convergence rate and in practice provide an acceptable solution quality obtained. When developing the first automated neural networks training systems, there wasn't a wide variety of computing devices, so they didn't have cross-platform property. However, with the computing technology development, more and more calculations are transferred from the central processor to connected computing devices. This allows us to use simultaneously a large number

of devices and significantly increase the efficiency of computing. Moreover, modern frameworks have become cross-platform. However, as digital technologies develop, the class of applied problems for which the solution quality obtained is critical is constantly expanding.

It should be noted here that many modern frameworks use gradient optimization methods, which do not always guarantee the optimal solution achievement. Consequently, when solving such applied problems, they are not effective enough, which makes the problem of developing a software package with alternative training methods up to date.

The paper proposes a framework's software implementation variant, in which random search algorithms are used to train neural networks.

II. PROBLEM ANALYSIS

Currently, a wide range of applied problems is solved using neural network technologies implemented in the form of frameworks. This technology is a set of software and algorithmic tools that implement the architectures of various types of neural networks focused on solving various classes of applied problems.

Today, there are a number of frameworks for solving machine learning problems. Among the most popular, in particular, the following can be distinguished.

MXNet is a high-performance and cross-platform framework that is widely used in solving applied problems. However, this framework has certain drawbacks. This is not a very convenient user interface compared to simpler frameworks and a rather meager range of optimization algorithms. It only supports some modifications of gradient descent. This framework completely lacks support for random search methods.

Tensorflow 2. is cross-platform and has a simple user interface. Currently, it is the most common framework for applied problems solving. Supports learning with various gradient methods and genetic algorithm. The disadvantages include insufficiently high performance, since it contains the costs of high-level programming languages and a poor variety of non-directional optimization algorithms.

Caffe 2 is a high performance cross platform framework. However, it lacks support for recurrent neural networks

and non-directional training algorithms. This framework significantly limits the class of constructed neural networks and has all the disadvantages of gradient methods.

Thus, today the following disadvantages are typical for modern frameworks.

Most of them either have a limited set of training algorithms, or support a limited neural networks class, or have low performance.

III. FRAMEWORK ARCHITECTURE AND STRUCTURE

Consider a variant of a software package implemented in a form of framework in which random search algorithms are used to train neural networks.

The software package was developed in C++ using the OpenMP and OpenCL libraries. OpenMP libraries provide efficient organization of parallel computing within a single processor, while OpenCL provides compatibility and parallel computing on a wide class of computing devices.

According to the ostis 2021 standard, the framework can be described as follows:

framework

```
⇒ decomposition*:
  {
  • algorithms library
  • train behaviour parameters
  • architecture library
  • database
  • compress/decompress module
  • predict module
  • load/save module
  }
```

algorithms library

```
⇒ FAST_ANNEALING
⇒ SLOW_ANNEALING
⇒ GENETIC
⇒ SGD
⇒ MOMGRAD
⇒ ADAM
⇒ FTML
```

train behaviour parameters

```
⇒ NO_TRAIN
⇒ JUST_TRAIN
⇒ NEW_TRAIN
⇒ CONTINUE_TRAIN
```

architecture library

```
⇒ RBM_BERNOULLI_BERNOULLI
⇒ RBM_GAUSS_BERNOULLI
⇒ AUTOENCODER
⇒ PERCEPTRON_NN
⇒ CONV_LAYER
⇒ POOLING_LAYER
```

PERCEPTRON_NN

```
⇒ part*:
   ActivationFunction
```

AUTOENCODER

```
⇒ part*:
   ActivationFunction
```

ActivationFunction

```
⊃ NONE
⊃ BIPOLYARSIGM
⊃ SIGM
⊃ ReLU
⊃ SOFTMAX
```

The developed software package consists of the following main modules: two libraries (algorithms and architectures of neural networks), a database and a database with configuration files (for setting up algorithms from the library), modules (for execution on connected computing devices and the neural networks functioning), and finally, user interaction interface.

The database contains all the necessary data sets for training neural networks. The data is loaded at the user request. The framework has built-in methods for generating various types of samples (from the requested data) for training and testing neural networks.

The algorithm library contains a wide range of different optimization algorithms: simple gradient, moment and adaptive moment methods, following the moving leader method, genetic algorithm and annealing method. Configuration files are loaded during the framework initialization and contain sets with optimal parameter values, which, if necessary, are recalculated taking into account the neural network architecture. Execution Modules on connected computing devices are loaded as they are found. The framework also has a high degree of flexibility, it can be executed on a limited number of processor threads (the limit can be adjusted), on several connected computing devices. In addition, a completely single-threaded execution mode is possible.

The framework allows assembly without the use of parallel computing on connected devices and the processor. For this, the constants.h file contains the *DISABLE_OPEN_CL*, *DISABLE_OPENMP* constants. When set to non-zero values, the framework disables the ability to use parallel computing. This allows the framework to be independent of the settings and computer configuration. The framework has additional constants *ENABLE_FAST_MATH* and *ENABLE_NORMAL_DISTR*. They allow us to activate the possibility of using accelerated trigonometric functions, for example, to generate a normal distribution. They also activate the tabled generator of basic random variables. The size of the table in the framework is more than a billion, so the random numbers repetition is not critical. To speed up trigonometric

functions, the segment of valid argument values is divided into a fixed number of parts (the accuracy is controlled by parameters). For each part, the value of cosine and sine is calculated and stored in a special table. The acceleration of calculations is achieved in the many times repeated use of tabular approximate values instead of the full calculation of the function value.

To train all non-recurrent networks with gradient methods, the back-propagation algorithm is used to calculate the gradient. For restricted Boltzmann machines, the gradient calculation uses the Constructive Divergence (CD-1) algorithm. Then, the user-selected gradient modifications are applied to the result.

The user interface contains a fairly wide range of functionality, covering all the procedures necessary for convenient framework use. This is loading (saving) a neural network from a hard disk, constructing neural networks based on the architecture base, forming a training dataset based on selected objects classes and data. Neural networks training process implementation, using various types of algorithms. Solving applied problems, color image compression and object recognition.

The neural networks training process can be described as follows. First, the user loads the training data from the database and calls the building training dataset methods (if needed). It then declares the neural network skeleton on which the neural network will be built. To do this, the framework implements a special fakeDeepNN data type. Using the prebuildDeepNN method, the user tells the designed neural network the images resolution, the images type (black and white, grayscale or color), indicates whether the network will be convolutional and assigns (if necessary) the future network a serial number.

Then, a neural network is constructed to the created skeleton (by sequentially adding different types of layers). The need for layer decomposition is indicated when adding the first layer of the network. In other layers, the decomposition size is calculated automatically based on the layer sizes entered by the user.

At the end of the neural network design process, the user declares an object of deepNN type and calls the buildDeepNN method. Passes the previously created network skeleton as a parameter. This function, in addition to connecting individual fragments into a single network, checks the correctness of the constructed network. For example, the incompatibility network layers' sizes, the incompatibility of layers' types (for example, one layer generates continuous data, and the next layer receives discrete data, or vice versa). If an error is found, the program generates the appropriate error and exits.

In the next step, the user sets the neural network training settings. The framework is highly flexible and allows for many different training modes. To do this, it declares an object of a special type TrainingSettings and, by calling the object's method for adding settings for a

separate layer, sequentially sets the training settings for all layers of the network. In the training settings of a separate layer, the following options are set: optimization algorithm, training style, layer training time, number of objects in the training and validation sets.

Setting the training style allows you not to train individual neural networks' layers, but to load network's trained fragments from the hard disk. The framework also supports the partial training option. To do this, the training style indicates: simple training or initial training is carried out — after which additional training of the layer is allowed, or continuation of training — the layer continues training. This option is useful when training large neural networks or when using low-power computing devices.

In the case of initial training, in addition to the trained network, the training state of a separate layer is stored. The description of the learning state depends on the optimizer chosen. In the case of continuing training, in addition to the initial training, the saved learning state of the neural network layer and the partially trained layer are loaded.

The training time is set separately for each layer. Inside all optimizers, there is a built-in implementation of the timer function that controls the time spent on training the layer. As soon as the time allocated for training has expired, the training process for this layer is completed.

The sizes of the training and validation sets allow us to control the amount of data required for layer-by-layer neural network training.

After creating a neural network and setting up the training of its individual layers, the user calls the constructed network training function using the trainDeepNN function. With this function, the user informs about the network being trained, training settings, some information about the input data and the need to use external devices for training.

The trainDeepNN function is key to the framework. At the beginning of the execution, if necessary, it scans the connected computing devices and initializes the executable modules on them. The function then checks that the the network layers training settings are correct. For example, it checks for the trained layers presence, that do not require training or will be retrained, etc. This function loads all settings of the entire algorithms library and optimizer selected by the user. Inside this function, a complex interaction with the input data for their decomposition (if necessary), linking the layers of the trained network, creating and deleting service buffers for the optimizer calling the training of the corresponding layer type by the corresponding optimizer, and other technical details are implemented. This method, at the end of training, saves the trained network to the hard disk.

The user can also call the compressImages and predict functions. Both functions take a trained neural network

and input data. The first of them performs data compression and returns their compressed image, the second returns a set of labels. The number of labels corresponds to the number of sended images. The label specifies the object class that the neural network has detected in the image.

IV. ARCHITECTURES LIBRARY

The architectures library fully fits into the ostis system described in [2] [3]. The software package supports the following types of neural networks: restricted Boltzmann machine of Gauss-Bernoulli and Bernoulli-Bernoulli types, autoencoders, decomposition of network layers from the above types, multilayer perceptrons, convolution layers, pooling layers.

A small number of frameworks support restricted Boltzmann machines, although they are used for key frames detecting in video sequences [4] filtering data [5], encoding key phrases in information retrieval [6] [informational retrieval]. Based on supported autoencoders and restricted Boltzmann machines, the framework allows us to design deep belief networks for data compression and preprocessing for subsequent data classification.

Support for sampling and convolution layers, multilayer perceptrons allows us to design deep convolutional neural networks to build neural network classifiers that are used for a wide class of applied problems.

Thanks to the support for the decomposition of individual layers, the framework allows you to significantly reduce the number of tunable network parameters, reduce the required amount of data for training, and increase the efficiency of parallelizing the neural network layers training.

The supported wide library of architectures allows us to design neural networks of almost any architecture, which makes the framework a universal tool for neural network data processing.

V. TRAINING ALGORITHM BASED ON THE ANNEALING METHOD

To solve the problem of efficiency of neural networks training algorithms, there are algorithms based on random search. The annealing method is the most promising random search algorithm for training neural networks. The paper proposes the following training algorithm based on the annealing method.

Let an objective function F be defined on a finite set of admissible solutions Ω and for each element $x \in \Omega$ of which the set of neighboring elements is $N(x) \subset \Omega$ given. The conditional optimization problem in this case can be specified as a triple (Ω, F, N) . Let us consider the possibilities of its solution using the annealing method. The algorithm includes the following main steps.

Preliminary stage. Initialization of the neural network initial state $Net_0 = Net(x_{10}, x_{20}, \dots, x_{m0})$ and temperature sequences T_0, T_1, \dots, T_k , related by the ratio:

$$T_k = \frac{T_0}{\ln(k+2)}, k > 0,$$

where T_0 - present value.

General k-th iteration.

Step 1. Random value generation. Generated m uniformly distributed on the segment from zero to the number of parameters in the set of discrete random variables a_1, a_2, \dots, a_m . Generated m random permutations of length equal to the number in the set of parameters. The first a_1, a_2, \dots, a_m elements of each permutation define the indexes of the parameters to be changed in each set of parameters, respectively.

Step 2. New solution generation. For each changing parameter, a uniformly distributed on the segment $[-l/2; l/2]$ random value b is generated. Value l depends on what set is changing parameter belongs to and equal to l_1, l_2, \dots, l_m respectively. Values l for each set are given as algorithm parameters.

Let x_i is changing parameter and x'_i is its new value, then:

$$x'_i = x_i + b$$

Step 3. Transition principle. Let x be a current solution and y was generated on step 2 as new solution. Then solution x' on the next iteration is determined in a following way:

$$P(x' = y|x) = \min\{1, \exp(\frac{F(x) - F(y)}{T_k})\}$$

Step 4. Stop criteria. If the time for training the neural network has expired, then the algorithm ends. Otherwise, the transition to the next iteration is performed.

Previously, it was proved that the proposed algorithm converges in probability to the optimal solution, and from any initial solution [7].

VI. GENETIC ALGORITHM FOR NEURAL NETWORKS TRAINING

Additionally, a special genetic algorithm modification for neural networks training was developed for the framework.

Preliminary stage. Generation of several random solutions. Each individual solution is a complete neural network, the architecture of which is set before training algorithm running and doesn't change. The number of solutions N is a parameter of the algorithm. For each solution, the value of the quality functional to be optimized is calculated.

General k-th iteration.

Step 1. The worst one is chosen from the current set of solutions.

Step 2. For the chosen solution, a "mutation" is performed – generation of a new solution from the current

one according to the same scheme as for the developed annealing algorithm. The only difference is that the values of the annealing parameters may differ from the genetic algorithm.

Step 3. For the obtained solution, the quality functional value is calculated.

Step 4. If the value of the obtained functional for the new solution is better than for the current one, then the current solution is replaced with a new one, otherwise the new solution is discarded.

Step 5. The solution b is randomly selected from the set of current solutions. The best solution a is also selected from the set.

Step 6. "Crossbreeding". For all values of the solution parameters a, b , the following calculations are made.

$$\begin{cases} d_i = b_i - a_i \\ c_i = a_i + d_i * \alpha \\ \alpha \in [0; \phi], 0 < \phi \leq 1 \end{cases}$$

where α is uniformly distributed random variable on the segment, ϕ is an algorithm parameter.

Step 7. For the obtained solution, the quality functional value is calculated.

Step 8. In the set of solutions, the worst one is chosen. If it is worse than the new solution, then it is replaced by a new solution, otherwise the new solution is discarded.

Step 9. Stop criteria check. If the time for training the neural network has expired, then the algorithm ends. Otherwise, the transition to the next iteration is performed.

VII. SOFTWARE PACKAGE IMPLEMENTATION FEATURES

The base of algorithms is developed taking into account the cross-platform framework property. When developing optimizers, the computing devices architectural features were taken into account. For example, video cards are characterized by a large number of low-power computing cores, which requires good scalability from the parallelization algorithm. In addition, the performance of the video card mainly depends on the efficiency of working with video memory, since it is very slow. This imposes requirements on the locality of the data used for the most efficient video card's cache use. It was also necessary to implement the interaction of the video card with the processor for the results collection and other data transfer.

The above requirements have resulted in each optimizer being implemented twice using different algorithms. One set of algorithms is used to train neural networks on a processor, the other set is used on video cards.

The most time-consuming part of training a neural network is moving data through the training layer. The input data is presented in matrix form, as are the parameters of the neural network. In a simplified form, moving data through a network layer is a matrix multiplication of data by network parameters. Thus, to ensure efficient

training, it is necessary to implement efficient matrix multiplication.

To ensure the best data caching, a block data multiplication algorithm was used. It is worth noting that improving caching increases the training efficiency both on the video card and on the processor. The block size for the matrices was selected empirically. For the processor, the optimal block size was 25, for the video card — 16. It should be noted that the data size, the amount of data, and the network size are often not a multiple of these values, so incomplete blocks were filled with zeros at the end of the real data. This approach makes it possible to increase the training efficiency by more than 70% on the processor and more than twice on the video card. This fact makes it possible to compensate for the significantly complicated implementation of optimizers and the entire complex as a whole due to the formation and disbanding of data blocks and work inside data matrix blocks.

It is known that random search methods have slow convergence and require a large amount of computation at a single iteration. Therefore, in order to increase the efficiency of annealing training, a calculation hiding approach (when using connected devices for training) was used.

Each iteration of the annealing method, as mentioned above, consists of 4 stages. All stages, except for calculating the functional value for a new solution, are performed by the processor. The most time-consuming stage is the calculation of the functional value and, with a small network architecture, the new solution generation. The computational power of a video card is on average 20 times higher than that of a processor. This leads to the fact that when training small neural networks, a significant part of the time is spent on generating new solutions. All stages in a separate iteration are performed strictly sequentially.

The generation of a new solution consists of two stages: determination of the number and selection of changeable network parameters; changing the values of the selected network parameters.

To reduce the execution time of a single iteration, you can use the fact that the definition and choice of network parameters to be changed don't depend on the stage of making a new decision. This can be explained by the fact that the generated increment to the values of the changing parameters will not change in this case. Changing the solution changes only the initial values of the parameters being changed. In this case, you can use the well-known calculations hiding trick.

To achieve the most efficient use of computing devices, 3 special procedures were designed.

Procedure A1. You can determine the number and select the parameters to be changed and their increments in parallel on the processor, when the video card calculates the value of the quality functional. Under such conditions,

already calculated parameter increments are used to generate a new solution already at the next iteration, which reduces the number of sequentially performed calculations.

Procedure A2. For the subsequent optimization of the training algorithm, let us consider in more detail the stage of calculating the functional value.

Let current solution be equal to x , and new solution equal to y . During the calculation of the functional value on the video card for the solution y , the processor simultaneously generates two new solutions $x_1 \in N(x)$ and $y_1 \in N(y)$ where N — set of possible generated solutions from the current. After calculating the value of the functional, the necessity of transition to a new solution is checked. If a new decision is accepted, the next solution to be tested would be y_1 otherwise x_1 . This procedure allows you to mask the stage of generating a new solution. Thus, when moving to the next iteration, the functional value for the new solution will be immediately calculated without its explicit generation. However, with this approach, the amount of computation on the processor almost doubles, which can be critical for a small network.

In some cases, procedure 1 will be optimal in other cases, procedure 2, which was discussed in detail in [8].

Procedure A3. Its main idea is to most accurately estimate the speed of the procedures A1 and A2 when solving a specific applied problem. Since the execution of one iteration of the algorithm requires, as a rule, less than one thousandth of a second of time, and the processor cache appears after some time and the timer has a non-zero error, a large number of iterations should be used for an accurate estimate.

Step 1. Running the procedure A1 on ten thousand iterations with the measurement of the running time.

Step 2. Running the procedure A2 on ten thousand iterations with the measurement of the running time.

Step 3. Based on the results of the execution time, a training procedure is selected for the remaining iterations with a shorter running time.

When training neural networks using the annealing method using connected devices, the framework automatically evaluates the power of computing devices using the A3 procedure and selects the most efficient parallelization option.

Matrix multiplication on the video card is implemented using two-level parallelization. The first level cyclically distributes the calculations of individual blocks of the resulting matrix between the working groups of the video card. The second level cyclically distributes the calculations of individual elements of the block of the resulting matrix between the cores of a separate group. Since the resulting matrix contains a large number of blocks, there is no problem of irregular loading of workgroups. The number of elements in the block is a multiple of the number of cores in the group, so there

is no irregular cores loading.

Computing the value of the objective function requires assembling the result on the processor. The calculation of the objective function value is carried out in several stages. At the first stage, shared video memory is created for individual workgroups. At the second stage, each core of the video card calculates a fragment of the objective function value. At the third stage, one core in each working group summarizes the results of the calculations of the cores of the group into memory shared between the groups. Exactly one shared memory cell is allocated for each group. At the fourth stage, the video card transfers the contents of the shared memory to the processor upon completion of work. At the fifth stage, the processor summarizes the results of the groups, because on a video card, synchronization of calculations is possible only within the same working group. The number of working groups is chosen in such a way that there is not too much data transfer and long assembly, and at the same time there is no too long calculation of objective function fragments on the video card.

VIII. EXPERIMENTS

Let's check the developed software package efficiency using the example of solving the color image compression problem.

For the experiments, the STL-10 dataset from Stanford University was used [9].

The dataset contains 100,000 color images with a resolution of 96x96 pixels. The images can show an arbitrary object [10], which makes compressing image data quite a challenge.

For experiments, 8-fold, 16-fold and 32-fold compressions were chosen. Lower compression ratios are more efficiently produced using classical compression algorithms, and higher ones are meaningless due to too large losses.

For all degrees of compression, the images were divided into fragments of 4 by 4 pixels. Splitting into smaller fragments leads to a decrease in the compression quality, an increase, in turn, leads to an oversized neural network architecture and requires too much data and computing resources for training. Each individual fragment is compressed by a separate restricted Gauss-Bernoulli Boltzmann machine. For 8-fold compression, the number of neurons in the hidden layer of each machine was 48, for 16-fold - 24, for 32-fold - 24, but to achieve the required degree of compression, another layer of restricted Bernoulli-Bernoulli type Boltzmann machines was added with 48 neurons in the input layer and 24 in the hidden layer.

To train restricted Boltzmann machines with the gradient method, the CD-1 algorithm will be used. The PCD algorithm is more efficient on a small number of iterations [11], however, it is based on the assumption

captionCompression results 3 bit per pixel

Training algorithm	ADAM	FTML	genetic	annealing
MSE	272	254	322	262
PSNR	23.9	24.2	23.1	24.0
PSNR_HVS	24.1	24.4	23.3	24.1
SSIM	0.746	0.756	0.698	0.733
Training time, h	10	10	30	30

that at a separate iteration the parameters of the network being trained don't change significantly, which doesn't correspond to the problem being solved. The CD-k algorithm requires k times more calculations per iteration than CD-1 and at the same time achieves better quality [12], however, in the problem being solved, the values of the gradients are very large and the use of the CD-k algorithm is not advisable.

To compare the effectiveness of the developed learning algorithms in the framework with existing analogues, the strongest optimization algorithms implemented in analogues were taken: the adaptive moment method [13] (ADAM) due to its stability used for training neural networks of complex architecture [14] and following the moving leader method [15] (FTML), and from the current framework — the original annealing method and genetic algorithm.

For training, the first 8000 images were used as a training set, the next 7000 images for the validation set, the remaining 85000 images formed a test set.

To evaluate the compression efficiency, the most common quality functionals were chosen: MSE (mean squared error), PSNR (peak signal to noise ratio), PSNR-HVS (PSNR human visual system), SSIM (structure similarity image measurement).

The experiments were held on the operating system Ubuntu 20.04 with 4-core CPU intel i7-4770k, 16 GB 1600 MHz RAM and GPU nvidia rx 3070 with 5888 cores. Compiler version gcc 9.4.0, GPU driver version 470.161.03. The framework has been configured to use the GPU for training.

```
Compilation options "gcc -xc++ -Wl,-z,stack-size=1000000000 superOpenCLFramework.cpp constants.cpp trainPause.cpp deviceConvLayer.cpp devicePoolingLayer.cpp deviceDeepNN.cpp trainDCNN.cpp poolingLayer.cpp convLayer.cpp mlp.cpp autoencoder.cpp testCompression.cpp deviceMLP.cpp RBMGaussBernoulli.cpp deviceRBMGaussBernoulli.cpp deepNN.cpp trainDeepNN.cpp deepNNFunctioning.cpp dataProcessing.cpp deviceData.cpp superFrameworkInit.cpp finalTrainStats.cpp finetuning.cpp trainMLP.cpp trainRBM.cpp fastMath.cpp trainSettings.cpp -lstdc++ -D_FORCE_INLINES -O3 -l OpenCL -lgomp -lm -fopenmp"
```

The experiments results are displayed by data compression ratio (see Table 1, 2, 3).

From the experimental results, it can be noted that

captionCompression results 1.5 bit per pixel

Training algorithm	ADAM	FTML	genetic	annealing
MSE	433	397	452	390
PSNR	21.9	22.3	21.7	22.3
PSNR_HVS	22.1	22.5	21.9	22.5
SSIM	0.663	0.673	0.638	0.669
Training time, h	4	4	18	22

captionCompression results 0.75 bit per pixel

Training algorithm	ADAM	FTML	genetic	annealing
MSE	836	756	697	640
PSNR	19.0	19.4	19.8	20.2
PSNR_HVS	19.2	19.5	20.0	20.3
SSIM	0.502	0.509	0.525	0.551
Training time, h	6	6	21	25

at low degrees of compression, the annealing method is approximately equal to the gradient methods in terms of the quality of training, but it lags behind them in terms of speed by about 4 times. However, as the compression task becomes more complex, the annealing method begins to surpass them in quality. With 32-fold compression, both random search algorithms significantly outperformed the gradient methods in quality, and the annealing method significantly outperformed the genetic algorithm in the resulting solution quality.

The higher the complexity of the problem being solved, the worse the solution is obtained by gradients compared to random search algorithms. Modern training neural networks frameworks either do not support this type of training algorithms in principle, or their functionality in this part is extremely poor, which makes them not the most effective means of solving complex applied problems.

IX. CONCLUSION

The paper presents a software package for training neural networks using random search algorithms.

High performance and cross-platform is achieved through the use of OpenMP, OpenCL libraries. A wide supported architectures library in the framework allows us to construct deep neural networks of various architectures, which makes it flexible in applied problems solving. The framework supports a wide variety of computing devices and is able to adapt to low-power computing devices, which makes it possible to use it on a wide variety of devices.

Thanks to the random search algorithms support, the framework is able to solve complex applied problems more efficiently than those existing on gradient methods. Using the example of solving the color image compression problem, it was shown that the proposed framework solves neural networks complex training problem more efficiently than existing analogues. From this we can conclude that the developed software package can be

used instead of analogues and has a great development prospect in the future.

REFERENCES

- [1] A. Prieto, B. Prieto, E. M. Ortigosa, E. Ros, F. Pelayo, J. Ortega, and I. Rojas, "Neural networks: an overview of early research, current frameworks and new challenges," *Neurocomputing*, pp. 242–268, 2016.
- [2] M. Kovalev, A. Kroshchanka, and V. Golovko, "Convergence and integration of artificial neural networks with knowledge bases in next-generation intelligent computer systems," *Otkrytye semanticheskie tehnologii proektirovaniya intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, pp. 173–186, 2022.
- [3] A. Kroshchanka, "Deep neural networks application in next-generation intelligent computer systems," *Otkrytye semanticheskie tehnologii proektirovaniya intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, pp. 187–194, 2022.
- [4] M. Knop, T. Kapuscinski, W. K. Mleczo, and R. Angruk, "Neural video compression based on rbm scene change detection algorithm," *International Conference on Artificial Intelligence and Soft Computing*, pp. 660–669, 2016.
- [5] M. T. Khanna, C. Ralekar, A. Goel, S. Chaudhury, and B. Lall, "Memorability-based image compression," *IET Image Process*, pp. 1490–1501, 2019.
- [6] B. Sharma, M. Tomer, and K. Kriti, "Extractive text summarization using f-rbm," *Journal of statistics and management systems*, p. 1093–1104, 2020.
- [7] V. Krasnoproshin and V. Matskevich, "Random search in neural networks training," in *Proceedings of the 13-th International Conference "Computer Data Analysis and Modeling" – CDAM'2022*. Minsk : BSU, 2022, pp. 96–99.
- [8] V. Matskevich, "Obuchenie neironnykh setei na osnove metoda otzhiga [neural networks training based on annealing method]," *Vestnik Polotskogo gosudarstvennogo universiteta [Bulletin of Polotsk state university]*, p. 21–29, 2022.
- [9] Stl-10 dataset. Mode of access: academicorrents.com/details/a799a2845ac29a66c07cf74e2a2838b6c5698a6a. — Date of access: 25.02.2023.
- [10] Stl-10 dataset description. Mode of access: stanford.edu/~acoates/stl10. — Date of access: 24.02.2023.
- [11] K. Oswin, A. Fischer, and C. Igel, "Population-contrastive-divergence: Does consistency help with rbm training?" *Pattern Recognition Letters*, pp. 1–7, 2018.
- [12] K. Brugge, A. Fischer, and C. Igel, "The flip-the-state transition operator for restricted boltzmann machines," *Machine Learning*, p. 53–69, 2013.
- [13] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. of the 3rd Intern. Conf. on Learning Representations (ICLR 2015)*, 2015, p. 1–15.
- [14] S. Hamis, T. Zaharia, and O. Rousseau, "Image compression at very low bitrate based on deep learned super-resolution," *IEEE 23rd Intern. Symposium on Consumer Technologies (ISCT)*, p. 128–133, 2019.
- [15] S. Zheng and J. T. Kwok, "Follow the moving leader in deep learning," in *Proc. of the 34-th International Conference on Machine Learning*, 2017, p. 4110–4119.

Нейросетевая программная технология, обучаемая на принципах случайного поиска

Краснопрошин В. В., Мацкевич В. В.

В работе рассматривается актуальная прикладная проблема, связанная с программной реализацией нейросетевой технологии, в рамках которой процесс обучения основан на алгоритмах случайного поиска.

Обучение нейронных сетей является типичной задачей оптимизации. На начальном этапе развития нейросетевых технологий для решения таких задач традиционно использовались различные варианты градиентных методов. Такие методы, как правило, удовлетворяли требования к задаче по качеству и скорости обучения. Однако с появлением нового класса прикладных задач ситуация изменилась. Традиционный подход к обучению с использованием градиентных методов не всегда соответствовал требованиям прикладной задачи по качеству получаемого решения.

В работе предлагается один из вариантов программной реализации нейросетевой технологии (в виде фреймворка) по стандарту ostis 2021, в которой для обучения нейронных сетей используются алгоритмы случайного поиска.

Received 18.03.2023

Implementation of Information Needs of ostis-systems Users

Sergei Samodumkin
Belarusian State University of
Informatics and Radioelectronics
Minsk, Belarus
Email: Samodumkin@bsuir.by

Abstract—The possibilities of knowledge bases of intelligent systems allow you to represent and structure knowledge about the world around you and derive this knowledge about it, thereby satisfying the user’s information need. In this paper, the formal specification of the *Question language* for intelligent systems is clarified, which allows describing and interpreting any classes of questions of users of intelligent systems developed on the basis of OSTIS technology

Keywords—OSTIS, intelligent system, question language, information need

I. INTRODUCTION

One of the key features of an *intelligent system* is that the *user* has the opportunity to formulate his information need. One of the ways to express such a need is a *question* [1], [2]. In the process of dialog communication, there is always a context that defines additional information that contributes to the correct understanding of the *meaning* of the message. The peculiarity of the presentation of information in the *knowledge bases* of *ostis-systems* simplifies the formation of the user’s information needs, since the information presented in the *knowledge bases* is already structured and the relations set on a certain concept are known, in relation to which the question-problem situation is resolved. The paper [3] shows that the question-problem situation cannot be solved within the framework of formal logic and the nature of the issue can be understood in the system of subject-object relations. Due to the fact that when forming *knowledge bases* of *ostis-systems*, subject-object relations are formed within a given *subject area*, thereby simplifying the expression of information needs by the user by means of *SC-code* [4].

In order to identify specific types of relations, types of questions and classes of answers in papers [5], [6], the need for semantic classification of question-and-answer texts is justified. At the same time, the conceptual basis for the formalization of questions is the language of questions and erotetic logic [7], which allows you to ask question-answer relations.

Within the framework of this work, fragments of structured texts in SCn-code [8], [9] will often be used, which are simultaneously fragments of the source texts of the knowledge base, understandable to both

humans and machines. This allows you to make the text more structured and formalized, while maintaining its readability. The symbol ":= " in such texts indicates alternative (synonymous) names of the described entity, revealing in more detail certain of its properties.

II. THE PURPOSE OF DEVELOPING

The purpose of developing *Question language for ostis-systems* and its subsequent developing is to implement the possibility of understanding the actions carried out by the *ostis-system* when forming an answer to the *question* posed. In the process of forming a conclusion to the *question* posed, the following options are possible:

- the answer to this question exists in the *knowledge base* and a *fragment of the knowledge base* is localized in the context of the *user’s* information needs expressed by means of *SC-code*;
- the answer is related to the resolution of some problem situation, which is contained in the context of the *question* and the formation of the *answer to the question* is assigned to the *problem solver*.

Question language for ostis-systems

- := [Proposed version of the language for describing questions and answers to them in ostis-systems]
- ∈ *sc-language*
- ⇒ *syntax of language**:
Syntax of Question Language for ostis-systems
⊂ *SC-code syntax*
- ⇒ *denotational semantics of language**:
Denotational semantics of Question language for ostis-systems
- := [Ontology of classes of signs and relations for describing the formulations of questions in SC-code]
- ⊃ *Semantic classification of questions*
- ⇒ *operational semantics of language**:
Operational semantics of Question Language for ostis-systems
- := [Collective of sc-agents displaying answers to the questions asked by the ostis-system user]

III. SYNTAX OF QUESTION LANGUAGE FOR OSTIS-SYSTEMS

Question Language for ostis-systems belongs to the family of semantic compatible languages – *sc-languages* and is intended for the formal description of the search prescription of *ostis-systems* in order to meet the information needs of the *user*. Therefore, the ***syntax of the Question Language for ostis-systems***, like the *syntax* of any other *sc-language*, is the *Syntax of SC-Code*. This approach allows you to:

- unify the form of presentation of *questions* and *knowledge*, with the help of which answers to the *questions* posed are built;
- use a minimum of means to interpret the *questions asked by users*;
- reduce the output of answers to most of the *questions* asked to the search for information in the current state of the *ostis-system knowledge base*.

IV. DENOTATIONAL SEMANTICS OF QUESTION LANGUAGE FOR OSTIS-SYSTEMS

Denotational semantics of the Question Language for ostis-systems includes *classes of questions* and corresponding *classes of answers* necessary for the specification of the formulations of *questions* and *answers* to them, as well as *classes of signs* and *relations* included in the structure of any *question*. *Semantic classification of questions of the Question Language for ostis-systems* is based on the idea described in the paper [5]. Any ***question*** in *Question Language for ostis-systems* is a *specification of an action* to search for or generate *knowledge* that satisfies the information need of the *user* initiating this *question*. That is, the *question* — is nothing more than a *problem* by which the user's need for some information is expressed, possibly stored or output in the *knowledge base* of the *ostis-system*.

Each *question* can be uniquely correlated with a certain set of *answers* to this *question*. Each *answer to the question* represents a certain *sc-structure* of the *semantic neighborhood of the main sign* disclosed in this *answer* to the *question*.

question

- := [request]
- := [not a procedural formulation of the task of searching (in the current state of the knowledge base) or generating knowledge that meets the specified requirements]
- := [in what way]
- := [request for a method (method) for solving a given (specified) *class of problems* or a *plan for solving* a specific specified *problem*]
- := [problem aimed at satisfying the information needs of a certain customer entity]
- ⊂ *problem*

answer to the question

- := [response to the request]
- := [query result]
- := [result of solving the problem of finding or generating knowledge that meets the specified requirements]
- := [semantic neighborhood of the *main sign*, the knowledge of which satisfies the information need of the user]
- ⊂ *knowledge*

Among all classes of *signs within the framework of a given question* of the *Question Language for ostis-systems*, the most common classes of *signs* in the hierarchy can be distinguished:

sign within the framework of a given question

- ⊂ *sign*
- ⇒ *splitting**:
 - *main sign within the framework of the question asked*
 - := [key *sc-element* within the given question]
 - := [*sign* about which the question is asked]
 - *non-core sign within the framework of the question asked*
 - := [*sign* that stands in some relation to the *main sign* within the framework of the question asked]

sign within a given question is any *sign of a concept* or *entity* belonging to that *question*. Between the *signs, within the framework of the given question*, a set of relationships of *relations* that are part of various *subject areas* is set. In addition, ***any relation within the framework of a given question*** is a *relation* between the *signs* of the *subject area* belonging to the given *question*. Among all classes of *relations within the framework of a given question*, one can distinguish a class of ***basic relations within the framework of a given question*** and a class of ***composite relations within the framework of a given question***.

attitude within the framework of the question asked

- := [a certain relationship between the signs of the *subject area* in the context of the *question*]
- ⊂ *attitude*

the basic attitude within the framework of the question asked

- := [a *class of relations* that unites *relations* in a given *question*, reflecting the same type of *meaning* and revealing a certain feature of the *signs* of the *subject area*]

- ⊂ *attitude within the framework of the question asked*
- ⇒ *decomposition**:
- {• *state attitude*
 - *action attitude*
 - *composition attitude*
 - *set-theoretic attitude*
 - *temporal attitude*
 - *spacial attitude*
 - *quantitative attitude*
 - *qualitative attitude*
- }

For instance, *relations within the framework of a given question*, such as “plays*”, “sleeps*”, “swims*”, are combined into a *class of state relations* on the basis of expressing the state of the *sign* (that is, these relations reveal the feature *sign* of the *subject area* — “to be in some state”).

compound relation within the framework of the question asked

:= [a stable combination of two *action attitudes*: an action aimed at the *parameter of the question'*, and an action aimed at *answering the question**]

For instance, an element of a *composite relations within the framework of a given question* between the *signs*: “Oil refinery”, “oil” and “petroleum products” — can be represented as an “Oil refinery that processes oil into petroleum products”.

Semantic classification of *questions* makes it possible to contrast each type of question with a limited set of permissible, in other words, *semantically correct information structures* that convey the correct *meaning* of the *question* depending on the class of the *question*. At the same time, the ***semantic classification of questions*** allows you to divide a lot of *questions* into classes, each of which requires the disclosure of some of the same type of *meaning* given by the class of this *question*.

question

- ⇒ *decomposition**:
- {• *question requiring the derivation of the semantic neighborhood of the main sign*
 - ⊃ *example'*:
 - *Question. What is the city of Minsk*
 - *question that requires disclosure in the answer of the basic relation of the main sign*
 - ⊃ *example'*:
 - *Question. Which is lighter: iron or wood*
 - *question requiring disclosure of the*

composite relation of the main sign in the answer

- ⇒ *explanation**:
- [This class of *questions* corresponds to the classes of *answers* in which the *main sign* is revealed through a *composite relation*.]
- ⊃ *example'*:
- *Question. What are the principles of component design in intelligent computer systems of the new generation*
 - *question requiring disclosure in the answer of an arbitrary combination of the basic relation and/or the composite relation of the main sign*
 - ⊃ *example'*:
 - *Question. How is the intelligence level of a cybernetic system determined?*
 - *question that requires disclosure of more than one main sign in the answer*
 - ⊃ *example'*:
 - *Question. Prove the Pythagorean theorem*
- }

question requiring disclosure in the answer of the basic relation of the main sign

- ⇒ *decomposition**:
- {• *question requiring disclosure in the response of the composition attitude of the main sign*
 - := [a class of questions in the answers to which the *main sign S* is revealed through its *composition attitude* in conjunction with its constituent signs *P* and *Q*]
 - ⊃ *example'*:
 - *Question. Which administrative districts are part of the City of Vitebsk*
 - ⇒ *answer to the question**:
{Zheleznodorozhny district of Vitebsk, Oktyabrsky district of Vitebsk, Pervomaisky district of Vitebsk}
 - *question requiring disclosure in the answer set-theoretic relation main sign*
 - := [class of questions in the answers

- to which *the main sign* S is revealed through its *set-theoretic relation* in conjunction with another sign P containing S as part of]
- ⊃ *example'*:
- *Question. Which region is Smolevichi district a part of*
 ⇒ *answer to the question**:
 {Smolevichi district is part of the Minsk region}
- *question requiring disclosure in the answer state relations main sign*
 := [a class of questions in the answers to which *main sign* S is revealed through its *state relation*]
 ⊃ *example'*:
 • *Question. Which cities of the modern territory of the Republic of Belarus had Magdeburg law*
 ⇒ *answer to the question**:
 {Volkovysk, Grodno, Mozyr and others had Magdeburg law}
- *question requiring disclosure in response action relationship main sign*
 := [a class of questions in the answers to which *the main sign* S is revealed through its *action relation* in conjunction with another sign P]
 ⊃ *example'*:
 • *Question. Which event happened earlier: The First Section of the Polish-Lithuanian Commonwealth or the Battle of Borodino*
 ⇒ *answer to the question**:
 {The first section of the Polish-Lithuanian Commonwealth was before the Battle of Borodino}
- *question requiring disclosure in the answer spatial relationship main sign*
 := [class of questions, in the answers to which *main sign* S is revealed through *spatial relation*, reflecting its position in space relative to another sign P]
 ⊃ *example'*:
 • *Question. What is the height of Dzerzhinskaya Mountain*
- *question requiring disclosure in the answer quantitative relation main sign*
 := [class of questions in the answers to which *quantitative relation* is revealed *main sign*]
 ⊃ *example'*:
 • *Question. The territory of which administrative region is larger: Minsk or Brest*
- }
question requiring disclosure in the answer of an arbitrary combination of base relation and/or compound relation main sign
 ⇒ *decomposition**:
 {• *question requiring disclosure in the answer of an arbitrary combination of composite description relation main sign*
 := [class of questions whose answers reveal arbitrary combinations of *basic relationship* and/or *composite relationship of the main sign* S in conjunction with other signs]
 ⊃ *example'*:
 • { S consists of P , Q , W . S translates X and Y and runs before Z }
 ⇐ *answer to question**:
 Question. What is S
- *question requiring disclosure in the answer of an arbitrary combination of compound definition relation main sign*
 - *question requiring disclosure in the*

answer of an arbitrary combination of compound definition relation main sign

⊃ example':

- {Minsk is the capital, which is located in the Republic of Belarus}

⇐ answer to question*:

Question. How is the city of Minsk defined

- question requiring disclosure in the answer of an arbitrary combination of a compound relation of the reason main sign

:= [class of questions, the answers to which reveal the condition for the existence of some relations the main sign *S* in conjunction with other signs]

⊃ example':

- Question. Why is the travel time from Minsk to Borisov less than the travel time from Minsk to Orsha

- question requiring disclosure in the answer of an arbitrary combination composite relation of the consequence main sign

:= [class of questions, the answers to which reveal the consequence of the existence of some relations the main sign *S* in conjunction with other sign]

⊃ example':

- Question. What follows from the fact that the distance from the city of Minsk to the city of Borisov is less than the distance from the city of Minsk to the city of Orsha

}

question requiring disclosure of more than one main sign in the answer

⊃ question requiring disclosure in the answer detail ratio of signs standing in some relationship with main sign

:= [class of questions in the answers to which there is a detailing of signs standing in some relationship with the main sign *S*]

⊃ example':

- Question. What is the connection in the same network between the city of Minsk and the city of

Svetlogorsk

⇒ answer to the question*:

{The city of Minsk is located on the Svisloch River, which flows into the Berezina River, which flows through the city of Svetlogorsk}

Thus, for each question ostis system user, you can find a class questions on which you can implement output of answers to this question. Described by Semantic classification of questions allows:

- automatically structure questions users according to the description of these questions;
- and also form answers to these questions taking into account non-procedural formulations of these questions.

V. OPERATIONAL SEMANTICS OF QUESTION LANGUAGE FOR OSTIS-SYSTEMS

Each class of questions must correspond to a certain team of sc-agents implementing the output (search or generation) from knowledge base ostis-system of the corresponding answers to the questions. It should be noted that depending on the degree of fullness of knowledge base answers may be contained in knowledge base or absent in the current version of knowledge base. If there is in knowledge base an answer to the question posed by question, the user's information need is implemented by information retrieval sc-agents [10], otherwise — depending on classes of questions, the implementation of the output of answers is carried out by specialized sc-agents, which in the process of work additionally perform computational tasks or perform synthesis based on logical inference.

All sc-agents that output answers to posed questions form team of sc-agents — **interpreter of the Question Language for ostis-systems**, with which you can interpret any classes questions. a Question Language interpreter for ostis-systems can be implemented in different ways: in the form of a team of scp-agents or platform-dependent sc-agents.

CONCLUSION

Let's list the main provisions of this work:

- the information need of ostis-system users can be expressed in the form of questions, and the satisfaction of this information need — in the form of answers to the questions asked;
- the conclusion of answers to the questions by the user of the ostis-system can be carried out by searching for knowledge in the current state of knowledge base of this ostis-system, or by generating new knowledge that is missing in knowledge base of this ostis-system;

- each *question* can be presented in the form of some *task specification* initiated by *ostis-system user* to meet their information needs, and *answer to this question* — in the form of *semantic neighborhood the main sign within the given question*;
- for each *question*, the corresponding class of *questions* can be found in *Semantic classification of questions*.

ACKNOWLEDGMENT

The author thanks the scientific staff of the Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics for their help in the work and valuable comments.

REFERENCES

- [1] S. Samodumkin, “Technology of intelligent questions-and-answers systems,” 2009.
- [2] —, “Semantic technology of intelligent questions-and-answers systems,” *Reports of the Belarusian State University of Informatics and Radioelectronics*, no. 7 (45), pp. 67–72, 2009.
- [3] L. Averyanov, “Pochemu lyudi zadayut voprosy[why do people ask questions],” *M.: Sociolog*, p. 152, 1993.
- [4] V. Golenkov, N. Gulyakina, and D. Shunkevich, *Otkrytaya tekhnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheski sovmestnykh gibridnykh intellektual’nykh komp’yuternykh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint, 2021.
- [5] J. S. Suleymanov, “Issledovanie bazovykh principov postroeniya semanticheskogo interpretatora voprosno-otvetnykh tekstov na estestvennom yazyke v aos[study of the basic principles of constructing a semantic interpreter of question-answer texts in natural language in aos],” *Obrazovatel’nye tekhnologii i obshchestvo[Educational technologies and society]*, vol. 4, no. 3, pp. 178–192, 2001.
- [6] —, “Sistema semanticheskogo analiza otvetnykh tekstov obuchaemogo na estestvennom yazyke[the system of semantic analysis of student response texts in natural language],” *Ontologiya proektirovaniya[Design ontology]*, vol. 1, no. 1, pp. 65–77, 2014.
- [7] N. Belnap *et al.*, “Logika voprosov i otvetov[logic question and answer],” 1981.
- [8] (2023, Feb) IMS.ostis Metasystem. [Online]. Available: <https://ims.ostis.net>
- [9] V. Golenkov, N. Guliakina, I. Davydenko, and A. Ereemeev, “Methods and tools for ensuring compatibility of computer systems,” in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2019, pp. 25–52.
- [10] D. Shunkevich, “Ontology-based design of hybrid problem solvers,” *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh sistem [Open semantic technologies for intelligent systems]*, pp. 101–131, 2022.

Реализация информационной потребности пользователей ostis-систем

Самодумкин С. А.

Возможности баз знаний интеллектуальных систем позволяют представлять и структурировать знания об окружающем мире и выводить эти знания о нём, тем самым удовлетворяется информационная потребность пользователя. В данной работе уточнена формальная спецификация *Языка вопросов* для интеллектуальных систем, позволяющая описывать и интерпретировать любые классы *вопросов пользователей интеллектуальных систем*, разработанных на основе Технологии OSTIS.

Received 13.03.2023

Construction of Intelligent Systems Based on Knowledge Discovery in Datasets

Viktor Krasnoproshin
*Faculty of Applied Mathematics
and Computer Science
Belarussian State University
Minsk, Belarus
Email: krasnoproshin@bsu.by*

Vadim Rodchenko and Anna Karkanitsa
*Faculty of Mathematics and Informatics
Yanka Kupala State University of Grodno
Grodno, Belarus
Email: rovar@grsu.by, a.karkanica@grsu.by*

Abstract—The original method of intelligent systems construction based on technology of knowledge discovery in databases is considered. To form the knowledge base of an intelligent system, it is proposed to abandon the classical approach based on the formalization of expert knowledge in favor of an alternative approach aimed at identifying interpretable empirical patterns using Data Mining methods.

Keywords—Artificial intelligence, knowledge base, data mining, precedent-based learning

I. INTRODUCTION

The contemporary development of information systems is closely related to artificial intelligence technologies [1], [2]. Therefore, the development of methods and technologies aimed at construction of Intelligent Systems (IS) and their components is one of the key challenges of computer science [3], [4].

An intelligent system is a technical (or software) system capable of solving creative problems in a particular subject domain (SD). At the same time, knowledge about the SD is stored in the memory of the system itself. The IS structure traditionally consists of three major subsystems - a knowledge base (KB), a decision-making mechanism, and an intelligent interface [5], [6].

The knowledge base is the central component of an intelligent system. It contains structured information about the subject domain as a set of facts and rules. The study of models and methods related to the extraction, structuring and representation of knowledge is dealt with by one of the sections of computer science called knowledge engineering [7].

Deductive and inductive learning methods are used to build knowledge base. Deductive methods provide for the formalization of expert knowledge for the purpose of their further placement in the knowledge base. An example of intelligent systems built on the principles of deduction are expert systems (ES) [8], [9]. Inductive methods are based, as a rule, on the principles of learning from precedents. These methods are aimed at identifying

empirical patterns in data and are currently associated with machine learning and data mining [10].

The result of machine learning is an algorithm that approximates the unknown target dependency both on the objects of the training set and on the entire initial set [11], [12]. Data mining methods provide for the detection of previously unknown (practically useful and accessible for interpretation) patterns for decision-making in various fields of human activity [13], [14].

Expert systems (since their appearance in the mid-60s of the twentieth century) have demonstrated their effectiveness in such areas of human activity as medicine, chemistry, transport logistics, nuclear engineering, etc. However, today the concept of expert systems is in a serious crisis. The traditional approach to ES construction, unfortunately, does not fit well with the data models of the classical database theory. This makes it impossible to effectively use modern industrial DBMS to form knowledge bases of ES [15].

Currently, the dominant position in the development and construction of intelligent systems is occupied by machine learning and artificial neural networks. At the same time, the task of high-quality learning from precedents is singled out as a central problem [16]. Traditionally, this problem is reduced to solving an optimization problem: it is required to build an algorithm that would best approximate the unknown target dependency, both on the elements of the training set and on the entire set.

The paper proposes a new approach to building intelligent systems based on the technology of knowledge discovery in databases and the original implementation of the data mining stage.

II. STRATEGIES FOR INTELLIGENT SYSTEMS CONSTRUCTION BASED ON DEDUCTIVE AND INDUCTIVE LEARNING METHODS

An intelligent system is a technical (or software) system capable of solving creative problems in a specific subject domain based on knowledge. Knowledge is a set

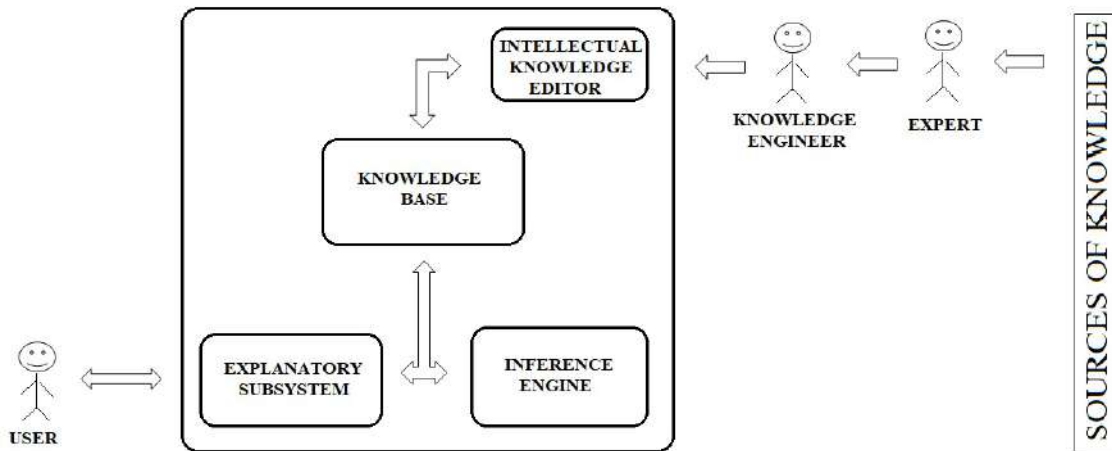


Figure 1. Expert System.

of facts, patterns and heuristic rules necessary to solve a given problem.

To build intelligent systems, two major strategies based on the principles of deduction or induction are used (Fig. 2).

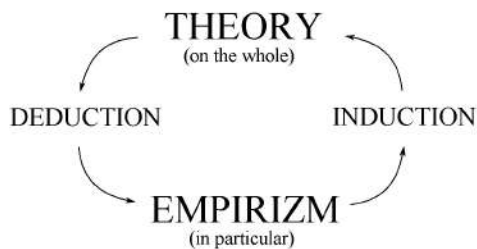


Figure 2. Deduction/Induction process.

Classical representatives of IS built on the basis of deduction are expert systems. The history of their application has more than fifty years. ES is an intelligent system built on knowledge about the subject domain received from experts or from other information sources [17], [18].

Traditionally, the architecture of an expert system consists of three main components: a knowledge base, an inference engine, and an intelligent interface.

The central component of ES is a knowledge base which contains a set of facts and rules necessary to solve the problem. The knowledge base is filled by an expert and a system analyst (knowledge engineer). The expert determines the knowledge base composition, ensures the completeness and correctness of the entered knowledge. The analyst helps the expert to identify and structure knowledge, choose an adequate model for knowledge

representation and an effective inference mechanism (Fig. 1).

At present, a separate direction of artificial intelligence has been formed - knowledge engineering. This direction is directly related to the theoretical and practical problems of designing and developing knowledge bases. Knowledge engineering also covers the issues of extracting (or acquiring) knowledge, its structuring and formalization [19].

The process of extracting knowledge is a time and load-consuming procedure. As a result of its execution, the analyst should build an adequate model of the subject domain that experts use for decision-making.

The inference mechanism implements a generalized procedure for searching the solution of the problem. In accordance with the user's need and on the basis of the knowledge base, a chain of reasoning is built, leading to a specific outcome.

The intelligent interface of the system provides a dialogue of the analyst and the user with the expert system. Using the knowledge subsystem, the analyst creates and edits the knowledge base. The user, through the subsystem of explanations, can ask the expert system questions and receive answers from it, can form and analyze chains of reasoning.

The accumulated experience of expert system developers allows us to state that the process of knowledge extraction remains the "bottleneck" in the construction of applied ES. The need for interaction between an expert and a data analyst excludes the possibility of implementing an automatic mode of knowledge base formation and reduces the objectivity of knowledge.

Currently, most intelligent systems are built on the basis of inductive learning methods (learning from precedents) [20], [21].

At the moment, there is a situation where learning from precedents is implemented on the basis of methods and technologies of machine learning and artificial neural networks (Fig. 3). The knowledge base of the intelligent system is formed as a result of the joint work of data analyst and ML-engineer.

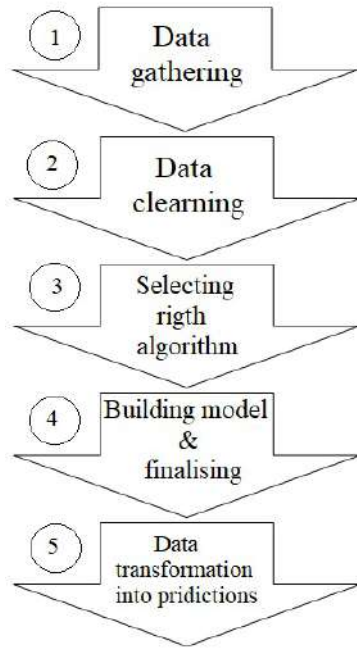


Figure 3. Machine Learning Process.

The work of data analyst begins with the identification and formulation of relevant applied problems in the subject domain. Then data analyst collects information and formulates hypotheses in order to improve certain indicators. And, further, it prepares data for analysis in the form of a training set.

ML-engineer is an expert in the practical application of neural network technologies and machine learning. The task of ML-engineer is to select a model of the problem, select its parameters, select a training method, and, finally, to qualitatively perform the process of model training. Since all this is carried out by the ML-engineer, learning can only be implemented in an automated but not automatic mode. The only useful outcome of learning is the classification algorithm, which is a «black box» and it's working mechanism cannot be interpreted. Thus, having spent serious resources on the formation of a training set, as a result, it is only possible to build a classification algorithm, but in no way to expand knowledge about properties of classes and subject domain.

Building intelligent systems (based on inductive methods of learning from precedents) it is proposed to use an alternative approach that allows to opt out of traditional method of learning (as part of solving the classification

problem) and which is based on using the idea of the compactness hypothesis.

III. ABOUT AN ALTERNATIVE METHOD OF LEARNING FROM PRECEDENTS

Learning from precedents is based on the identification of empirical patterns in data, and in practice it is performed as part of solving the classification problem [22]. The classification problem has the following statement:

Let X be the set of objects descriptions, Y — the set of numbers (names) of classes. Suppose there is an unknown target dependency $y^ : X \rightarrow Y$, which values are known only on the objects of the final training set $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$. It is necessary to construct an algorithm $a : X \rightarrow Y$, that would approximate this target dependency for any object from the initial set X .*

At present, the generally accepted scenario for solving the problem can be described as follows:

1. Some model of algorithms $A = \{a : X \rightarrow Y\}$ is selected.

2. A loss function $L(y, y')$ is introduced to measure the deviation of the algorithm ($y = a(x)$) for an arbitrary $x \in X$ from the correct value ($y' = y^*(x)$).

3. A quality functional $Q(a, X^m)$ is introduced as a value of the average error of the algorithm a on objects of the training set X^m .

4. Within model A , an algorithm that ensures the minimum value of the mean error on the entire sample X^m is constructed.

At least two serious shortcomings of this scenario should be pointed out. Firstly, the choice of an algorithm model, a loss function, and a quality functional is a non-trivial task for the ML-engineer. Secondly, the result of solving the problem is only a classification algorithm, which is actually a «black box» (Fig. 4).

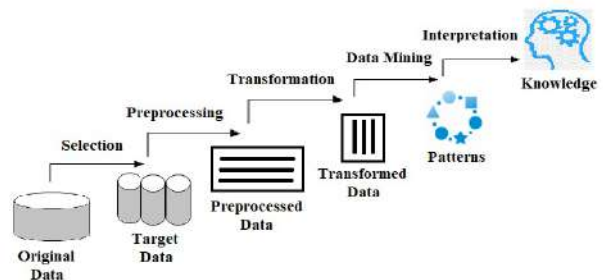


Figure 4. Knowledge Discovery Process.

It is proposed to implement the process of learning from precedents in an alternative way using the idea of the compactness hypothesis.

Learning will be carried out as part of solving the knowledge discovery problem (Fig. 5). The statement of this problem is as follows:

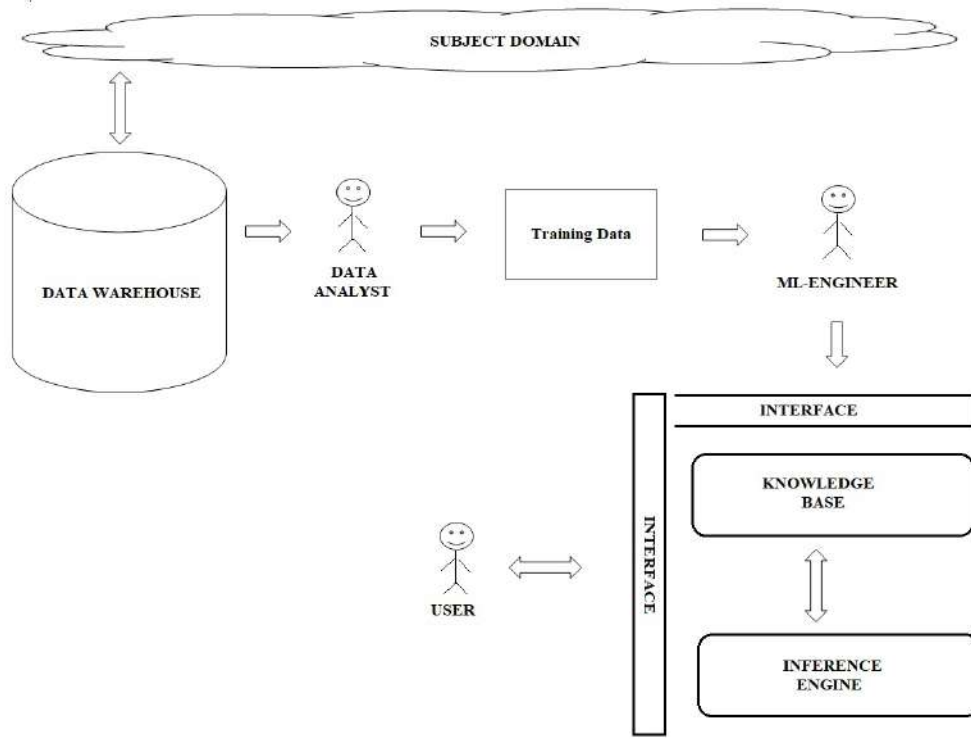


Figure 5. Intelligent System based on Machine Learning.

Let the objects descriptions X , an a priori dictionary of features $F = \{f_1, \dots, f_n\}$ and classes alphabet $Y = \{y_1, \dots, y_k\}$ are given. And let the training set $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$ be formed based on the dictionary F . The set $V = \{v_1, \dots, v_q\}$, where $q = 2^n - 1$, is the set of possible combinations of features from F . It is required to find such combinations of features from V for which class patterns do not intersect in the corresponding feature subspace.

The algorithm of searching combinations of features from $V = \{v_1, \dots, v_q\}$ for which class patterns do not intersect in the corresponding feature subspace, is as follows:

Step 1. Select from V a subset $V^+ = \{v_1^+, \dots, v_n^+\}$, where v_i^+ contains only one feature.

Step 2. For each v_i^+ , class patterns are built and their mutual placement is estimated [23].

Step 3. If the patterns do not intersect, then v_i^+ is included in the resulting set $V^* = \{v_1^*, \dots, v_k^*\}$.

Step 4. The subset $V^+ = \{v_1^+, \dots, v_n^+\}$ is excluded from the set $V = \{v_1, \dots, v_q\}$ and set $V^\Delta = \{v_1^\Delta, \dots, v_p^\Delta\}$ is obtained.

Step 5. All combinations v_i^Δ that contain any combination of $V^* = \{v_1^*, \dots, v_k^*\}$ are excluded from V^Δ .

Step 6. The next combination v_i^Δ is selected from V^Δ and on its basis a feature space is build.

Step 7. In this feature space, class patterns are build and their mutual placement is estimated.

Step 8. If the class patterns do not intersect, then the combination v_i^Δ is included in the resulting set V^* , and all combinations containing v_i^Δ are excluded from V^Δ .

Step 9. Steps 6-8 are repeated until V^Δ becomes empty.

The result of the algorithm execution will be the set $V^* = \{v_1^*, \dots, v_t^*\}$, where $0 \leq t \leq q$ and each combination $v_i^* \in V^*$ determines the regularity: «in the space of combination of features v_i^* classes do not intersect».

Since each combination of features from the set V^* defines a decision space where class patterns do not intersect, the construction of classification algorithms does not cause difficulties.

IV. INTELLIGENT SYSTEMS BASED ON KNOWLEDGE DISCOVERY IN DATASETS AND OSTIS TECHNOLOGY

An intelligent system built on the basis of the learning method proposed above differs from an intelligent system built on the basis of machine learning in that for formation of a knowledge base an ML-engineer is not required. In addition, the user is provided with the opportunity to analyze the chain of reasoning through the subsystem of explanations. The architecture of an intelligent system based on knowledge discovery in datasets (KDD) is shown in Fig. 6.

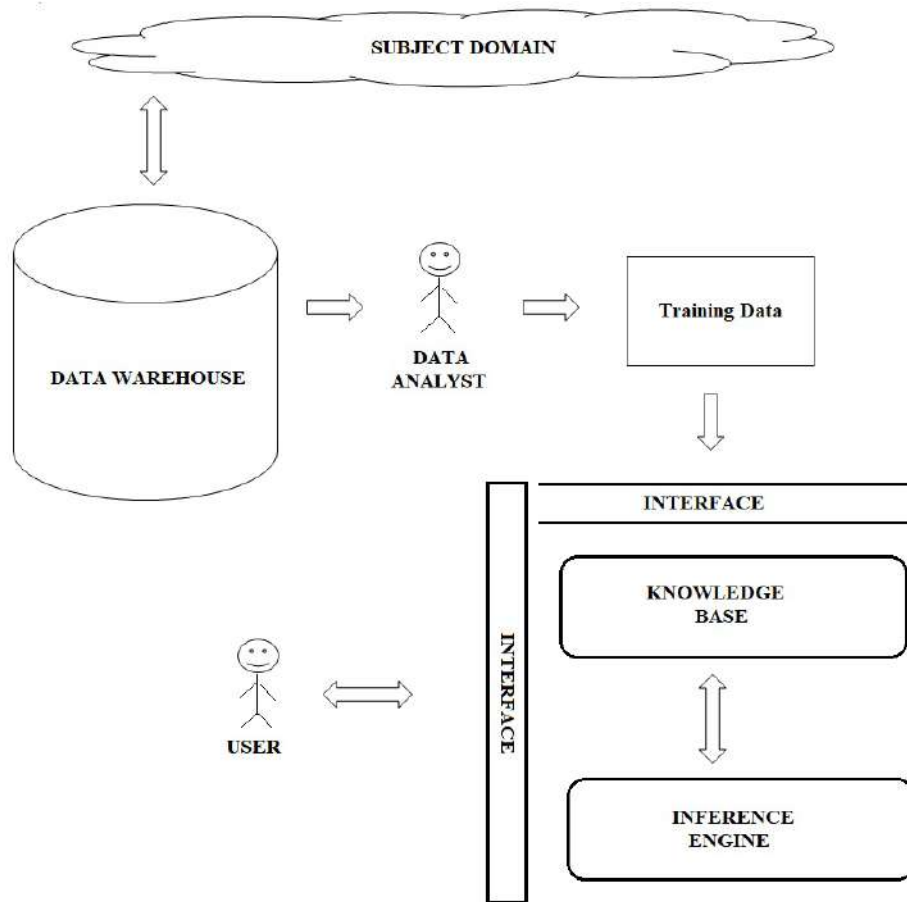


Figure 6. Intelligent System based on Knowledge Discovery.

Data analyst should have a profound knowledge and deeply understanding of processes occurring in the subject domain, be able to correctly set the problem, to collect, process, study and interpret data and, finally, to translate research results into the subject domain language (to make effective decisions). In fact, one of the most important functions of data analyst is the construction of a formal domain ontology.

To design an IS (based on KDD) and represent a formal ontology of the subject domain, it is proposed to use the technology named OSTIS (*Open Semantic Technology for Intelligent Systems*). OSTIS is a comprehensive technology for component-based design of IS [24].

The work of data analyst in forming a training set begins with the construction of an alphabet of classes and an a priori dictionary of features. The usage of OSTIS technology allows to implement a semantic representation of classes, to formalize representation of an a priori dictionary of features and build a training set model using the SC-code (sc-model).

The result of solving the knowledge discovery problem is a set of feature combinations $V^* = \{v_1^*, \dots, v_t^*\}$. Each

combination $v_i^* \in V^*$ defines a pattern of the form «*in the space of combination of features v_i^* classes do not intersect*» and is formally described in the KB using a unified, universal representation language – SC-code.

V. CONCLUSION

The paper proposes an original method of intelligent system construction based on knowledge discovery. The formation of the IS knowledge base is carried out on the basis of the analysis of training set data. Instead of the generally accepted goal-setting on the construction of a classification algorithm, it is proposed to place an emphasis in the learning process on the study of the classes properties that provide classes distinction.

The original knowledge discovery algorithm is developed which allows to identify automatically combinations of features that provide classes distinction using the features of the a priori dictionary and data of the training set.

It is shown that as a result, it is possible to detect previously unknown, interpreted within the subject domain and practically useful empirical patterns. This knowledge

can be effectively used to build the knowledge base of an intelligent system.

REFERENCES

- [1] Yu. Nechaev, A. Degtyarev *Intellektualnyje sistemy: kontseptsii i prilozheniya* [Intelligent systems: concepts and applications], Sankt-Peterburg (Saint Petersburg), SPbGU, 2011, P. 269
- [2] S. Russel, P. Norving *Iskusstvennyj intellekt: sovremennyy podkhod* [Artificial intelligence: a modern approach], Moskva (Moscow), Izdatel'skii dom Vil'yams, 2006, P. 1408
- [3] V. Krasnoproshin, V. Rodchenko, A. Karkanitsa *Spetsialized KD-agent for knowledge ecosystems. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system* [Open semantic technologies for intelligent systems], 2021, pp. 59–62.
- [4] P. Flach *Mashinnoe obuchenie. Nauka i iskusstvo postroeniya algoritmov, kotorye izvlekayut znaniya iz dannykh* [Machine learning. Science and art of constructing algorithms that extract knowledge from data], Moskva (Moscow), DMK Press, 2015, P. 400.
- [5] G. Luger *Iskusstvennyj intellekt: strategii i metody resheniya slozhnykh problem* [Artificial intelligence: structures and strategies for complex problem solving], Moskva (Moscow), Izdatel'skii dom Vil'yams, 2005, P. 864
- [6] A. Osrtoukh *Intellektualnye sistemy: monografiya* [Intelligent systems: monograph], Krasnoyarsk (Krasnoyarsk), Nauchno-innovatsionnyi tsentr, 2020, P. 316
- [7] T. Gavrilo, D. Kudryavtsev, D. Muromtsev *Ingenierya znaniy. Modeli i metody* [Knowledge engineering. Models and methods.], Sankt-Peterburg (Saint Petersburg), Izdatel'skii dom Lan, 2016, P. 324
- [8] J. Giarratano, G. Riley *Ekspertnye sistemy: printsipy razrabotki i programmirovaniya* [Expert systems: principles and programming], Moskva (Moscow), Izdatel'skii dom Vil'yams, 2007, P. 1152
- [9] C. Naylor *Kak postroit' svoju ekspertnuju sistemu* [Build your own expert system], Moskva (Moscow), Energoatomizdat, 1991, P. 286
- [10] *Deductive reasoning*. Available at: https://en.wikipedia.org/wiki/Deductive_reasoning/ (accessed 2023, Feb)
- [11] V. Krasnoproshin, V. Rodchenko *Obuchenie po pretsedentam na osnove analiza svoystv priznakov* [Learning by precedents based on the analysis of the features], *Doklady BGUIR*, 2017, No. 6, pp. 35–41.
- [12] V. Rodchenko *Pattern Recognition: Supervised Learning on the Bases of Cluster Structures, Pattern Recognition and Information Processing. PRIP 2016. Communications in Computer and Information Science*, Springer, 2017, vol. 673, pp. 106–113.
- [13] *Data mining*. Available at: https://en.wikipedia.org/wiki/Data_mining (accessed 2023, March)
- [14] V. Krasnoproshin, V. Rodchenko, A. Karkanitsa *Automatic construction of classifiers by knowledge ecosystem agents. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system* [Open semantic technologies for intelligent systems], 2022, pp. 195–198.
- [15] V. Ruchkin, V. Fulin *Universalnuy iskusstvennuy intellekt i ekspertnue sistemu* [Universal artificial intelligence and expert systems], Sankt-Peterburg (Saint Petersburg), BHV-Peterburg, 2009, P. 240.
- [16] V. Krasnoproshin, V. Rodchenko *Klassifikatsiya na osnove prostanstv reshenij* [Classification based on decision spaces], *Doklady BGUIR*, 2019, No. 6, pp. 20–25.
- [17] P. Jackson *Vvedenie v ekspertnyje sistemu* [Introduction to expert systems], Moskva (Moscow), Izdatel'skii dom Vil'yams, 2001, P. 624.
- [18] L. Rutkovskiy *Metody i tekhnologii iskusstvennogo intellekta* [Methods and technologies of artificial intelligence], Moskva (Moscow), Telekom, 2010, P. 520.
- [19] T. Gavrilo, V. Khoroshevskiy *Bazy znaniy intellektualnykh sistem* [Knowledge bases of intelligent systems], Sankt-Peterburg (Saint Petersburg), Izdatel'skii dom Piter, 2000, P. 384.
- [20] S. Raschka *Python i mashinnoe obuchenie* [Python machine learning], Moskva (Moscow), DMK Press, 2017, P. 418.
- [21] *Artificial neural network*. Available at: https://en.wikipedia.org/wiki/Artificial_neural_network/ (accessed 2023, March)
- [22] *Machine learning*. Available at: https://en.wikipedia.org/wiki/Machine_learning/ (accessed 2023, March)
- [23] V. Krasnoproshin, V. Rodchenko *Klasternye struktury i ikh primeneniye v intellektualnom analize dannykh* [Cluster structures and their applications in data mining], *Informatika* [Informatics], 2016, No. 2, pp. 71–77.
- [24] V. Golenkov, N. Guliakina, I. Davydenko, D. Shunkevich, A. Eremeev *Ontologicheskoe proektirovaniye gibridnykh semanticheskikh system na osnove smyslovogo predstavleniya znaniy* [Ontological design of hybrid semantically compatible intelligent systems based on sense representation of knowledge]. *Ontologiya proektirovaniya* [Ontology of designing], 2019, No. 1, pp. 132–151.

Построение интеллектуальных систем на основе Knowledge Discovery in Datasets

Краснопрошин В. В., Родченко В. Г.,
Карканица А. В.

Рассмотрен оригинальный способ построения интеллектуальных систем на основе технологии knowledge discovery in databases. Для формирования базы знаний интеллектуальной системы предложено отказаться от классического подхода, основанного на формализации знаний экспертов, в пользу альтернативного, направленного на выявление методами Data Mining интерпретируемых эмпирических закономерностей.

Received 20.03.2023

User Interface of the OSTIS Ecosystem

Mikhail Sadouski

*Belarusian State University of
Informatics and Radioelectronics*

Minsk, Belarus

Email: sadovski@bsuir.by

Abstract—In the article, the principles of organizing user interaction with the OSTIS Ecosystem, the concept of the user interface of the OSTIS Ecosystem, the use of component approach to the design of adaptive intelligent multimodal interfaces of ostis-systems are considered.

Keywords—adaptive intelligent multimodal interface, OSTIS Ecosystem, user interface of the OSTIS Ecosystem

I. INTRODUCTION

Modern information technologies have led to the emergence of a huge number of services and systems designed to solve a variety of problems. However, due to the variety of these services, the user faced the problem of the need to study the features of interaction with each of them, which can significantly complicate the use of such services. In addition, each system has its own unique *user interface*, which may be incomprehensible and require certain interaction skills from the user. Thus, the user has to independently study the features of interaction with each system in order to successfully solve their problems.

In this regard, there is an urgent problem of usability of information services, which requires the development of approaches to the unification of *user interfaces* and the creation of tools that simplify working with various systems. The development of such approaches can significantly speed up the process of user interaction with systems and increase the efficiency of their use.

The purpose of this article is to analyze existing solutions to this problem and an approach to eliminating this problem using the *OSTIS Technology*.

II. RELATED WORKS

Currently, in order to solve the problem of usability of various information services and unification of *user interfaces*, the following is used:

- *personal assistants*, which simplify the processes of user interaction with various systems;
- standards of protocols and *application interfaces* that simplify the integration and interaction of various systems with each other.

Digital *personal assistants* are programs based on artificial intelligence and machine learning technologies that help users perform everyday tasks, such as scheduling, managing contacts, searching for information, reminding about important events, etc. [1], [2], [3], [4].

Nowadays, many companies try to develop their own *personal assistant*, and the first company that managed to integrate such an agent into their operating systems was Apple when they introduced Siri [5] in 2010. Soon after, many other companies implemented assistance in the same area in order to help people perform ordinary everyday actions (Microsoft Cortana [6], Google Now [7], LG Voice Mate [8], [9]).

The *user interface* of the *personal assistant* must represent the system as a single unified set of back-end task assistants, enabling the user to conduct a dialog in which it is easy to switch between these domains. It involves getting user input commands either as text or speech and processing the input, which is in natural language, to understand it [10].

Using the *personal assistant* to interact with the user does not solve the problem of compatibility of various services and systems with each other. The use of standard protocols and *application interfaces* partially solves this problem, but the fully specified problem can only be solved by creating the *ecosystem* of semantically compatible systems.

As part of the work [9], it is proposed to use the LISSA *personal assistant* that is operating in a Virtual Education Space (VES) and whose purpose is to aid the students in their learning process. VES is built as an Internet-of-Things *ecosystem* consisting of autonomous intelligent components displaying a context-aware behaviour. This approach makes it possible to effectively use a *personal assistant* within the specified *ecosystem* with the possibility of further expansion of its application.

III. PROPOSED APPROACH

Based on the analysis, it can be concluded that the user does not have to know a lot of services from which they should choose the functionality that suits them. The complex of semantically compatible services should be located “behind the scenes”. Therefore, all information resources and services used must be semantically compatible. The choice of a resource or service suitable for the user should be made by their *personal assistant*.

Thus, when implementing digital *personal assistants*, it is necessary to ensure their scalability and adaptability to user needs. This means that the system should be able

to automatically adapt to changes in user behavior, taking into account their preferences, work characteristics, and other factors.

The *OSTIS Technology* allows creating semantically compatible systems (*ostis-systems*) that are able to process user requests and problems, taking into account their context and meaning. This is achieved through the use of semantic networks that allow describing knowledge and the connections between them. Also, the *OSTIS Technology* provides scalability and flexibility of the system, which allows it to adapt to changes in user behavior and needs [11].

The *OSTIS Technology* provides a universal language for the semantic representation (encoding) of information in the memory of *intelligent computer systems*, called an *SC-code*. Texts of the *SC-code* (*sc-texts*) are unified semantic networks with a basic set-theoretic interpretation, which allows solving the problem of compatibility of various knowledge types. The elements of such semantic networks are called *sc-elements* (*sc-nodes* and *sc-connectors*, which, in turn, depending on orientation, can be *sc-arcs* or *sc-edges*). The *Alphabet of the SC-code* consists of five main elements, on the basis of which *SC-code* constructions of any complexity are built, including more specific types of *sc-elements* (for example, new concepts). The memory that stores *SC-code* constructions is called semantic memory, or *sc-memory* [12].

The architecture of each *ostis-system* includes a platform for interpreting semantic models of *ostis-systems*, as well as a *semantic model of the ostis-system* described using the *SC-code* (*sc-model of the ostis-system*). In turn, the *sc-model of the ostis-system* includes the *sc-model of the knowledge base*, *sc-model of the interface*, and *sc-model of the problem solver*. The principles of the design and structure of *knowledge bases* and *problem solvers* are discussed in more detail in [13] and [14], respectively. The principles of the *sc-model of the user interface* were described in the articles [15], [16], and [17], on which this article is based.

Within the *OSTIS Technology*, the concept of the *OSTIS Ecosystem* is introduced [18].

The *OSTIS Ecosystem* is a socio-technical network, which is a collective of interacting:

- *ostis-systems* themselves;
- users of the specified *ostis-systems* (both end-users and developers);
- some *computer systems* that are not *ostis-systems* (they can be used as additional information resources or services).

The objectives of the *OSTIS Ecosystem* are:

- rapid implementation of all agreed changes in *ostis-systems*;
- permanent maintenance of a high-level of mutual understanding between all the systems that are part of the *OSTIS Ecosystem*, as well as all their users;

- corporate solution of various complex problems requiring the coordination of several (most often a priori unknown) *ostis-systems* and possibly some users.

Within the *OSTIS Ecosystem*, the concept of a *personal ostis-assistant* is specified. A *personal ostis-assistant* is the *ostis-system*, which is a *personal assistant* of the user within the *OSTIS Ecosystem*. Such a system provides opportunities:

- to analyze user activity and form recommendations for its optimization;
- to adapt to the mood of the user, their personal qualities, the general environment, the problems that the user most often solves;
- to permanently train the assistant in the process of solving new problems, while learnability is potentially unlimited;
- to conduct a dialog with the user in natural language, including in speech form;
- to answer questions of various classes, while if the system does not understand something, it can ask counter-questions itself;
- to independently receive information from the entire environment, and not just from the user (in text or speech form).

At the same time, the system can both analyze available information sources (for example, on the Internet) and analyze the physical world surrounding it, for example, surrounding objects or the appearance of the user.

Advantages of the *personal ostis-assistant*:

- the user does not need to store different information in different forms in different places: all information is stored in a single *knowledge base* compactly and without duplication;
- thanks to unlimited learnability, assistants can potentially automate almost any activity, not just the most routine one;
- thanks to the *knowledge base*, its structuring, and the means of searching for information in the *knowledge base*, the user can get more accurate information more quickly.

Personal assistants have a very different purpose and can be used for a wide variety of categories of users (patient, legal service, administrative service, customer, consumer of various services). The *personal ostis-assistant* can use the knowledge and data stored in other *ostis-systems*, such as *corporate ostis-systems*, to provide the user with more complete and up-to-date information. This can be especially useful for users who work with a lot of data and information. The *personal ostis-assistant* is automatically integrated with other *ostis-systems*, which allows it to work more efficiently with data and information. It can use machine learning and artificial intelligence technologies to adapt to user behavior and improve its productivity and efficiency. The *personal*

ostis-assistant can be created and configured to meet the specific needs of the organization and its processes, which can lead to significant economic and production advantages.

The *personal assistant* should take into account that the user roles in society can change, expand, as well as their interests and goals. At the same time, all *personal assistants* must be semantically compatible in order to understand each other and also have the ability to independently interact within various *corporate systems*, representing the interests of their users.

The *user interface* of the *personal intelligent assistant*:

- provides the user with the means to control their individual activities carried out jointly with the corresponding personal intelligent assistant;
- provides the unified nature of user interaction within the various communities in which they belong. The simplest type of community is a one-time dialog between two users.

Since the user interaction with the *OSTIS Ecosystem* occurs only through a *personal ostis-assistant*, the *user interface of the OSTIS Ecosystem* for the user is the *user interface* of their *personal ostis-assistant*. Such an *interface* should be *adaptive, intelligent, and multimodal*. The structure of such an *interface* was proposed in the work [17].

The *knowledge base of the ostis-system user interface* includes:

- usage context model:
 - user model;
 - environment model;
 - platform model;
- interface model;
- interface component model;
- user action model;
- logical rules for interface adaptation;
- model of sensory and effector subsystems to ensure multimodality;
- interface design techniques;
- model of interface design tools;
- etc.

The model of the user and their actions in the context of the *user interface of the OSTIS Ecosystem* should be stored only within the user's *personal ostis-assistant* and share this knowledge with other *ostis-systems*, if necessary.

User interface design is based on a component-based approach. Any *user interface component* can be described in the *ostis-system knowledge base*. An example of the “calculator” *user interface component* with its corresponding description fragment in the *ostis-system knowledge base* in the *SCg-code* is shown in Figure 1.

The *user interface of the personal ostis-assistant* by default contains only the *interface components* necessary

for the user to start interaction. At the same time, the *interface* is constantly being adapted based on the user model and adaptation rules.

At the same time, *ostis-systems* that are not *personal ostis-assistants*, as a rule, should not have a *user interface*. However, developers of such systems can add the necessary *user interface components* to the *knowledge base*, which can be displayed by a *personal ostis-assistant* when interacting with the user.

Thus, the general process of interaction of a *personal ostis-assistant* with a user can be described as follows:

- the *user interface of the personal ostis-assistant* is displayed to the user by default;
- when interacting with the user, the *user interface of the personal ostis-assistant* automatically adapts;
- the *personal assistant*, when interacting with the user, can use the *user interface components* of any *ostis-system* within the *OSTIS Ecosystem*, adapt such a *component*, and display it to the user, if necessary.

An example of using a *user interface component* within the *OSTIS Ecosystem* is shown in Figure 2.

The user, their *personal ostis-assistant*, and other *ostis-systems* (in the Figure, *intelligent systems* in mathematics, chemistry, biology, and geography are shown) are part of the *OSTIS Ecosystem*.

The developer of each *intelligent system* can describe the *user interface components* necessary for this system. In the example given, the “calculator” *component* was described for the mathematics system and the map component – for the geography system.

Since the user interacts with all *ostis-systems* only through a *personal ostis-assistant*, the *assistant* itself accesses the system necessary for interaction at the moment and loads the necessary *user interface component* of some applied *ostis-system* into its part of the *knowledge base* with the purpose of subsequent display to the user, taking into account the rules of adaptation, which are also stored in the *knowledge base of the personal ostis-system*.

It is important to note that a wide variety of *interfaces* entails the development of a large number of *components*. Both already designed *interfaces* and specified *interface components* can act as *reusable interface components* of the *ostis-system*. A large number of *reusable interface components* of the *ostis-system* creates the problem of storing and searching them. To solve this problem, within the *OSTIS Technology*, a *library of reusable interface components* of *ostis-systems* and a *manager of reusable ostis-systems components* are used [19]. The use of a library of *reusable interface components* of *ostis-systems* in the design of an applied system *interface* can significantly reduce the time for design, as well as reduce the requirements for the initial qualification of the developer. This is achieved by designing the *interface* from pre-prepared *interface models*, which also allows improving the quality of the designed *interface* [20].

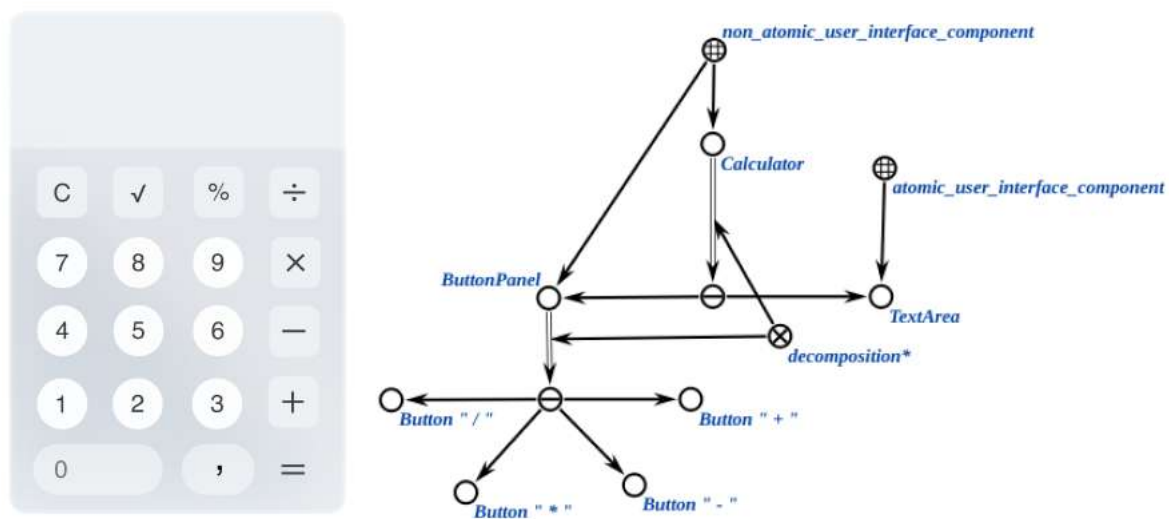


Figure 1. Example of a description of a user interface component

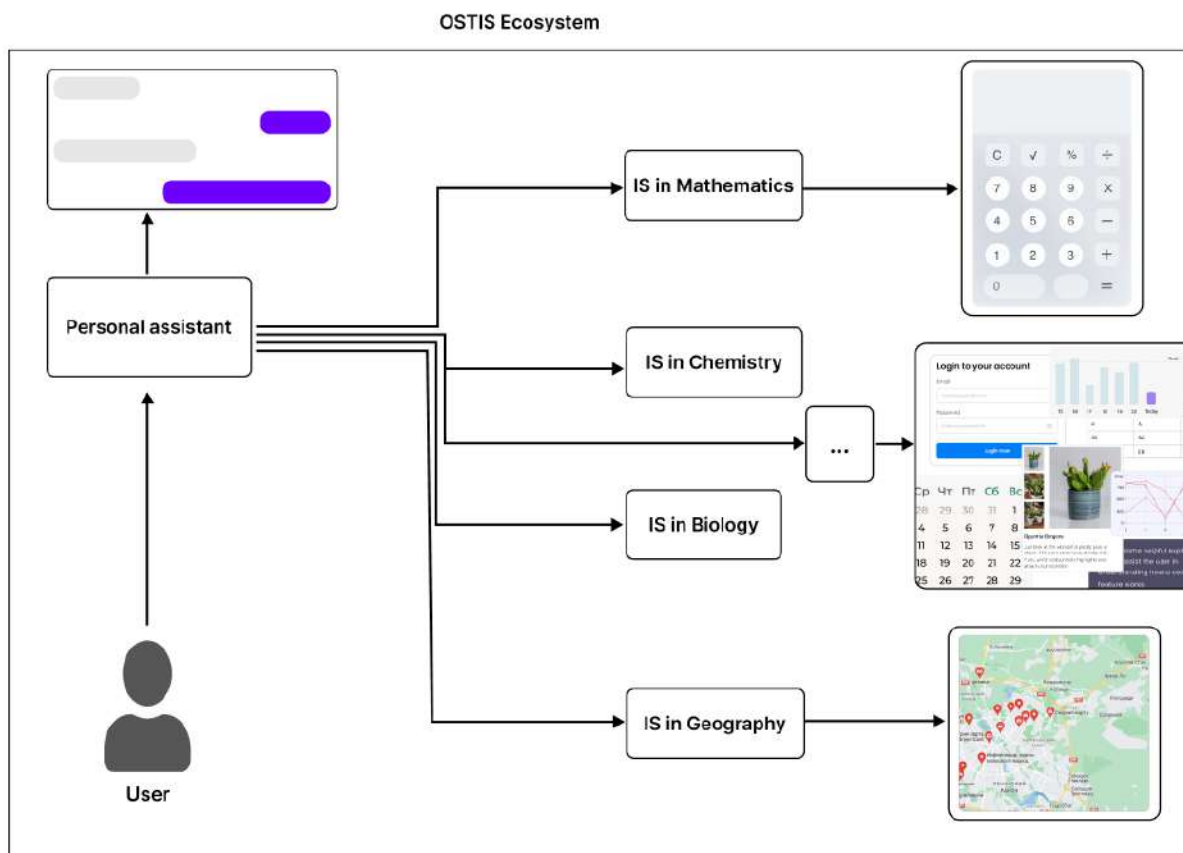


Figure 2. Components of the user interface of the OSTIS Ecosystem

A unique feature of the *OSTIS Technology* is ensuring the compatibility for the *components of ostis-system knowledge bases, ostis-system problem solvers, and ostis-system interfaces* due to a single unified formal basis. Thus, a *user interface component* for its work must usually include not only a description of its visual characteristics in the *knowledge base* but also *components of the problem solver* (for example, *sc-agents*), as well as the necessary fragments of some *subject domain*. So, in Figure 3, fragments of the *interface components, the problem solver, the knowledge base* for the “calculator” *user interface component* are represented in the *SCg-code*.

In the demonstrated Figure, the representation for the *operator of the sum of two numbers* and a fragment of the *Subject domain of numbers and numerical structures* necessary for the operation of the “calculator” *component* is displayed. Due to the unified representation of all the necessary parts, such a *component* can be easily integrated into any *ostis-system*, including the *personal ostis-assistant system*.

IV. CONCLUSION

Within the article, the problem of usability of information services and systems was considered. Their existing diversity requires additional efforts from users to explore their features and gain interaction skills.

To solve this problem, relevant works on the use of *personal assistants* were considered. It was concluded that in addition to using *personal assistants*, it is necessary to ensure the compatibility of various services and systems, which can be implemented by creating an *ecosystem of semantically compatible systems*.

An approach was proposed, which assumes the use of the *OSTIS Technology*, which includes the *OSTIS Ecosystem* and *personal ostis-assistants* to ensure effective and comfortable user interaction with the *ecosystem*.

Within the proposed approach, the *user interface of the OSTIS Ecosystem* is considered as the *user interface of a personal ostis-assistant*, since the user interacts with the *ecosystem* only through their *personal assistant*. The principles of the *user interface of the OSTIS Ecosystem* were described, the main of which is the *component approach* to design and the possibility for a *personal assistant* to use any *user interface component* within the *OSTIS Ecosystem*.

REFERENCES

- [1] C. Meurisch, M.-D. Ionescu, B. Schmidt, and M. Mühlhäuser, “Reference model of next-generation digital personal assistant: Integrating proactive behavior,” in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, ser. UbiComp ’17. New York, NY, USA: Association for Computing Machinery, 2017, p. 149–152. [Online]. Available: <https://doi.org/10.1145/3123024.3123145>
- [2] C. Meurisch, C. A. Mihale-Wilson, A. Hawlitschek, F. Giger, F. Müller, O. Hinz, and M. Mühlhäuser, “Exploring user expectations of proactive ai systems,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 4, Dec. 2020. [Online]. Available: <https://doi.org/10.1145/3432193>
- [3] J. Paay, J. Kjeldskov, E. Papachristos, K. M. Hansen, T. Jørgensen, and K. L. Overgaard, “Can digital personal assistants persuade people to exercise?” *Behaviour & Information Technology*, vol. 41, no. 2, pp. 416–432, 2022.
- [4] A. Akbar, “Proactivity in intelligent personal assistants: A simulation-based approach,” in *Multi-Agent Systems*, D. Baumeister and J. Rothe, Eds. Springer International Publishing, 2022, pp. 423–426.
- [5] (2023, Mar) Siri web page. [Online]. Available: <https://www.apple.com/siri/>
- [6] (2023, Mar) Microsoft Cortana web page. [Online]. Available: <https://www.microsoft.com/en-us/cortana>
- [7] (2023, Mar) Google Now web page. [Online]. Available: <https://www.digitaltrends.com/mobile/what-is-google-now/>
- [8] (2023, Mar) LGvoice mate web page. [Online]. Available: <https://www.lg.com/us/mobile-phones/V5985/Userguide/388.html>
- [9] J. Todorov, V. Valkanov, and I. Popchev, “Intelligent personal assistant for aiding students,” 10 2017.
- [10] S. Ahmad and M. Imran, “Asr based intelligent personal assistant,” 03 2019.
- [11] V. Golenkov, N. Gulyakina, and D. Shunkevich, *Otkrytaya tekhnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheskii sovmestimyykh gibridnykh intellektual’nykh komp’yuternyykh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*. Minsk: Bestprint, 2021, (In Russ.).
- [12] V. Golenkov, N. Gulyakina, I. Davydenko, and D. Shunkevich, “Semanticheskii tekhnologii proektirovaniya intellektual’nykh sistem i semanticheskii asociativnyye komp’yutery [Semantic technologies of intelligent systems design and semantic associative computers],” *Otkrytye semanticheskii tekhnologii proektirovaniya intellektual’nykh sistem [Open semantic technologies for intelligent systems]*, pp. 42–50, 2019.
- [13] I. Davydenko, “Semantic models, method and tools of knowledge bases coordinated development based on reusable components,” in *Otkrytye semanticheskii tekhnologii proektirovaniya intellektual’nykh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2018, pp. 99–118.
- [14] D. Shunkevich, “Agentno-orientirovannyye reshateli zadach intellektual’nykh sistem [Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems],” in *Otkrytye semanticheskii tekhnologii proektirovaniya intellektual’nykh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2018, pp. 119–132, (In Russ.).
- [15] A. Boriskin, M. Sadouski, and D. Koronchik, “Ontology-based design of intelligent systems user interface,” *Otkrytye semanticheskii tekhnologii proektirovaniya intellektual’nykh sistem [Open semantic technologies for intelligent systems]*, pp. 95–106, 2017.
- [16] M. Sadouski, “Semantic-based design of an adaptive user interface,” in *Open Semantic Technologies for Intelligent Systems*. Springer, 2022, pp. 165–191.
- [17] —, “The structure of next-generation intelligent computer system interfaces,” *Open semantic technologies for intelligent systems*, pp. 199–208, 2022.
- [18] A. Zagorskiy, “Principles for implementing the ecosystem of next-generation intelligent computer systems,” *Open semantic technologies for intelligent systems*, pp. 347–356, 2022.
- [19] M. K. Orlov, “Comprehensive library of reusable semantically compatible components of next-generation intelligent computer systems,” *Open semantic technologies for intelligent systems*, pp. 261–272, 2022.
- [20] M. Sadouski and A. Zhmyrko, “Methodology and tools for component interface design of next-generation intelligent computer systems,” *Open semantic technologies for intelligent systems*, pp. 279–284, 2022.

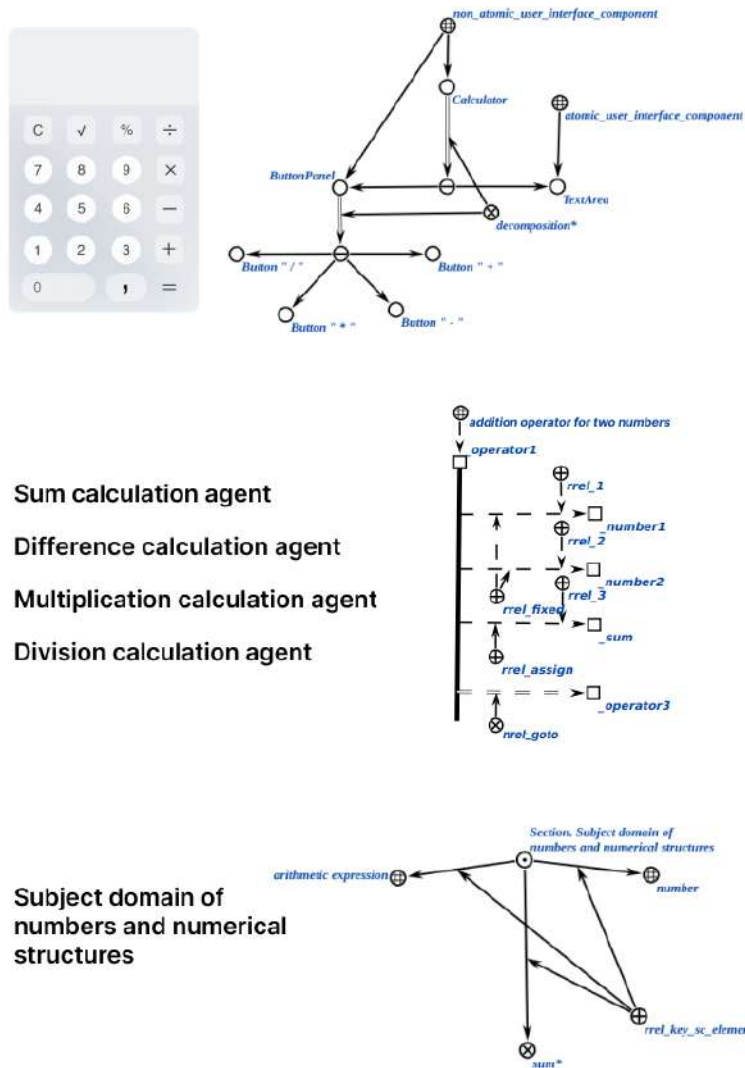


Figure 3. Components of the interface, the problem solver, and the knowledge base for the “calculator” component

Пользовательский интерфейс Экосистемы OSTIS

Садовский М. Е.

В статье рассматриваются принципы организации взаимодействия пользователя с Экосистемой OSTIS, понятие пользовательского интерфейса Экосистемы OSTIS, применение компонентного подхода к проектированию адаптивных интеллектуальных мультимодальных интерфейсов ostis-систем.

Received 13.03.2023

User interaction design in next-generation immersive systems

Aliaksandr Halavaty and Valeria Bahai
and Danila Bernat
Belarusian State University
Minsk, Belarus
Email: alex.halavaty@gmail.com, lerika1307@mail.ru,
danila.bernat@gmail.com

Katsiaryna Halavataya
Email: kat.golovataya@gmail.com

Abstract—This article explores the issues of designing intelligent systems using augmented reality (AR) and virtual reality (VR) technologies. These technologies are based on the use of three-dimensional representation of the surrounding real or pre-designed virtual scene, the capabilities of these systems for determining the orientation of user and objects in space, intelligent analysis algorithms and machine learning. The complexity of developing individual algorithms, the need to take into account the specifics of the subject area for each applied system, interactive display based on three-dimensional and other types of representations necessitates the development of new approaches and design principles. The article focuses on the capabilities of OSTIS Technology for use in these tasks, and also provides examples of building a system for technical description of equipment and a guided quest in augmented reality.

Keywords—virtual reality, augmented reality, 3D representation, knowledge base

I. INTRODUCTION

When designing virtual reality (VR) and augmented reality (AR) systems, it is necessary not only to create a high-quality image and show it to the user at the right time – it is necessary to design the user’s environment, other characters, interaction with other characters and objects, and use informational environment and metadata to work with specific objects. The user must feel like a part of the space (in the case of virtual reality) or perceive the content in relation to the physical environment (in the case of augmented reality). At the same time, it is necessary to properly implement and design various optical and auditory effects, correctly track user movements with sufficient precision, implement ways to interact with various objects and subjects of the surrounding space. In addition, all visualisations, reactions, interactions and updates to the designed space and objects must be carried out in real time in order not to disrupt user’s immersion and their psycho-emotional state while using the system. This research area is related, among other things, to the field of immersive systems design. The paper [1] describes four different components of user interaction immersion: sensory-motor immersion, cognitive immersion, emotional immersion and spatial immersion.

Another recently popular concept is mixed reality (abbreviated XR). Mixed reality systems approach constructing immersive interactions by separating surrounding systems, subjects, objects and informational space into two different types of so-called world layers — physical world layer, containing objects that are naturally present and can be perceived by the user as separate entities, and informational world layer, that considers various types of information and data that can be used to design and enhance user’s interaction with the system. Mixed reality strives to build a combination of objects from these two different world layers. At the moment, the concepts of virtual reality and augmented reality are most often used in applied systems [2].

Virtual Reality is a high-level user interface that includes real-time simulation and interaction through multiple sensory channels. In VR, the scene is constructed artificially and then perceived by the user by effectively overtaking their sensory channels (primarily visual and auditory) and supplying pre-designed data to these channels instead of the real world data.

The main differences between virtual reality and other interfaces that use the visual presentation of information to the user are:

- 3D stereo vision,
- user’s viewpoint control – the system changes the visible display point of view in the constructed space by using user’s viewpoint and view orientation changes in the real world as input,
- possibility of interaction with the virtual environment in real time.

Augmented Reality is the result of introducing any additional constructed types of visual representation of data into the regular visual information feed, in order to provide additional information about the environment and change the user’s perception of this environment.

Promising applied areas in which the number of virtual and augmented reality systems is expected to grow, in addition to gaming and entertainment industry, include healthcare and medical devices, education, staff develop-

ment and training, manufacturing, automotive industry, marketing and advertising, logistics and transport, retail, scientific visualization.

The main advantages of technology include:

- Visualization and interaction with objects and concepts that are hard or impossible to implement in the real world.
- Visualization of 3D concepts, modeling, viewing objects and scenes from different perspectives using a more natural way of observing them compared to traditional display devices.
- Modelling and research of hazardous and potentially dangerous environments, objects and situations.
- Promoting innovative styles and methods of study and learning.

II. ANALYSIS OF EXISTING APPROACHES

Despite the prospects and advantages of using these technologies, there are a number of limiting factors that significantly hinder all developments in the area and applying them to more real-world problems:

- High complexity of application systems development for VR, AR and XR. When designing a virtual reality application, it is necessary to consider the specifics of the target hardware optical system, user movement tracking sensors, and APIs for retrieving this information from the device, which can be fundamentally different for different manufacturers. When constructing a scene observed by the user, it is necessary to generate a high-quality image, use directional acoustic models, etc., which requires significant hardware processing capacity on the end-user device, or stable and very low-latency remote compute server link. Augmented reality systems are additionally based on a technical vision hardware and algorithms, and building such models also presents multiple challenges. All these factors mean that it is practically impossible to design such systems by individual developers, and developing such systems requires sufficient expertise.
- Insufficient amount of existing content. The lack of content, in addition to the complexity of developing individual systems, is also due to the fact that existing virtual reality systems are based on different hardware and application platforms, often tied to the physical characteristics of the processing system and end-user equipment, which makes them software or hardware incompatible.
- Unsatisfactory user experience. For virtual reality, a very common complaint from users is feeling of dizziness and motion sickness, caused by the specifics of the human perception and vestibular system, and heavily compounded by the possible low quality or high latency of the visual feed. For augmented reality, main cause of poor user

experience is rather limited capabilities of the end-user device, leading, in turn, to poor performance of recognition systems. All this leads to a negative perception of the user's experience of interaction with such systems and, consequently, to the rejection of the use of technology.

- Lack of proper operational procedures when using the devices and poor legal base. For virtual reality systems, in particular in the educational process, it is critical to establish proper safety regulations and operational allowances to prevent unwanted side-effects to the users. It is also very important to further study the issues of intellectual property and the potential violation of the boundaries of the user's personal life.
- The total cost of equipment and content for end users.

Together, these factors often lead to unreasonably high cost of equipment and content, leading, in many cases, to the practical impossibility of using these kinds of systems. Coordinating various approaches for designing and applying such systems on a conceptual level will facilitate the implementation of relevant solutions, which, in turn, can greatly increase interoperability, integration and convergence of all related systems and knowledge bases.

III. SUGGESTED APPROACH

This paper proposes a unified description for the process of designing virtual and augmented reality systems in the form of an ostis-system knowledge base [3]. As part of building the knowledge base and implementing a platform for developing systems in this subject area, the following stages are proposed:

- creation of a framework for semantic representation of the scene and user interaction;
- systematization of the subject area, existing approaches and establishing links with related areas;
- development of a set of agents that implement the operational specifics of virtual or augmented reality systems.

Within the framework of the knowledge base, it is proposed to establish the following blocks:

- 3D representation of surrounding objects and scenes;
- description of the physical principles of operation and equipment specifications;
- description of the principles of user immersion when using a virtual reality system;
- technical vision methods and algorithms used in virtual and augmented reality systems;
- generation of images, object models and scenes in the user's visibility area;
- semantic representation description of three-dimensional scenes and objects associated with the subject area;

- description of scenarios of user interaction with the system.

Next, we will consider in more detail the basic principles of designing virtual and augmented reality scenarios.

IV. SEMANTIC REPRESENTATION SCENES AND INTERACTIONS IN VIRTUAL REALITY

The basis and a primary distinctive feature of all immersive systems is the creation of the effect of presence. To describe this concept and recreate the reality of the scene in human perception, paper [4] indicates the need to create three main illusions: illusion of place (inducing the feeling of being in a simulated place), illusion of realism (making sure the environment feels natural for the user) and illusion of impersonation (aligning user's virtual avatar to match their perception of self). These concepts are achieved by influencing human senses in a specific way. Primary methods are using panoramic stereoscopic displays (visual senses), surround sound (auditory senses) and tactile feedback (haptic senses), all of which combined also indirectly influence user's equilibrioception. At the same time, all designed interaction should be coordinated and take sensory-motor correspondence effect. Creating all the additional effects that a person perceives, even in a simple physical scene, is a very hard process that requires knowledge from many areas. At the same time, actual implementation for most of these effects can be inferred from context, semantic representation of the scene, and a proper knowledge bases in related subject areas.

Thus, using the semantic representation of the virtual reality scene, it is possible to establish action subject (user), description of the place and conditions for this action (may include an additional auditory and haptic feedback), as well as descriptions of the object of the action and all additional objects within the scene. Basic audio and tactile interaction can then be generated from this shared description. To support such a representation, it is necessary to create a knowledge base that includes basic actions and conditions, which will later be compared with the scene contents, user and object conditions.

Within the framework of OSTIS Technology, interaction and implementation of actions can be implemented by agents [5]. To design a virtual reality scene, two types of agents are proposed - generating agents and provisioning agents. Generating agents select an appropriate 3D representation and corresponding models according to the semantic content of the scene, determine and selecting the necessary auditory components, and select appropriate tactile feedback for all the interactions that can be performed by the user in this scene. Provisioning agents map this set of possible responses of the system, scene and user actions on the physical capabilities (configuration) of the end user equipment; this includes dynamic rendering

resolution selection, ensuring proper reaction times for auditory and physical tactile interaction, and adjusting other scene generation parameters to maintain a balance between feed quality and perceived user latency.

To form a three-dimensional scene model, the semantic description of a three-dimensional scene given in [6] can be used, which consists of specifying individual objects, their absolute properties and referential relations. The use of referential relations can be effectively used in the process of rendering individual scenes.

For example, if the VR scene is of a person walking down a gravel path in a forest on a sunny day, then a 3D model of the surrounding forest can be generated from the database. In addition, the analysis of existing scenes can be used to generate background auditory effects (like the noise of foliage), and comparing actions and descriptions of the user and the environment can be used to generate reaction auditory effects (the sound of steps on a gravel road). Tactile interaction can be generated by user actions - for example, if the user picked up a pebble, it can be communicated using general vibration feedback in case of regular controller or a more detailed feedback in case of a haptic feedback glove. Therefore, actions are not coupled to specific equipment, it is possible to expand the capabilities of the equipment whenever necessary, and proper action feedback can be determined fully only by using the semantic description component.

Thus, using semantic representation of a virtual reality scene as part of scene generation pipeline contributes to producing a greater immersion effect due to using this representation as a single source of truth for all possible interactions, which guarantees effect and reaction consistency while allowing for extension and effective complexity management due to abstracting underlying interactions, feed modifications and end user equipment feedback.

V. SEMANTIC ENVIRONMENT REPRESENTATION IN AUGMENTED REALITY

To design an augmented reality system, it is necessary to have a description of the physical world, a description of the informational world concept superimposed on it through augmented reality, and a description of their respective relationship. At the same time, the relationship should take technical vision system and object recognition method into account and ensure correct backprojection of the object model to display additional information on device screen in order to properly anchor virtual modelling space into the real world image and implement effects like object occlusion.

Augmented reality systems can use various types of sensors and, in turn, appropriate algorithms for orientation calculation and tracking. Existing local positioning and 3D reconstruction systems, as well as their representation within the OSTIS knowledge base, are described in

more detail in [6]. When designing augmented reality object tracking systems, all approaches for object tracking and scene anchoring can be divided into marker and markerless tracking [7].

To organize the operation of augmented reality system, four main types of agents have been identified:

- Device agents that process image from the camera or aggregate data from other types of sensors.
- Surface agents that are responsible for detecting and classifying surfaces for the purpose of anchoring, object placement and occlusion.
- Rendering agents that are responsible for visualization of 3D models.
- Application logic agents that are responsible for implementing user interaction scenarios and handling user input.

Later in this work, we also present examples of designing augmented reality systems based on semantic representation. When implementing concrete applications, in addition to designing the operation of the augmented reality system as an interface for the user, it is also necessary to describe relevant user scenarios. Designed scenarios can also be included into the ostis-system knowledge base and presented using an ontological description.

A. Application of augmented reality for technical device manuals

Technical device manuals and instructions are necessary elements of any type of hardware or equipment. Usage of augmented reality systems in the implementation of these kinds of manuals and instructions provides advantages in terms of clarity and visibility - for example, the user does not need to compare schemas, images or diagrams with the operating object in order to find the necessary structural elements and understand required operational, technical and maintenance procedures they need to perform. In addition, complex scenarios can be implemented: resolving non-standard situations can be presented in a sequence of steps, and intelligent visual analysis from the technical vision system can be used to spot and diagnose various issues. Many manufacturers are already implementing these types of applications, but at the moment all solutions in this area are limited to select products and distributed as narrowly-scoped standalone mobile applications. Using ontological representation of the technical specification and manual of the device [8] [9] makes it possible to design scenarios for personnel training, equipment diagnostics and repair. OSTIS Technology allows to extract an ontological description, associate it with a specific subject area and with the semantic representation of the augmented reality environment. As a result, ready-made user scenarios based on data received from the technical vision system are obtained. For the practical

implementation of object recognition possible to use additional solutions or use the algorithms described in the knowledge base directly. The following entities can be described:

```

device
:= [subject of technical manuals and descriptions]
⇒ includes*:
{
  part of the device
  ⇒ includes*:
  {
    device component
    := [A separate element responsible for
       specific functionality and having
       an additional description. Example:
       power button]
  }
}

```

Each entity can be highly nested and must have a description and visualization provided in the database. A system component can be an integral part of the device. If it is necessary to project the corresponding 3D device models during AR environment modelling, these entities can be compared with entities within the framework of the semantic representation of 3D scenes and objects [6]:

```

3D object
:= [a set of points in space connected to each other
    and having a shared semantic representation]

```

```

3D composite object
:= [object in 3D representation that supports decom-
    position into separate individual objects]

```

```

part of an object in 3D
:= [a set of points in space belonging to some object
    in a three-dimensional representation, which can
    be distinguished by its geometric or semantic
    representation]

```

```

3D scene
:= [collection of several objects in three-dimensional
    representation and data about their features and
    relative position in space, including absolute
    and relative referential oriented and unoriented
    relations]

```

The following entity types are also introduced to describe user interaction:

```

equipment status
:= [a set of technical characteristics of equipment
    related to the specifics of its operation and
    determining its performance]

```

equipment operation

:= [a sequence of actions designed to achieve a specific user goal; examples of operations can be switching on, diagnostics, repair, installation, dismantling, transportation, etc.]

equipment action

:= [user interaction with one of the device components in a specified way; example of an action would be pressing a button, plugging into an outlet.]

Based on these entity types, an arbitrary custom scenario can be described. For example: turning on (operation) of new equipment (equipment status) is carried out by the following sequence of operations and actions: unpacking (can be an operation or action depending on the device), installation (operation), plugging in (action), pressing the power button (component). Device diagnostics or repairs can be described in the same way. The connection of each component to its visual display and semantic 3D representation allows to display descriptions of the stages aligned with the corresponding components using augmented reality, for example, on a smartphone screen.

As a practical implementation, an augmented reality training application based on markerless technology was developed with a technical manual of the oscilloscope for use during laboratory work. This application allows to visualise equipment operating instructions in AR, and also provides repair instructions using a mobile device that supports iOS 16. Device, surface and rendering agents implemented in this system are based on the ARKit SDK [10]. After the application is launched, the scene is initialized and a surface agent is used to find a surface matching the appearance of the device. When required surface is found, a WorldAnchor is set, and descriptions of individual device components that are present in the frame are displayed on the smartphone screen. While working with the application, the user can move in space, and appropriate component descriptions are constantly updated to correctly match location and spatial orientation of the device. Position recalculation is a computationally expensive task, so it is performed in 0.4 second increments. The user can also choose one of the operations to indicate how they want to interact with the device. In this case, the application enters instruction mode, and user can perform all the steps in order or view the entire sequence of actions.

B. Augmented reality quest

One of the most commonly used formats for augmented reality systems is gaming. Characteristics of this format can be used in educational systems, marketing campaigns, tours, and many other areas. One of the examples of designing an augmented reality applications presented in

this paper is an augmented reality quest app - an educational and marketing application designed to familiarize applicants with an educational institution.

To create such an application, a set of scenes and interaction scenarios associated with them are described and added to ostis-system knowledge base, which will later be used as a data source.

Interaction implementation is based on a common location - some part of the space of the real world, which is described in terms of semantic scene representation and object relationships. These relationships provide a shared description of separate scenes and objects of these scenes in the real world. To design a scene in AR, additional relations are established between the objects of the real world and the virtual modelling space. Objects superimposed onto the scene are generated by the rendering agents, that use the same semantic description for the scene to work in a shared space. To perform visualisation of the resulting rendered scene, objects in the shared description and must be first located and recognized.

AR quest is formed as a sequence of the scenes and their respective descriptions. Each of these scenes also contains a description for a certain number of individual, which are, in turn, attached to the objects of the scene and can be presented to the user. An example of such a task is a set of questions about the observed object, a mini-game, an instruction to find a specific object, an instruction to move to the different location to change the scene. Application keeps track of task execution status using device agents that supply information about end user device spatial characteristics from device sensors. When the task is completed or scene change is detected, application may suggest new tasks.

To implement device, surface and rendering agents for proposed application, Unity platform and Vuforia software package [11] for AR were used. Developed system uses a combination of marker and markerless technology for tracking and searching for objects in a real scene. Application is designed to work under the Android OS. Managing content for describing scenes, objects of a real scene and superimposed objects of AR, as well as tasks for the user, can be performed using ontological representation of OSTIS knowledge base by implementing appropriate data mapping for translating ontological representation to the format used by corresponding platform APIs.

VI. CONCLUSION

The paper considers the possibilities of using OSTIS Technology for designing virtual and augmented reality systems, and discusses the possibility of using ontological descriptions of related subject areas as a basis for implementing these systems. For proposed concepts, descriptions of applications designed with the proposed approach are also provided.

The advantages of using OSTIS Technology for the presented tasks are:

- Introduction of a common concept and description system for different methods in a unified and consistent form.
- Support for convergence of VR and AR application design subject area with other applied subject areas, 3D scenes and environment modelling subject area, and computer vision subject area.
- Simplification of virtual reality systems development by selection of necessary scene elements according to scene semantic description to implement user interaction and achieve maximum user immersion.
- Ability to build a complex design technology using intelligent agents that rely on proposed description and utilise data about individual algorithms and user scenarios from the shared knowledge base.
- Ability to create integration tools for individual components, describe stages of various methods and build different types of internal representations in a unified way.

Using proposed approach enables to design dynamically expandable VR and AR systems that provide a greater immersion degree for the user, while using proposed semantic scene description for objects and their relations can be used to model various rendering and interaction scenarios without tightly coupling the implementation to specific hardware.

Building realistic scenes in virtual and augmented reality is a rather complicated process that must be able to achieve necessary degree of realism, while taking limitations on real-time scene rendering into account. Using OSTIS Technology to enable the possibility to use developments in related areas allows to form a new approach for developing less resource-intensive interaction scenarios.

ACKNOWLEDGMENT

The authors would like to thank the scientific team of Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics for their help in the work and valuable comments.

REFERENCES

- [1] S. Bjork and J. Holopainen, *Patterns in Game Design*. Charles River Media, 2004.
- [2] D. Saredakis, A. Szpak, B. Birkhead, H. A. D. Keage, A. Rizzo, and T. Loetscher, "Factors associated with virtual reality sickness in head-mounted displays: A systematic review and meta-analysis," *Frontiers in Human Neuroscience*, vol. 14, Mar. 2020. [Online]. Available: <https://doi.org/10.3389/fnhum.2020.00096>
- [3] V. Golenkov, N. Gulyakina, and D. Shunkevich, *Otkrytaya tekhnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheskii sovmestimyykh gibridnykh intellektual'nykh komp'yuternyykh sistem [Open technology for ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint, 2021.

- [4] M. Slater, "Immersion and the illusion of presence in virtual reality," *British Journal of Psychology*, vol. 109, no. 3, pp. 431–433, May 2018. [Online]. Available: <https://doi.org/10.1111/bjop.12305>
- [5] D. Shunkevich, "Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems," *Otkrytye semanticheskie tehnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 119–132, 2018.
- [6] K. Halavataya and A. Halavaty, "3d representation of objects in new generation intelligent computer systems," in *Open semantic technologies for intelligent systems*. Minsk : BSUIR, nov 2022, pp. 251–260.
- [7] J. C. P. Cheng, K. Chen, and W. Chen, "Comparison of marker-based and markerless AR: A case study of an indoor decoration system," in *Lean and Computing in Construction Congress - Volume 1: Proceedings of the Joint Conference on Computing in Construction*. Heriot-Watt University, Jul. 2017. [Online]. Available: <https://doi.org/10.24928/jc3-2017/0231>
- [8] N. Dorodnykh, O. Nikolaychuk, A. Yurin, and and, "Using ontological content patterns in knowledge base engineering for maintenance and repair of aviation equipment," *Ontology of Designing*, vol. 12, no. 2, pp. 158–171, Jul. 2022. [Online]. Available: <https://doi.org/10.18287/2223-9537-2022-12-2-158-171>
- [9] A. Timoshenko, A. Zuev, E. Mursalimov, V. Gribova, and A. Inzartsev, "Description and diagnosis of malfunctions in autonomous uninhabited underwater vehicles based on ontologies," *Ontology of Designing*, vol. 12, no. 3, pp. 310–324, Sep. 2022. [Online]. Available: <https://doi.org/10.18287/2223-9537-2022-12-3-310-324>
- [10] P. Nowacki and M. Woda, "Capabilities of ARCore and ARKit platforms for AR/VR applications," in *Advances in Intelligent Systems and Computing*. Springer International Publishing, May 2019, pp. 358–370. [Online]. Available: https://doi.org/10.1007/978-3-030-19501-4_36
- [11] M. Sarosa, A. Chalim, S. Suhari, Z. Sari, and H. B. Hakim, "Developing augmented reality based application for character education using unity with vuforia SDK," *Journal of Physics: Conference Series*, vol. 1375, no. 1, p. 012035, Nov. 2019. [Online]. Available: <https://doi.org/10.1088/1742-6596/1375/1/012035>

Проектирование пользовательского взаимодействия в иммерсивных системах нового поколения

Головатый А. И., Багай В. Д.,
Бернат Д. А., Головатая Е. А.

Данная статья посвящена рассмотрению вопросов проектирования интеллектуальных систем, использующих технологии дополненной и виртуальной реальности. Данные технологии основаны на использовании трехмерного представления окружающей реальной или спроектированной сцены, особенностях систем определения ориентации человека и объекта в пространстве, алгоритмах интеллектуального анализа и машинного обучения. Сложность разработки отдельных алгоритмов, необходимость учета специфики предметной области для каждой прикладной системы, отображение на основе трехмерных и других видов представлений вызывает необходимость разработки новых подходов и принципов проектирования. В статье основное внимание уделено возможностям Технологии OSTIS для использования в данных задачах, а также приводятся примеры построения системы технического описания оборудования и квеста в дополненной реальности.

Received 13.03.2023

Developing Birds Sound Recognition System Using an Ontological Approach

Yauheniya Zianouka and Dzianis Bialiauski
Liesia Kajharodava and Aliaksandr Trafimau
Vitalij Chachlou and Juras Hetsevich
*United Institute of Informatics Problems
National Academy of Sciences of Belarus*
Minsk, Belarus
{evgeniakacan, dzianis.bialiauski,
lesia.cordell, cncntrt,
vitalikhokhlov, yuras.hetsevich}@gmail.com

Vadim Zahariev and Kuanysh Zhaksylyk
*Belarusian State University of
Informatics and Radioelectronics*
Minsk, Belarus
zahariev@bsuir.by, kuanysh.zhk@gmail.com

Abstract—The article presents an intelligent model of automated voice recognition systems (on the example of birds). To develop it, a dataset of birds' voices was annotated and processed using Mel-Frequency Cepstral Coefficient as an effective tool for modelling the subjective pitch and frequency content of audio signals. For composing and training the model, Convolutional Neural Network is used to implement high level results. The possibilities of using ontological approaches and OSTIS technology for further improvement of the quality of ML models are shown.

Keywords—recognition system, machine learning, dataset, automatic processing, Mel-frequency cepstral coefficients (MFCCs), Convolutional Neural Network, EfficientNet, ontological approach

I. INTRODUCTION

There are many voice signals in nature, and each type of sound signal has its own function. In general, sound signals can be divided into singing and other voices (for example, streams, noise). Singing is a more melodic type of bird voice. It is usually longer and more complex than the stream. Due to many variations of the sound signals of different bird species, there is a problem of their recognition. The relevance of creating such a system is due to the fact that all existing developments on the recognition of animal species are not suitable for Belarusian birds. Only some of them are able to recognize the sound signals of European species, which also affect our project. However, the development and use of an automated system facilitates the recognition process, the system itself has recognition accuracy problems that need to be mentioned. Software for determining the species of animals (for example, birds) by voice signals is based on mathematical calculation models (most often twisted neural networks), which, with insufficient training, can make a computational error that leads to incorrect determination of the biological species we need [1]. Thus, the tasks were set:

- to develop methodological foundations for collecting, annotating and recognizing animal voice signals on the territory of Belarus (in terms of technical implementation).
- to compose a structural scheme for automated recognition of animal voice signals for autonomous continuous monitoring of rare, threatened species and indicator species.
- to increase the accuracy of recognition of animal species (for example, birds).

II. DATASET FOR TRAINING THE RECOGNITION MODEL

A dataset of electronic voice signals corpora is a substantial component for training the recognition model. The primary source of publicly available arrays of animal vocalisations used in the dataset is the Xeno-Canto (<http://www.xeno-canto.org>). Its resources are available for listening, downloading, and studying the characteristics of sound recordings. It is one of the largest sources of audio data of bird vocalisations collected from around the world. The site has API endpoints that can be used to automatically search, download data by scientific or common name of a species or family, region tags, sound types, country, etc.

Machine learning is heavily dependent on data. That's why one of the main tasks that need to be performed during its preparation is the annotation of audio recordings. When developing the method of data annotation for bird voice recognition systems, it is taken into account that the composition of the Belarusian fauna includes rare species of animals and birds, for which there is no or limited scope of annotated sounds and graphic data [2]. It is also considered that the selected data may contain different sounds of the same species, sounds of many other species that are heard together with this species.

In order to improve the results of processing and recognizing birds' voices, each individual audio file is

labelled with the name of its own species. The data downloaded during the collection phase from open sources may include some part of the annotated data, but needs to review and fix the annotation according to the audio event classes. Each audio part of the signal is detected as a silent audio segment or an audio event. Then it is assigned to its appropriate audio subclass based on the nature of its content. Next, all the annotated information with the corresponding timestamps is written to a text file markup. Since the data is mostly recorded in the birds' natural habitats, each recording may also contain some background information, including various noises from other animals or people. In order to adjust and improve the performance of the recognition algorithm, it is recommended to add and annotate recordings in the base for training the algorithm, where there are no bird sounds, but there is a background sound of the environment in which certain species of birds usually exist.

In our work, the system for recognizing the voices of birds was built on fourteen species: Parus Major (Sinica vialikaja), Fringilla Coelebs (Bierascianka), Turdus Philomelos (Drozd-spiavun), Emberiza Citrinella (Strynatka zvyčajnaja), Phylloscopus Collybita (Pi-ačuraŭka-cieŭkaŭka), Turdus Merula (Drozd čorny), Sylvia Atricapilla (Lieska-čornahaloŭka), Luscinia Luscinia (Salaviej uschodni), Acrocephalus Dumetorum (Čarotaŭka sadovaja), Erithacus Rubecula (Malinaŭka), Loxia Curvirostra (Kryžadziub-jalovik), Hippolais Icterina (Pierasmieška), Periparus Ater (Sinica-maskoŭka), Sylvia Communis (Lieska šeraja). Using the API (Application Programming Interface) a dataset was collected (audio recordings) according to the above mentioned bird species. The number of entries was about two hundred for each species. The criteria by which records were selected for training were as follows:

- The duration of audio recordings is more than three seconds and less than 10 minutes.
- Proximity by distance to Belarus (Minsk).

The following information is available for each uploaded audio file: Bird species (Specific epithet); Bird subspecies of the (subspecific epithet); The group to which the species belongs (bird, grasshopper); The name of the species in English; The name of the person who made the audio recording; The country where the sound was recorded; The name of the area where the recording was conducted; Geographical latitude of the place where the recording was conducted; Geographical longitude of the place where the recording was made; The type of sound that a bird makes (singing, streaming, etc.); Gender (female or male), etc.

Currently, a total dataset contains 21737 audio recordings with a planned test sample of 4348 audio recordings. On the basis of this corpus, work on improving recognition and optimization algorithms will continue.

III. DATA PREPROCESSING

Before starting to create a machine learning model and predicting or classifying, it is necessary to carry out preliminary data processing (Preprocessing) [3]. It is the first and integral step of machine learning, as the quality of the data and the useful information that can be extracted from it, directly affects the learning ability of the model. The main tasks at the data preprocessing stage are:

- Processing of zero values;
- Data normalisation (its transformation to some dimensionless units);
- Control of outliers. These are not errors, but values that abnormally stand out and can distort the model's operation;
- Processing of categorical features;
- The problem of multicollinearity (the presence of a high mutual correlation between two or more independent variables in the model).

The technology stack used to develop the recognition model includes Python programming language. Open Source Python Libraries are *Librosa*; *Tensorflow*; *NumPy*; *Keras*; *Pandas*. Development environments are *Jupyter* and *PyCharm*. Optimization of algorithms and metrics for building a recognition model allows reducing training time by orders of magnitude. Therefore, it usually makes sense to start building the model using reduced datasets, successively improving the process parameters (see figure 1).

IV. MODELS FOR BUILDING VOICE RECOGNITION SYSTEMS

To date, there are several approaches to the construction of voice recognition systems.

- Recognition models based on spectrograms. The audio signal is converted into a spectrogram - a visual representation of the signal's frequencies over time.
- Models based on the amplitude component of the signal without analysing the frequency characteristics. Very often, recurrent networks RNN, LSTM and GRUs are used as models in this case.
- Models, based on the synthesis of specific useful characteristics of the signal. Among such characteristics, one can distinguish mel-cepstral coefficients (MFCCs), coding with a linear predictor (LPC), gammatone filters (Gammatone filter banks).
- Hybrid models. Usually include several types of models for the synthesis of the recognition model to get the best recognition quality.
- Transfer learning. Training is based on an already pre-trained model on other data where weights are already present. At the same time, the model is further studied on the available data of audio recordings.

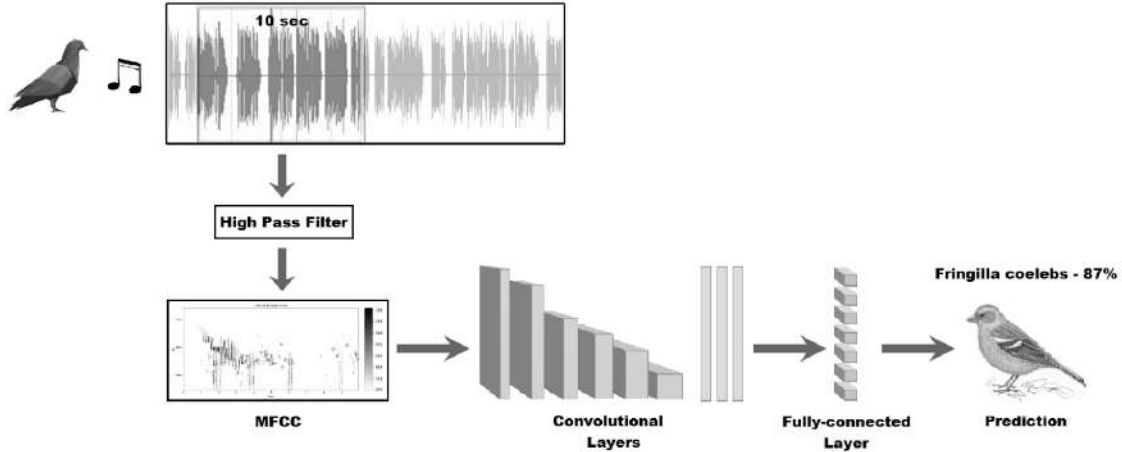


Figure 1. An Audio data preprocessing and neural network model

In our work, we use the first approach, since it certainly results in the best quality of recognized audio signals. For this, we need annotated audio recordings that allow training the model more accurately and recognize bird species.

V. RECOGNITION MODEL BASED ON SPECTROGRAMS

Spectrogram-based models are a powerful tool for sound prediction tasks due to their ability to capture both temporal and spectral features of audio signals. This approach has proven effective in a wide range of applications [4].

Every sound we hear consists of sound frequencies at the same time interval. The essence of a spectrogram is the visualisation of this set of frequencies on a single graph, as opposed to a sonogram, where only the amplitude of the signal is displayed. A spectrogram is a graphical representation of the spectrum of an audio signal as a function of time. It shows which sound frequencies are present in the audio recording at any given time. A spectrogram is built by applying a Fourier transform to short sections of an audio signal called windows. The resulting spectrum is then displayed as a colour map with time on the horizontal axis and frequency on the vertical axis. The colour of each pixel of the spectrogram corresponds to the amplitude of the corresponding frequency.

Mel-spectrogram is a graphical representation of an audio signal in which frequencies are represented on a Mel scale instead of the linear frequency scale used in a conventional spectrogram. The Mel scale is a reproducible scale based on human perception of sound. It is based on the fact that at low frequencies the audio signal can be distinguished with greater resolution, while at higher

frequencies the human ear is less sensitive to changes. Thus, the Mel scale reduces resolution at high frequencies and increases it at low frequencies to better match the human perception of sound. If we combine these two ideas into one, we get a modified spectrogram (MFCC, mel frequency cepstral coefficients), which filters out the frequencies of sounds that a person does not hear, and leaves the most characteristic ones.

Mel scale is calculated as (1):

$$Mel(f) = 2595 \log_{10} \left(1 + \frac{f}{700} \right), \quad (1)$$

$$C_n = \sqrt{\frac{2}{k}} \sum_{k=1}^K (\log S_k) \cos[n(k-0.5)\pi/k], \quad n = 1, 2, \dots, N \quad (2)$$

where $Mel(f)$ is the logarithmic scale of the normal frequency scale f . Mel scale has a constant mel-frequency interval, and covers the frequency range of 0 Hz - 20050 Hz. The Mel-Frequency Cepstral Coefficients (MFCCs) are computed from the FFT power coefficients which are filtered by a triangular band pass filter bank. The filter bank consists of 12 triangular filters. The MFCCs are calculated as (2)

where $S_k(k = 1, 2, \dots, K)$ is the output of the filter banks and N is the total number of samples in a 20 ms audio unit.

Since birds sing at high frequencies, a high-pass filter is used to remove unnecessary noise (leave frequencies at a minimum value of 1400 Hz). These coefficients will be sent to the input of the recognition model. The Python library librosa was used to generate the Mel spectrogram:

```
signal, sr = librosa.load(fp, sr=self.sr,
duration=self.duration, mono=self.mono)
```

```
S_ms = librosa.feature.melspectrogram(y=signal,
sr=sr, n_fft=self.n_fft, hop_length=self.hop_length,
n_mels=self.n_mels, fmin=self.fmin, htk=self.is_htk, )
```

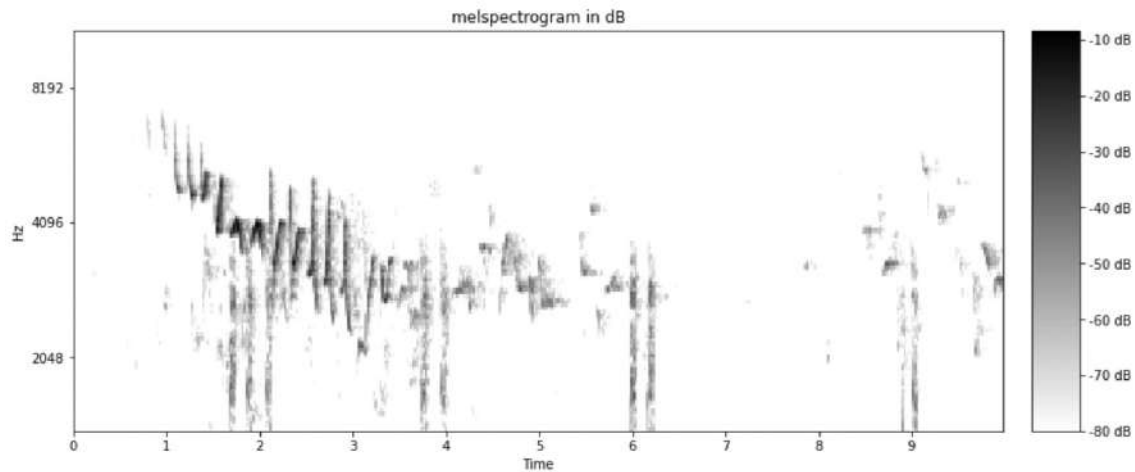


Figure 2. A spectrogram of preprocessing the voice of a Sylvia Atricapilla bird (Lieska-cornahalouka)

```
S_ms_db = librosa.amplitude_to_db(np.abs(S_ms),
ref=np.max)
```

Where parameters are

```
sr=22050, n_fft=1024, hop_length=512, n_mels=128.
```

An example of preprocessing the voice of a Sylvia Atricapilla bird (Lieska-cornahalouka) is a high-pass filter, calculation of mel frequency cepstral coefficients and display as a spectrogram on a graph is shown on figure 2.

VI. MODEL ARCHITECTURE

Having a Mel spectrogram of the bird's sounds, we will recognize its species among a pre-prepared list of species (classes). The model extracts specific characteristics of the processed data (images), sends them to the input of a deep neural network, and outputs a set of probabilities that correspond to the likelihood that the image belongs to each of these classes. We assume that the class with the highest of these probabilities is the output of the model. This is a deep neural network of the CNN type (Convolutional Neural Network - a convolutional neural network) for recognizing the image class of birds' voices. It is a type of deep learning algorithm that is commonly used in image and video processing applications. Convolutional neural networks are specifically designed to process data that has a grid-like topology, such as images, where the goal is to classify an image into one of several categories. They work by applying a series of filters or convolutions to the input data, which extract features from the input data. These features are then used to make predictions about the input data. They are also used in object detection to locate objects within an image, and in image segmentation for partitioning an image into regions or segments. Overall, CNNs have revolutionised the field of computer vision and have led to significant improvements in image and video processing applications.

In the architecture, we used the EfficientNetB3 network as well as three more layers:

(*Flatten, Dropout, Dense with the softmax function as an output*) to build a CNN network. The following network structure is *EfficientNetB3 Flatten Dropout Dense(with softmax function as output)*.

EfficientNet (<https://arxiv.org/abs/1905.11946>, <https://keras.io/api/applications/efficientnet/efficientnetb3-function/>) is a modern development of a convolutional neural network from Google Brain (see figure 3). The main objective of EfficientNet was to thoroughly test how to scale the size of convolutional neural networks. For example, one can scale ConvNet based on layer width, layer depth, input image size, or a combination of all these parameters. Thus, the final model was built on the basis of EfficientNetB3 and 14 different classes (bird species) with Adam optimizer, categorical cross-entropy loss function and balanced class weights.

VII. QUALITY OF MODEL TRAINING

The quality of bird voice recognition by the represented model is 75.6 percent of the total accuracy on the test data, where

- 7 classes have an F1-score of more than 80 percent
- 3 classes have an F1-score between 70 percent and 80 percent
- 2 classes have an F1-score between 60 percent and 70 percent
- 2 classes have F1-score less than 60 percent.

Moreover, a prototype of the model for automated bird voice recognition "*Bird Sound Recognizer*" was created for the implementation of autonomous continuous monitoring of rare threatened species, indicator species and the state of biodiversity in forest ecosystems [4]. It is located on the online platform corpus.by (<http://corpus.by/BirdSoundsRecognizer/?lang=en>) and is open and free for using [5].

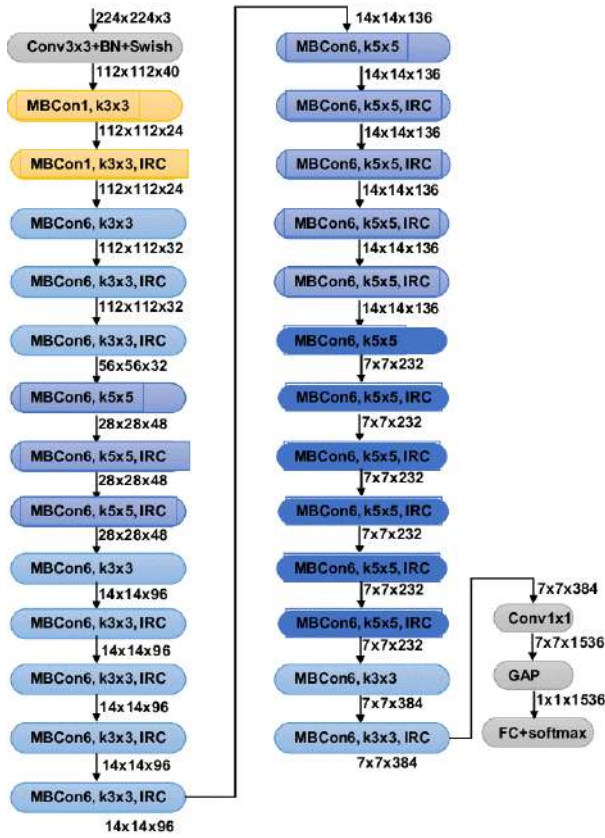


Figure 3. EfficientNet Network architecture

VIII. MODEL IMPROVEMENTS BASED ON ONTOLOGICAL APPROACHES

One of the ways to further improve the quality of the model is the ability to take into account, in addition to the sound signal itself, various meta-information: place, time, habitat of a particular species of birds. For a comprehensive solution of this problem, allowing to take into account the various features of the subject area, it is proposed to use the ontological approach.

When it comes to recognizing birdsong from an audio signal, ontological approaches are essential for several reasons. First and foremost, ontologies provide a way to organize and structure the vast amounts of data that are available for each uploaded audio file. In this case, the meta-information available for each audio file can be used to create a taxonomy of bird species, subspecies, and groups. This makes it easier to organize and analyze the data, and can help to improve the accuracy of the recognition process.

In addition, ontological approaches can help to address issues related to data inconsistency and incompleteness. By using a well-defined and structured ontology, it becomes easier to identify and correct inconsistencies in the data, such as misspellings or variations in naming conventions.

Furthermore, ontologies can help to facilitate the integration of different data sources, such as data from different recording locations or from different researchers. By using a common ontology, it becomes easier to merge and compare data from different sources, which can help to improve the accuracy of the recognition process.

Finally, ontological approaches can help to support the development of intelligent systems that are capable of learning and adapting to new data. By using a well-defined ontology as a basis for machine learning algorithms, it becomes possible to create systems that can recognize new bird species and adapt to new recording conditions.

The choice of technology for implementing the ontological approach is also important. Because it should provide a sufficient level of flexibility and scalability to integrate various types of systems and knowledge into them. As a technological basis, it is proposed to use the OSTIS technology, the main advantages and specifics of which in signal processing and integration with DNN are presented in papers [6-10]. An example of a fragment of a top-level ontology using OSTIS sc-code is presented below.

Sylvia atricapilla

```

:= [Eurasian blackcap]
  ∈ English language
:= [Lieska-čornahaloŭka]
  ∈ Belarusian language
  ∈ specie
  ⊂ sylvia
  ∈ genus
  ⊂ old world warbler
  ∈ family
  ⊂ perching birds
  ∈ order
  ⊂ vertebrates
  ∈ phylum
  ⊂ animal
  ∈ kingdom

```

```

⇒ habitat*:
{
• Eastern Europe
  ⇒ coordinates*:
  {
• [From: 54.1343° N, 28.5079° E]
• [To: 54.5028° N, 28.8794° E]
  }
• Western Asia
• Northwestern Africa
}

```

By using a structured and well-defined ontology to organize and analyze the available data, it becomes possible to create intelligent systems that are capable of accurately recognizing a wide range of bird species, subspecies, and groups.

IX. CONCLUSION

The article describes a model for recognizing the voices of Belarusian birds. It is based on the analysis of a mel-frequency cepstrum (MFCC, mel-frequency cepstrum). The Mel spectrogram is a graphical representation of an audio signal in which frequencies are represented on Mel scale instead of the linear frequency scale used in a conventional spectrogram. For machine learning of the model, a deep neural network of the CNN (Convolutional Neural Network) was used to recognize the image class of birds' voices, since this type of network is more suitable for image recognition tasks. To build the CNN network, we chose the EfficientNetB3 network, as well as three more layers (Flatten, Dropout, Dense with the softmax function as output). Thus, the final model was built on the basis of EfficientNetB3 and 14 different classes (bird species) with Adam optimizer, categorical cross-entropy loss function and balanced class weights.

The overall recognition quality of the model was 75.6 percent. The conduction of the experiment was fulfilled on 14 species of birds with 200 records for each of the species using a CNN-based model with spectrograms as a characteristic of the input signal to the model. In the future, it is planned to expand the list of bird species for recognition to 116. The next step of the project is to monitor the continuous signal for the detection of the bird's voice in real time.

To solve this task, a recognition system will be designed using another data set, the audio files of which are annotated in detail by ornithologists, taking into account the time stamps for the bird's voice.

In paper also proposed approaches to further improve the quality of ML-models through the use of the ontological approach and OSTIS technology.

REFERENCES

- [1] F. Briggs, R. Raich, X. Z. Fern Audio Classification of Bird Species: A Statistical Manifold Approach. ICDM 2009, The Ninth IEEE International Conference on Data Mining, Miami, Florida, USA, 6-9 December 2009, pp. 51–60.
- [2] D. Stowell, M. D. Plumbley An open dataset for research on audio field recording archives: freefield1010. *ArXiv*, 2013, vol. abs/1309.5275.
- [3] D. Stowell, M. D. Plumbley Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning. *PeerJ*, 2014, vol. 2.
- [4] S. A. Hajdurau, D. I. Latysevich, A. A. Bakunovic, L. I. Kajharodava, V. A. Chachlou, Ja. S. Zianouka, Ju. S. Hiecevic Madel baz danych dlia tehnalohii avtomatyzavanaha raspznavannia halasavych sihnalau zvyviol [Database model for the technology of automated recognition of animal voice signals]. XXI International Scientific and Technical Conference "Development of Informatization and State System of Scientific and Technical Information RINTI-2022", UIIP NASB, Minsk, 2022, pp. 236–240.
- [5] J. S. Hiecevič, J. S. Zianouka, A. S. Trafimaŭ, A. A. Bakunovič, D. I. Latyševič, A. Ja. Drahu, M. M. Sliesarava, M. S. Tukaj Kompleks srodkaŭ realizacyi zadač štučnaha inteliectu dlia bielaruskaj movy [A complex of means to implement tasks of artificial intelligence for the Belarusian language]. *Piervaja vystavka-forum IT-akadiemhrada Iskusstviennyj intelliect v Bielarusi*. UIIP NASB. Minsk. 2022. P. 64–73. (In Belarusian)

- [6] V. V. Golenkov, N. A. Gulyakina, D. V. Shunkevich Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems, Bestprint, Minsk, 2021, P. 690
- [7] V. Golovko and et el. Integration of artificial neural networks and knowledge bases *Open Semantic Technologies for Intelligent Systems (OSTIS-2018)*, BSUIR, Minsk, 2018, pp. 133–146.
- [8] M. Kovalev, A. Kroshchanka, V. Golovko, Convergence and integration of artificial neural networks with knowledge bases in next-generation intelligent computer systems *Open Semantic Technologies for Intelligent Systems (OSTIS-2022)*, BSUIR, Minsk, 2022, pp. 173–186.
- [9] V. Zahariev, E. Azarov, K. Rusetski. An approach to speech ambiguities eliminating using semantically-acoustical analysis *Open Semantic Technologies for Intelligent Systems (OSTIS-2018)*, BSUIR, Minsk, 2018, pp. 211–222.
- [10] V. Zahariev, K. Zhaksylyk, D. Likhachov, N. Petrovsky, M. Vashkevich, E. Azarov. Audio interface of next-generation intelligent computer systems *Open Semantic Technologies for Intelligent Systems (OSTIS-2022)*, BSUIR, Minsk, 2022, pp. 239–250.

Разработка системы распознавания звуков птиц с использованием онтологического подхода

Зеновко Е., Белявский Д., Кайгородова Л.,
Трофимов А., Хохлов В., Гецевич Ю.,
Захарьев В., Жаксылык К.

В работе предложена модель распознавания голосов птиц Республики Беларусь, основанная на анализе мел-спектрограмм (MFCC, mel-frequency cepstrum). Мел-спектрограмма — это графическое представление звукового сигнала, в котором частоты представлены в мел-шкале вместо линейной шкалы частот, используемой в обычной спектрограмме. Шкала Mel — шкала высоты звуков, отсеивающая частоты звуков, которые человек не слышит, и оставляет самые характерные, находящиеся на одинаковой дистанции для слушателя. Для машинного обучения модели была использована глубокая нейронная сеть типа CNN (Convolutional Neural Network) для распознавания класса изображения голоса птиц, так как именно этот вид сети больше подходит для задач распознавания изображений. Для построения сети CNN мы применили сеть EfficientNetB3, а также еще три слоя (Flatten, Dropout, Dense с функцией softmax в качестве выхода). Таким образом, окончательная модель была построена на основе EfficientNetB3 и 14 различных классов (видов птиц) с оптимизатором Адама (Adam optimizer), категориальной функцией потерь перекрестной энтропии (categorical cross-entropy loss function) и сбалансированными весами классов.

При проведении данного эксперимента на 14 видах птиц с 200 записями для каждого из видов и использованием модели на базе CNN со спектрограммами в качестве характеристики сигнала для входа на модель, получено общее качество распознавания 75,6 процентов. В дальнейшем планируется расширение списка видов птиц для распознавания до 116.

Показаны возможности использования онтологических подходов и технологии OSTIS для дальнейшего повышения качества моделей машинного обучения.

Received 13.03.2023

Ontological Approach to Chinese Language Interface Development in Intelligent Systems

Longwei Qian

*Department of Intelligent Information Technology
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: qianlw1226@gmail.com*

Abstract—This article is devoted to development of a unified semantic model of natural language interface, which allows combine various linguistic knowledge on natural language processing into a single knowledge base, as well as deep integration of logical models on rules, neural network models and other problem solving models for natural language processing towards solving conversion natural language texts into knowledge base fragments and generation natural language texts from knowledge base fragments. Moreover main principles of building Chinese language interface is described on the basis of unified model of natural language interface. Finally the developed Chinese language interface is evaluated in order to prove the effectiveness of unified semantic model.

Keywords—ontology, knowledge-based system, knowledge acquisition, text generation, Chinese language processing

I. INTRODUCTION

With the development of knowledge engineering technology, knowledge-based intelligent systems are actively developed in the direction of solving various complex tasks (for example, natural language processing, images analysis, speech analysis and so on). Moreover the knowledge-based question answering system considered as the next generation search engine is the top research topic in the research area of knowledge-based intelligent systems. As a key component of knowledge-based intelligent systems, the natural language interface is oriented to achieve information exchange between human users and knowledge base of intelligent systems in natural language texts (especially declarative sentences). With the development of intelligent systems, the need to conversion between natural language texts and knowledge base through natural language interface is increasingly prominent. In framework of this article we mainly consider the task of *conversion input natural language texts into knowledge base fragments* and *generation natural language texts from knowledge base fragments* in natural language interface.

Both conversion natural language texts and generation natural language texts are considered subtasks of natural language processing. The natural language processing is a kind of so-called complex problem, the research of which

has been a hot topic in the research area of artificial intelligence.

II. RELATED WORKS

In accordance with the types of processed natural language and the range of knowledge base of intelligent systems, in our works natural language interface of intelligent systems is divided into following four classes:

- natural language interface that is independent of the specific natural language and the specific domain;
- natural language interface that is dependent on the particular natural language, but is independent of the particular domain;
- natural language interface that is independent of the particular natural language, but is dependent on the particular domain;
- natural language interface that is dependent on the particular natural language, and as well as is dependent on the particular domains;

Due to the time-consuming and laborious development of world-wide knowledge base and its narrow application scenarios [1], natural language interface that is independent of the particular natural language, but is dependent on the particular domain is the main object of our research in this article. The more detailed description about classes of natural language interface can be seen in [2].

The task of conversion natural language texts into knowledge base fragments is considered as the factual knowledge extraction. In our works this task solved in the natural language interface refers to obtaining factual knowledge (mainly named entities and relations between them) from natural language texts, then formed in the knowledge representation.

In the early stage the factual knowledge extraction is performed in the closed domains, which often requires predefined types of named entities and relations between them. The factual knowledge is extracted from natural language texts using artificial constructed templates and rules, then is formalized in RDF [3]. Nowadays various large language modelings with the help of neural network models are applied for factual knowledge extraction

[4]. However the training of large language models requires expensive hardware equipment and a large of training corpus with annotations, which are time-consuming and laborious to obtain and construct. The factual knowledge extraction using rules doesn't require a huge of training corpus, but the construction of rules manually is inefficient. The factual knowledge extraction from open domains is to extract factual knowledge without requiring a predetermined vocabulary to define the types of named entities and relations between them [5]. This task solution uses a common syntax and lexical constraints for extracting factual knowledge from natural language texts [6]. Early systems are focused on extracting factual knowledge from English sentences. For other natural languages, such as Chinese, the structure and the grammatical features of English language usually are not totally applied for Chinese language. Therefore it is necessary to consider the characteristics of the specific language when extracting factual knowledge. Moreover, when developing factual knowledge extraction systems there is no unified basis, which leads to time-consuming and laborious for systems development.

For generation natural language texts from knowledge base fragments, in the early text generation systems widely use templates and grammar rules to generate natural language texts from structured data (for example, tabular data, fragments in RDF, fragments in OWL and others) [7], [8]. However, the constructed rules and templates are highly dependent on the specific application fields and specific natural language. The construction of rules and templates requires significant amount of labor and time. Recently the large language modelings also are applied for text generation in various applications (for example, text summary, caption generation, text generation from RDF and others) [9], [10]. In these applications the obtaining and construction of high-quality aligned corpus is a huge challenge in solving text generation tasks using large language models.

For development of natural language interface, in modern knowledge-based intelligent systems, the following problems still need to be considered:

- In modern intelligent systems, the lack of unification for development of natural language interfaces leads to significant overhead costs for integration of various components (e.g. knowledge base on natural language processing, component for factual knowledge acquisition, component for natural language texts generation) in the process of developing natural language interfaces;
- Due to the diversity of natural languages, when solving the factual knowledge extraction from open domains and text generation, it's necessary to take into account the characteristics of specific natural languages. However, the lack of unified model to integrate linguistic knowledge at various levels (e.g.

lexical aspect, syntactic aspect and others) for natural language processing into unified knowledge base significantly complicates the construction of various systems with their use;

- Whether conversion natural language texts into knowledge base fragments or generation natural language texts from knowledge base fragments, in modern systems, the solution of two tasks usually requires the use of various types of problem solving models. Existing ontology-based approaches are only applicable to the factual knowledge extraction from closed domains. Moreover the lack of unified principles for using various problem solving models for natural language processing (for example, logical models on rules, neural network models and so on). In turn, when development of specific natural language interface the overhead costs increase.

The solution of two above-mentioned tasks, in generally, requires a combination of various linguistic knowledge on natural language processing and the use of various problem solving models for natural language processing. In this article we proposed to use ontological approach to develop a unified semantic model of natural language interfaces, which has ability to implement factual knowledge extraction and text generation. From the perspective of the model structure, the unified semantic model mainly consists of semantic model of knowledge base of linguistic and semantic model of corresponding problem solver for natural language processing, which effectively combines linguistic knowledge at various levels and integrates various problem solving models for two above-mentioned tasks solution.

III. PROPOSED APPROACH

The ontological approach within OSTIS Technology framework [11] is proposed to develop the unified semantic model of natural language interface of intelligent systems for solution of conversion natural language texts into knowledge base fragments and generation natural language texts from knowledge base fragments. The intelligent systems developed using the OSTIS technology is called knowledge-driven computer systems (ostis-systems). The SC-code is used as an internal formal language for the semantic representation of knowledge in the memory of ostis-systems and provides a unified version of information encoding and a formal basis for developing model of ostis-systems. Several universal variants of visualization of SC-code [11], such as SCg-code, SCn-code, SCs-code will be shown below. Ontological model of any entity described by SC-code will call sc-model.

Within OSTIS Technology framework, natural language interfaces of the ostis-systems is considered as the specialized ostis-system focusing on solving the specific tasks in natural language interfaces [12] (in our work,

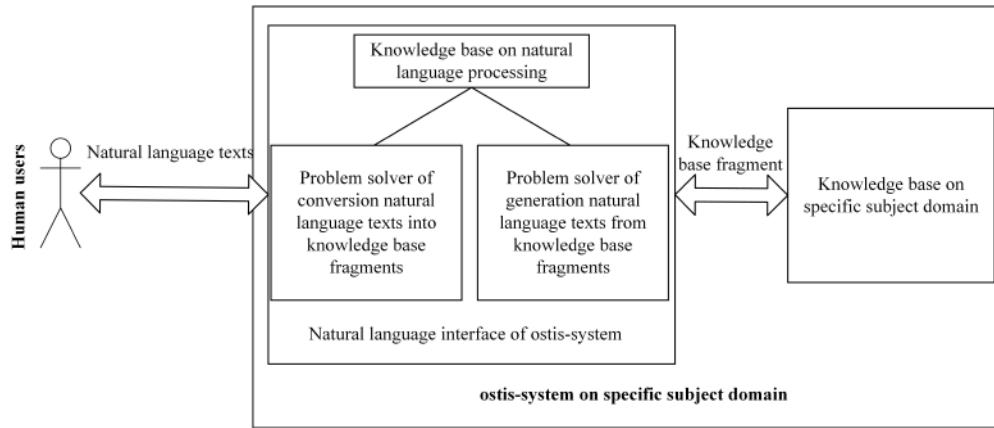


Figure 1: Model of process of natural language processing in the natural language interface

conversion natural language texts into knowledge base fragments and generation natural language texts from knowledge base fragments). The development of sc-model of natural language interface generally includes sc-model of knowledge base (mainly consists of sc-model of knowledge base of linguistic, actions for texts analysis and actions for texts generation) and sc-model of problem solvers of natural language interface. Due to the use of component approach, the development of the entire natural language interface comes down to development and improvement of separate specified components (e.g. knowledge base on natural language processing, component for natural language texts analysis, component for natural language texts generation). The model of the process of natural language processing (Figure 1) in the natural language interface describes the overall process of processing of natural language texts fragments with knowledge base fragments of intelligent systems.

In order to implement conversion natural language texts into knowledge base fragments and generation natural language texts from knowledge base fragments, it is necessary to describe the various linguistic knowledge for natural language processing, the construction of extraction rules, rules and templates for text generation. The development of SC-model of knowledge base of linguistic is to consider structure of knowledge base as a hierarchical system of subject domains and their corresponding ontologies, shown below in SCn-code:

SC-model of knowledge base of linguistics

```

:= [SC-model of knowledge base on natural language
    processing]
<= section decomposition*:
{
• Subject domain of lexical analysis
• Subject domain of syntactic analysis
• Subject domain of semantic analysis
}

```

These subject domains describe specification of linguistic knowledge at various levels (for example, knowledge on lexical analysis, syntactic analysis, as well as extraction rules, templates for text generation and others) respectively. Usually the knowledge base on natural language processing is not built from scratch. In this work, SC-model of knowledge base is built taking into account existing knowledge bases, such as Wordnet, Verbnnet, Treebank, Mandarin VerbNet, Chinese Treebank and others.

In addition to the various type of linguistic knowledge provided in the knowledge base of linguistic, the natural language interfaces should perform some *actions* to solve corresponding tasks. Within OSTIS Technology framework each *internal action of ostis-systems* denotes some transformation performed by some sc-agent (or a group of sc-agents). Therefore when discussing SC-model of actions, we can consider corresponding sc-model of problem solvers. Within OSTIS Technology framework, sc-model of problem solvers is developed as a hierarchical system of agents (sc-agents) [13]. Such approach provides the flexibility and modularity of developed sc-agents, as well as provides the ability to integrate various problem solving models corresponding to these developed sc-agents. In term of abstract sc-agent, the abstract sc-agent is a certain class of functionally equivalent sc-agents, various items of which can be implemented in different ways to specific problems in different programming languages [13].

SC-model of problem solvers of natural language interface

```

<= decomposition*:
{
• SC-model of problem solver for
  conversion natural language texts into
  knowledge base fragments
• SC-model of problem solver for
  generation natural language texts from
  knowledge base fragments
}

```

}

Let us consider the main structure of SC-model of problem solver for conversion natural language texts into knowledge base fragments and generation natural language texts from knowledge base fragments in natural language interfaces of ostis-systems in SCn-code, respectively:

SC-model of problem solver for conversion natural language texts into knowledge base fragments

```

:= [SC-model of problem solver for natural language
    texts analysis]
⇐ decomposition of an abstract sc-agent*:
{• Abstract sc-agent of lexical analysis
⇐ decomposition of an abstract sc-agent*:
  {• Abstract sc-agent of decomposing
    texts into segmentation units
    • Abstract sc-agent of marking up
    segmentation units
  }
  • Abstract sc-agent of syntactic analysis
  • Abstract sc-agent of semantic analysis
  • Abstract sc-agent of extracting factual
    knowledge structures into the knowledge
    base
  • Abstract sc-agent of logical inference
}

```

The SC-model of problem solver for natural language texts analysis is constructed on the basis of the proposed following process for factual knowledge acquisition:

- natural language text is loaded into the interface;
- lexical analysis and syntactic analysis of the input natural language text is performed;
- named entities and relations between them is extracted based on the analyzed syntactic structure and extraction rules.

In principle, this SC-model of problem solver can potentially extract structured knowledge (generally, named entities and relations between them) from texts in different language into the knowledge base of the ostis-systems for a specific subject domain, but the construction of knowledge base on the specific natural language processing, which includes rules for specific natural language processing and extraction rules, will become more complex. In turn overhead costs of construction will increase.

For generation natural language texts from knowledge base fragments the classical pipeline of natural language generation is used as the basis to develop the SC-model of problem solver for generation natural language texts from knowledge base fragments. The developed SC-model of problem solver has higher flexibility. For specific natural language, the developed problem solver can be easily modified accordingly.

SC-model of problem solver for generation natural language texts from knowledge base fragments

```

:= [SC-model of problem solver for natural language
    texts generation]
⇐ decomposition of an abstract sc-agent*:
{• Abstract sc-agent determining sc-structure
  • Abstract sc-agent dividing determined
    sc-structure into basic sc-constructions
  • Abstract sc-agent determining the
    candidate sc-constructions
  • Abstract sc-agent transferring candidate
    sc-constructions into message triples
  • Abstract sc-agent text planning
  • Abstract sc-agent for micro-planning
  • Abstract sc-agent for surface realization
}

```

The SC-model of problem solver for natural language texts generation is constructed on the basis of the proposed following process for texts generation:

- a specific sc-structure (fragment of knowledge base) is selected in the knowledge base;
- the candidate basic sc-constructions from the sc-structure is determined, then is translated into a message triple (in the form of subject-relation-object);
- the resulted natural language texts is generated from the message triple as output.

It is worth noting that the composition of sc-constructs has sc-arcs that have specific meanings. Therefore sc-constructions with sc-arcs need to be converted into the corresponding message triples in form of text, which is easier to represent in the form of natural language texts.

The developed unified semantic model of natural language interface ensures the flexibility of developing a specific natural language interface and integration of various components (knowledge base on natural language processing, component for conversion natural language texts into knowledge base fragment and component for text generation) in the interface. The development of natural language interface consists in the development of individual components independently of each other. It is flexible to adjust and make extensions of linguistic knowledge and sc-agents for tasks solution in specific natural language interface. The more detailed description about function of each abstract sc-agent can be seen in [2].

IV. IMPLEMENTATION OF CHINESE LANGUAGE INTERFACE

On the basis of unified semantic model of natural language interfaces of ostis-systems, we can implement a specific natural language interface of intelligent help systems for various subject domains. In this section we will describe the implementation of the prototype of Chinese language interface of a intelligent help system

about discrete mathematics. For developing Chinese language interface it's necessary to construct knowledge base on Chinese language processing and corresponding problem solvers for conversion Chinese language texts into sc-structures and generation Chinese language texts from sc-structures, which has ability to integrate logical models on rules and neural network models for Chinese language processing. The detailed processing stage of conversion Chinese language texts into sc-structures and generation Chinese language texts from sc-structures will be shown in followings.

A. factual knowledge extraction from Chinese language texts

Currently there are some restrictions for extracting factual knowledge from Chinese language texts:

- the processed Chinese language texts are Chinese declarative sentences;
- there are specific factual knowledge (named entities and relations between them) in the Chinese declarative sentences;
- due to features of Chinese language, the result of decomposition of Chinese declarative sentences into segmentation units greatly influences the factual knowledge extraction.

In this section the general processing stage of conversion Chinese declarative sentence into sc-structure will be shown in the followings.

Step 1: From the point of view of OSTIS technology, any natural language text is a file (sc-node with content or so-called sc-file). The Chinese declarative sentence shown in our example is represented in such a node in Fig 2 and describes: "有限集合(the finite set) , (comma) 严格地(strictly) 包含 (includes) 二元组(pairs)". (full stop)".

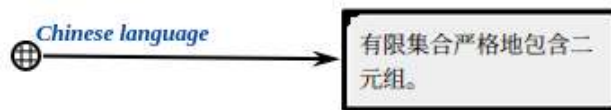


Figure 2: The representation of the Chinese sentence

As shown in Fig 2, according to the written tradition of Chinese language texts, Chinese characters are written one after the other and there are no natural gaps between them. As we know, the lexeme is a term commonly used for lexical analysis in European languages processing. However in Chinese language processing the segmentation unit is considered as the smallest unit. In the "Modern Chinese word segmentation standard used for information processing", a word in Chinese language is represented as a segmentation unit. The precise definition of segmentation units is "a basic unit for Chinese language processing with certain semantic or grammatical functions".

Step 2: The Chinese declarative sentence is decomposed into separate segmentation units, lexical analysis is carried out. Afterwards syntactic structure or semantic structure of

sentence is analysed, the relations between input sentence and divided segment units, as well as between these segment units in sentence are revealed. The analyzed results of input Chinese sentence is shown in the Figure 3.

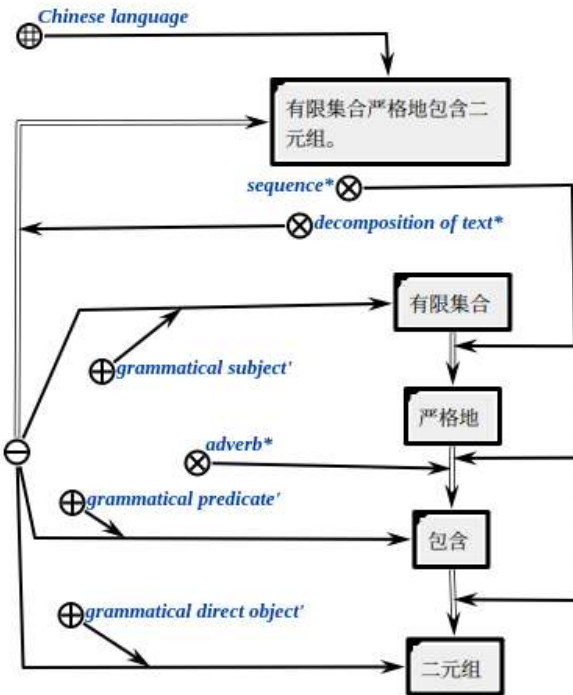


Figure 3: The syntactic structure of input Chinese sentence

Step 3: The factual knowledge that mainly consists of named entities and relations between them is extracted based on previous text analysis and extraction rules without contradiction detection. The resulted constructed knowledge base fragment (sc-structure) from input Chinese declarative sentence is shown in the Figure 4.

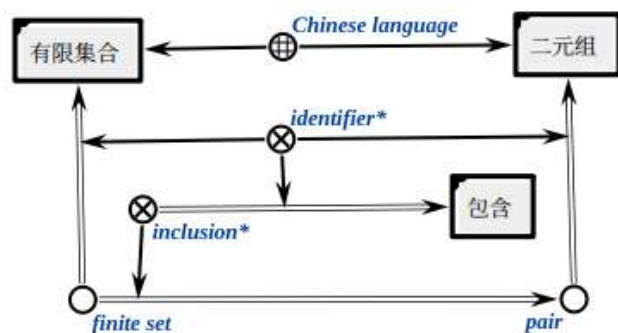


Figure 4: The constructed sc-structure from input Chinese sentence

It is important to note that in this case, a knowledge base fragment can be directly converted into knowledge base without linking extracted named entities and relations between them from the input Chinese sentence with the corresponding entities and relations defined in the knowledge base of intelligent help system.

B. text generation from knowledge base

In this section the processing stage of generation Chinese declarative sentence from sc-structure is described. The processing stage is roughly divided into two steps: firstly converting sc-structure from knowledge base into message triples; then generating Chinese declarative sentence from translated message triples. The description about concept message triple can be found in [14].

In our works there are some restrictions for Chinese language texts generation from knowledge base fragments:

- the knowledge base fragment is completed and has sc-elements with identifiers in Chinese language;
- the generated Chinese language texts are Chinese declarative sentences.

Step 1: The selected sc-structure is divided into standard basic sc-constructions, afterwards from which the candidate sc-construction is selected and will be converted into "message triple", then into resulted Chinese sentences. A candidate sc-construction (belong to standard basic sc-construction) is shown in SCg (Figure 5). The candidate sc-construction contains sc-elements with identifiers in Chinese language. Identifiers in Chinese language of each sc-element of sc-construction have corresponding specific segmentation units of Chinese language.

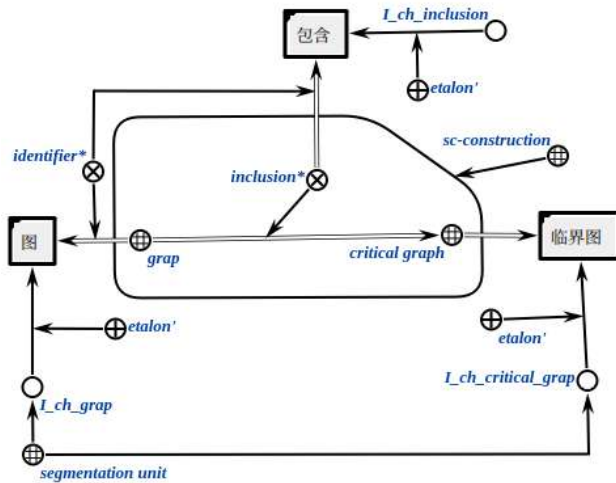


Figure 5: The determination of candidate sc-construction

Step 2: The candidate sc-construction is transferred to message triple. The converted message triple consists of sc-files (sc-node with content) containing segmentation units written by trained native Chinese speakers and verified by others. The message triple that corresponds to candidate sc-construction is generated in the Figure 6, in which each sc-element is a file corresponding to a certain segment unit in Chinese language. The contents of some sc-files (e.g. "临界图(critical graph)") correspond to the identifier of sc-element in the sc-construction, meanwhile the contents of some sc-files are added when building message triple.

It is important to note that relation of each message triple is the core. Sometimes the relation represents the specific meaning of sc-arc or sc-edge in the sc-structure in form of texts. The main task of text generation is to find suitable text fragment to explain the relation of each message triple in order to generate fluent texts. In general, the subject and object of each message triple are kept constant.

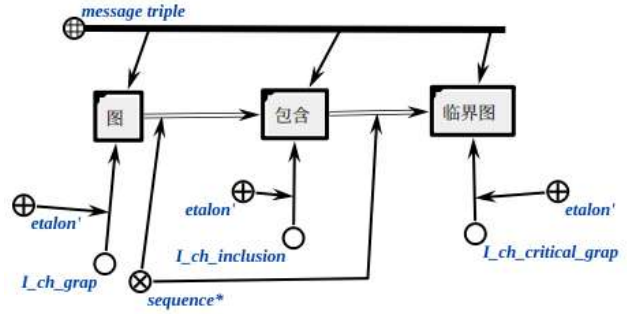


Figure 6: The message triple for candidate sc-construction

Step 3: Finally the sc-files are concatenated with certain form of that segment units to generate the resulting Chinese narrative sentence according to the permissible sequence on the constructed template for message triple with the relation "inclusion" (Figure 7). When generating result texts for some natural languages, word forms are changed according to syntactic rules (e.g. capitalizing the first word in a sentence, subject-verb agreement and others), and then added to the result texts. The relation *reference expression** is a quasi-binary relation, connecting a word to its combinatory variants.

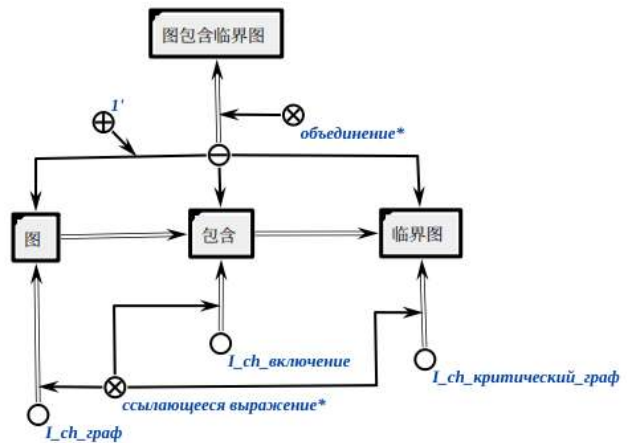


Figure 7: The generated Chinese declarative sentence

For some European languages, The inflected form of the lexical units in sc-files (e.g. singular or plural and other inflected forms) is expressed in the resulted generated texts according to the syntactic rules of a particular natural language. However, due to features of Chinese language, the processing of this step is relatively easier. In this

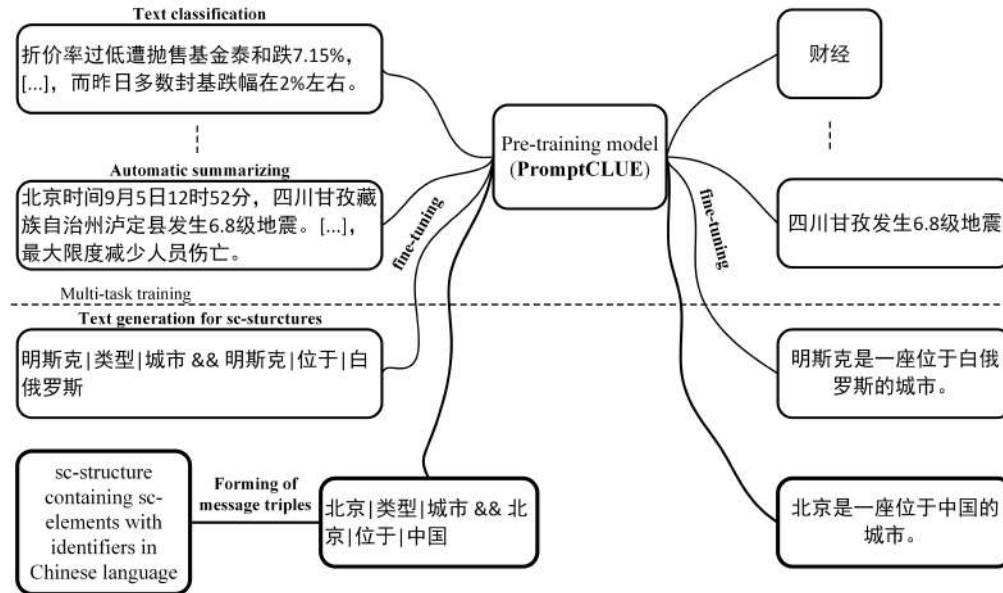


Figure 8: Diagram for text generation based on the pre-training model PromptCLUE

example, when generating the resulted Chinese sentence, the referring expression of each segmentation unit means the final form of the segmentation unit in the resulted generated Chinese sentence. According to the constructed template, the referring expression of the segmentation unit "图 (graph)" is the subject. The segmentation unit "临界图 (critical graph)" is considered as object. The generated Chinese declarative sentence describes "图 (graph) 包含 (inclusion) 临界图 (critical graph)."

Within Technology OSTIS framework some relations are already predefined in the IMS system for the development of ostis-systems, for example "inclusion*", "equivalence*" and so on. For these finite relations (we call domain-independent relations), templates is suitable for text generation. However in various subject domains there are a large amount of infinite relations. In this case, the neural network models can be integrated into the problem solver. In the Figure 8 shown the process of using pre-training model and fine-tuning paradigm to solve the tasks of generation Chinese sentence from sc-structure.

For task of generation Chinese sentence from sc-structure we use the pre-training PromptCLUE, which uses an encoder-decoder architecture using Transformer model and is pre-trained on several sets of Chinese language processing tasks using a huge Chinese corpus (hundreds GB of Chinese corpus) [15]. Afterwards we fine-tuned this model using task of generation Chinese texts from sc-structure (constructed dataset in from of *message triple/Chinese sentence* pairs). After fine-tuning on the PromptCLUE model, our retrained model can be used to generate Chinese texts from pre-processed sc-structures into message triples.

C. evaluation for Chinese language interface

In order to prove the effectiveness of the sc-model of natural language interface within OSTIS Technology framework, the developed Chinese language interface is currently being evaluated mainly in the following three aspects:

- evaluation of the knowledge base on Chinese language processing;
- evaluation of the efficiency of sc-structures generation;
- evaluation of the Chinese texts generation.

In order to compare the knowledge base on Chinese language processing with other similar existing knowledge bases used for Chinese language processing, the following proposed criteria for comparison of knowledge base within OSTIS Technology framework are highlighted:

- form of knowledge base structuring;
- independence of subject domains from each other;
- form of knowledge representation and form of knowledge storage in the knowledge base;
- possibility to solve problems using logical statements;
- presence of means to visualize the knowledge base.

In Table. I shown the result of comparing the developed knowledge base on Chinese language processing with other knowledge bases about Chinese language processing according the selected criteria.

In principle, on the basis of sc-model of knowledge base in natural language interface, linguistic knowledge at various levels based on existing knowledge base can be integrated in unified knowledge base on Chinese language processing. In addition, the various extraction rules or

Table I: Evaluation of knowledge base on Chinese language processing

Knowledge base	Criteria					
	structured representation and storage of knowledge	subject domain of words	subject domain of sentences	linguistic knowledge on phrases and others	presence and use of logical statements to solve problems	presence of means to visualize the knowledge base
Grammatical KB of Contemporary (GKB)	+	+	-	-	-	-
Mandarin Verb-Net	+/-	+	-	-	-	-
HowNet	+	+	+	-	-	-
Chinese Treebank 8.0	-/+	-	+	-	-	-
Knowledge base on Chinese language processing	+	+	+	+	+	+

templates for Chinese texts generation also can be built in knowledge base on Chinese language processing. This advantage is completely absent from other knowledge bases. Moreover, the developed knowledge base on Chinese language processing is structured into the respective subject domains. Sufficient independence between subject domains allows team development, which significantly reduces the time and labor costs in developing a knowledge base compared to developing other knowledge bases.

To evaluate the efficiency of Chinese text analysis (conversion Chinese texts into knowledge base fragments), the ideal way is to calculate the similarity between the sc-structure existing in the knowledge base and the corresponding sc-structure generated by the problem solver of Chinese text analysis. In our situation, the sc-structure is a graphical structure with identifiers in Chinese language. In [16], an approach was proposed for calculating the similarity between semantic graphs (sc-structures), focused on checking the answer to the target question. Therefore the approach can be used to calculate the similarity between the sc-structure existing in the knowledge base and the corresponding sc-structure.

However, this approach does not take into account the influence of the identifiers of each element in sc-structures. The additional metric exact matching is always used for evaluating the effectiveness of knowledge acquisition. The exact matching means that the identifiers of each extracted element (named entities and relations between them) must exactly match the identifiers of the element in knowledge base. To calculate the similarities between the standard sc-structures in knowledge base and the sc-structures generated by the problem solver of Chinese text analysis, we manually selected several different kinds

of sc-structures.

Table II: Evaluation of similarities between sc-structures

	Three-element construction	Five-element construction	Non-standard construction	Total
Number	15	15	10	40
Average similarity score	0.8125	0.8387	0.7273	0.7928

In Table. II shown the results. Depending on the complexity of the sc-structures, the different numbers for different types of sc-structures is selected, then calculate the average similarity score for these sc-structures, finally calculate the overall similarity score to evaluate the efficiency of the problem solver.

As can be seen from Table. II, as the complexity of sc-structures increases, the similarity score decreases. Overall the developed problem solver still achieves a relatively good result.

Table III: Evaluation of exact matching of identifiers

	Precision	Recall	F1
Problem solver of Chinese text analysis	0.8289	0.7875	0.8076
CORE	0.8308	0.6750	0.7448

According to the metric exact matching, the identifier of each element of selected sc-structures was manually added in Chinese language by trained native Chinese speakers and verified by others. Currently there is the CORE system [17] that basically extracts structure in RDF from Chinese sentences. Therefore for metric exact matching, the CORE system can be used to evaluate the performance of the developed problem solver of Chinese text analysis.

In Table. III shown the experimental results. In summary, the results show that the use of series of Chinese text analysis and constructed extraction rules is effective in extracting knowledge base fragments without any specific human intervention.

To evaluate the generated Chinese texts, in other text generation systems, automatic metrics BLEU-4 [18] and ROUGE-L [19] scores are commonly used to evaluate the quality of generated texts. To evaluate the quality of the generated Chinese texts, the corresponding reference Chinese sentences corresponding to several various types of sc-structures are built manually by trained native Chinese speakers and verified by others.

Currently, there is only Melbourne’s best WebNLG system for generating English texts, which is focused on generating English sentences from knowledge base fragments in form of RDF [20]. With the advent of the pre-training model, WebNLG provides a basic system implemented on pre-training model T5 for generating English texts [21]. Without other Chinese text generation systems to compare performance, therefore the performance of developed problem solver of Chinese text generation and other generation systems for English language in the same evaluation metrics BLEU-4 and ROUGE-L are shown in Table. IV and Table. V separately.

Table IV: Evaluation of efficiency for Chinese text generation

	BLEU-4	ROUGE-L
Problem solver of Chinese text generation	0.5885	0.6793

Table V: Evaluation of efficiency for generation systems for English language

	BLEU-4	ROUGE-L
T5-baseline	0.5520	0.6543
Melbourne	0.5452	0.6350

As can be seen from Table. IV and Table. V, although the generation systems for English language is oriented on generating English language texts from knowledge base fragments in form of RDF, with the help of the

combined use of neural network models and semantic models for generating Chinese texts, the developed problem solver achieved relatively promising BLEU-4 and ROUGE-L scores on Chinese texts generation. Moreover experimental results show that the developed problem solver is more suitable for generating Chinese texts when developing interface of ostis-systems.

V. CONCLUSION

This article had proposed a unified semantic model of natural language interface for intelligent system, oriented on conversion natural language texts into knowledge base fragments and generation natural language texts from knowledge base fragments within OSTIS Technology framework. The proposed semantic model of natural language interface mainly consists of sc-model of knowledge base of linguistics, in which the linguistic knowledge at various levels can be constructed, as well as sc-model of problem solvers, which have ability of deeply integrating logical models on rules and neural network models for natural language texts conversion and texts generation using multi-agent approach. Moreover on the basis of the unified semantic model of natural language interface the Chinese language interface of intelligent system in specific subject domains can be implemented with help of developed knowledge base on Chinese language processing and corresponding specific problem solvers for Chinese language processing. In order to verify the performance of the semantic model of natural language interface, we evaluated the developed Chinese language interface in three aspects. According to evaluated results the developed knowledge base on Chinese language processing has ability to integrate various linguistic knowledge for Chinese language processing. Compared to other systems (in these system factual knowledge is represented in form of RDF) for knowledge extraction and text generation, developed corresponding problem solvers could achieve relatively promising scores on specific metrics.

REFERENCES

- [1] Y. C. Liu Survey on Domain Knowledge Graph Research. *Computer Systems Applications*, 2020, vol. 26, No. 06, pp. 1–12.
- [2] L. W. Qian Ontological Approach to the development of natural language interface for intelligent computer systems. *Otkrytie semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, Minsk, 2022, pp. 217–238.
- [3] Q. Liu, Y. Li, H. Duan, Y. Liu, Z. G. Qin Knowledge Graph Construction Techniques. *Journal of Computer Research and Development*, 2016, vol. 53, No. 03, pp. 582–600.
- [4] Y. Yang, Z. Wu, Y. Yang, S. Lian, F. Guo, Z. Wang A Survey of Information Extraction Based on Deep Learning. *Journal of Software*, 2019, vol. 30, No. 06, pp. 1793–1818.
- [5] Y. Gao Review of Open Domain Information Extraction. *Modern Computer*, 2021, No. 07, pp. 103–110.
- [6] A. Fader, S. Soderland, O. Etzioni Identifying Relations for Open Information Extraction. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, John McIntyre Conference Centre, Edinburgh, 27-31 July 2011, pp. 1535–1545.

- [7] I. Androutsopoulos, G. Lampouras, D. Galanis Generating Natural Language Descriptions from OWL Ontologies: the NaturalOWL System. *Journal of Artificial Intelligence Research*, 2013, vol. 48, No. 01, pp. 671–715.
- [8] A. Gatt, E. Krahmer Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 2018, vol. 61, pp. 65–170.
- [9] K. Hu, X. F. Xi, Z. M. Cui, Y. Y. Zhou, Y. J. Qiu Survey of Deep Learning Table-to-Text Generation. *Journal of Frontiers of Computer Science and Technology*, 2022, vol. 16, No. 11, pp. 2487–2504.
- [10] G. Claire, S. Anastasia, N. Shashi The WebNLG Challenge: Generating Text from RDF Data. In Proceedings of the 10th International Conference on Natural Language Generation, Santiago de Compostela, Spain, 2017, pp. 124–133.
- [11] V. V. Golenkov, N. A. Gulyakina Proekt otkrytoi semanticheskoi tekhnologii komponentnogo proektirovaniya intellektual'nykh sistem. Chast' 1 Printsipy sozdaniya [Project of open semantic technology of component designing of intelligent systems. Part 1 Principles of creation]. *Ontologiya proektirovaniya [Ontology of designing]*, 2014, No. 1, pp. 42–64 (In Russ.).
- [12] M. E. Sadoski The structure of next-generation intelligent computer system interfaces. *Otkrytye semanticheskije tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, Minsk, 2022, pp. 199–208.
- [13] D. V. Shunkevich Hybrid problem solvers of intelligent computer systems of a new generation. *Otkrytye semanticheskije tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, Minsk, 2022, pp. 119–144.
- [14] L. W. Qian, W. Z. Li Ontological Approach for Generating Natural Language Texts from Knowledge Base. *Otkrytye semanticheskije tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, Minsk, 2021, pp. 159–168.
- [15] J. Li, H. Hu, X. Zhang, M. Li, L. Li, L. Xu Light Pre-Trained Chinese Language Model for NLP Tasks. In CCF International Conference on Natural Language Processing and Chinese Computing, Minsk, 14 October 2020, Springer, Cham. zhengzhou, 2020, pp. 567–578.
- [16] W. Z. Li, L. W. Qian Development of a problem solver for automatic answer verification in the intelligent tutoring systems. *Otkrytye semanticheskije tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, Minsk, 2021, pp. 169–178.
- [17] Y. H. Tseng, L. H. Lee Chinese Open Relation Extraction for Knowledge Acquisition. Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, 26–30 April 2014, pp. 12–16.
- [18] K. Papineni, S. Roukos, T. Ward, W. J. Zhu BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02), USA, 2002, pp. 311–318.
- [19] C. Y. Lin ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, Barcelona Spain, 2004, pp. 74–81.
- [20] B. D. Trisedya, J. Z. Qi, R. Zhang, W. Wang GTR-LSTM: A Triple Encoder for Sentence Generation from RDF Data. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne Australia, 2018, pp. 1627–1637.
- [21] M. Kale, A. Rastogi Text-to-Text Pre-Training for Data-to-Text Tasks. In Proceedings of the 13th International Conference on Natural Language Generation, Dublin Ireland, 2020, pp. 97–102.

Онтологический подход к разработке китайско-языкового интерфейса в интеллектуальных системах

Цянь Лунвэй

В статье рассматриваются существующие подходы к приобретению фактографических знаний из текстов естественного языка и генерации текстов естественного языка из фрагментов базы знаний (фактографических знаний), которые рассматриваются как две основные задачи, решаемые естественно-языковыми интерфейсами интеллектуальных систем в нашей работе. Был проведен анализ проблем, возникающих при разработке естественно-языкового интерфейса интеллектуальных систем, а также приобретении фактографических знаний из текстов естественного языка и генерации текстов естественного языка из фрагментов базы знаний в настоящее время.

В рамках технологии OSTIS был предложена разработка единой семантической модели естественно-языкового интерфейса интеллектуальных систем, которые в основном состоят из sc-модели базы знаний лингвистики и sc-модели соответствующих решателей задач для обработки естественного языка. Среди них в sc-модели базы знаний лингвистики позволяет объединение лингвистических знаний на различных уровнях, в sc-модели соответствующих решателей задач позволяет интеграция моделей на основе правил и моделей нейронных сетей для обработки естественного языка. Более того, на основе единой семантической модели естественно-языкового интерфейса был реализован китайско-языковой интерфейс ostis-систем и оценен разработанный китайско-языковой интерфейс по трём аспектам. По сравнению с другими системами, разработанный китайско-языковой интерфейс имеет лучшую эффективность.

Received 29.03.2023

Tools for Creating and Maintaining a Knowledge Base by Integrating Wolfram Mathematica System and Nevod Package

Valery B. Taranchuk
*Department of Computer
Applications and Systems
Belarusian State University*
Minsk, Republic of Belarus
taranchuk@bsu.by

Vladislav A. Savionok
*Department of Software
for Information Technologies
Belarusian State University
of Informatics and Radioelectronics*
Minsk, Republic of Belarus
v.savenok@bsuir.by

Abstract—One of the outcomes of the current review of the state of knowledge base design and analysis technologies, software and hardware platforms for the implementation of semantically compatible intelligent computer systems is the conclusion about the need to formulate the principles of collective design, development, verification of knowledge bases. Accordingly, it is important not only to formulate and justify the theory, to formalize the requirements, but also to develop tools to represent such formal theory in the form (format) of the knowledge base of the corresponding scientific knowledge portal. It is this concept that is the goal of implementation and development of the OSTIS Ecosystem, which, in particular, is intended to solve problems of convergence and merging of functional properties of systems of different classes; range expansion, organizational and technical unification, realization of co-ordination of software, computing and telecommunication means; unification of intelligent computer systems. A special place in such unification should be given to the solution of problems of integration of OSTIS Ecosystem tools with computer mathematics systems, especially with computer algebra systems.

This paper presents an example of integration of the local intelligent computer system based on Nevod library with the knowledge base of Wolfram Mathematica computer algebra system, which can be interpreted as an analogue of the integration of knowledge bases of the corporate OSTIS-system into the OSTIS Ecosystem. Examples of the use of tools to analyze the local knowledge base, its transfer from virtual to real status are presented and explained.

Keywords—Semantic analysis, OSTIS technology, Wolfram Mathematica, Wolfram Knowledgebase, Entity, temporal markers, Nevod

I. INTRODUCTION

According to the assessment of the current state of the field of artificial intelligence (AI), given in [1], there is an active development of many different areas, such as formal ontologies, artificial neural networks, machine learning, multi-agent systems, etc. However, this activity does not bring an aggregate increase in the level of intelligence of modern intelligent computer systems (ICS). This is due to the current isolation between methods and

designing tools in each of the areas of AI. The solution of this problem is seen in the construction of a general formal theory of ICS, and designing a comprehensive technology for their development and life cycle support — the OSTIS technology [1]. This will allow to achieve convergence of all areas of AI through their mutual integration and joint development.

Modern design support frameworks in the field of AI are mainly aimed at the development of highly specialized solutions, which can act as individual components of the ICS. In order to obtain guaranteed compatibility of all developed components, it is necessary to transform these tools into a unified technology for comprehensive design and support of the full life cycle of the ICS. Despite the independent development of the ICS, special attention should also be paid to their external interfaces, since the intercommunication of ICS between each other will be required as part of complex systems for the automation of various human activities. In other words, there is a need for unification and convergence of next-generation ICS, along with their components. This will open the way to the design of optimized complexes that include all the necessary AI to solve the tasks at hand. It is important to note that in order to meet the optimization requirements when solving certain tasks and achieve maximum performance, it is necessary to organize the effective interaction of the ICS with the connected information resources. The main actions for solving the key methodological problems, which are the reason for the current state of the field of AI, are also given in [1]. Note that similar problems are solved in the field of computer algebra systems: in their design, development, content update and functionality expansion [2], [3].

Methodological and technical solutions for integration of various types of knowledge implemented in the computer algebra system Wolfram Mathematica (WM), Wolfram Language (WL) are described in this paper.

The software solutions implemented in the Wolfram Knowledgebase are marked and illustrated with examples. From the standpoint of the need of unification of next-generation ICS, integration with the Wolfram Knowledgebase is performed on the general basis, meaning that the same method can be applied to integrate the WM with other knowledge systems, including the OSTIS Metasystem. The examples in this paper demonstrate several methods of integration of various tools implemented in Wolfram Mathematica computer algebra system, and by means of independent library Nevod [4].

II. CREATING A THEMATIC BLOCK FOR TEMPORAL MARKERS ANALYSIS

A. Temporal markers analysis

One of the main directions in the field of text processing is the extraction of their semantic component — semantic analysis. In this direction a number of problems are solved, such as document search in local and global networks, automatic annotation and abstracting, classification and clustering of documents, synthesis of texts and machine translation, text tone analysis, and fact extraction (publications mentioned in [5]).

An integral part of the task of extraction of facts and determination of relations between objects is localization in time of the event corresponding to the fact. The information, allowing to localize the event on a time axis, is transferred by means of various in the form and content textual expressions — temporal markers (pointers). The final result of the extraction of temporal markers from the text is their representation and interpretation within the framework of the formal model set in the process of semantic analysis [6].

To solve the problem of extracting temporal references from text the toolkit of one of the leaders in the field of entity recognition Microsoft.Recognizers.Text [7] is widely used.

B. Preparation of data for the thematic block of temporal markers analysis

The MS Recognizers Text (MRT) library provides the ability to recognize entities in texts of various languages and is widely used in Microsoft products, for example: in pre-defined templates for LUIS (Language Understanding Intelligent Service), in the platform for creating dialog bots Power Virtual Agents [8], in cognitive language services in Azure cloud infrastructure — NER (Named Entity Recognition). The library is distributed under an open source and free software license from MIT; along with the source code in the repository on GitHub [7] test dataset for different languages are published.

The Microsoft.Recognizers.Text.DateTime module, and in particular its BaseDateExtractor component, is used in MRT to search for temporal markers in the text. This component corresponds to a test dataset represented

in JSON format – the DateExtractor.json file [9]. The dataset contains 143 elements that include absolute and relative dates in different forms, as well as meta-information, which is used to check the correctness of the extraction results. A search context, a reference date that indicates the point in time used to translate relative temporal markers into absolute ones, can be attached to the dataset element. An example of a test dataset element with comments is shown in Fig. 1.

```
{
  "Input": "i will leave in 3 weeks", // - input text for search
  // search context:
  "Context": { "ReferenceDateTime": "2018-06-20T00:00:00" },
  "NotSupported": "python,javascript",
  "Results": [ // list of expected results:
    // each result includes text, type, start position and length
    { "Text": "in 3 weeks", "Type": "date", "Start": 13, "Length": 10 }
  ]
},
```

Figure 1. Example of a test dataset element.

In [5] the results of comparing the capabilities in temporal pointers extraction of MRT and Nevod library [4], which implements the search method in the text [10], are described. For this purpose two software modules were developed: mMRT and mNevod, which provide search and extraction of temporal markers from text. Comparative testing of the software modules was performed on the described test dataset using the means of the computer algebra system Wolfram Mathematica to analyze the results.

Input data for mNevod and mMRT modules is *Input* string. The results of temporal pointer extraction modules are compared with the *Results* dictionary, which contains the expected position in the text, length and contents of the extracted temporal pointer. The DateExtractor test dataset is used to confirm the functional completeness of the libraries that extract temporary pointers from text.

When checking and tuning fact extraction tools, in particular temporal pointers, an important position to evaluate is the focus on recognition rather than unambiguous identification of entities in the text. The original DateExtractor test dataset of the MRT library does not allow to fully analyze the functionality of corresponding tools of this type — it covers most variants of dates writing in English, includes common abbreviations, but does not take into account the possibility of distortion of the input text. It seems advisable to compile a new test dataset that takes this aspect into account when evaluating fact extraction tools. The methodology for forming a representative test dataset is outlined in [5].

C. Using Wolfram Mathematica to form a test dataset

Focusing on the tools for extracting temporal markers in the text, using fragments from DateExtractor, a new test dataset was prepared. In the resulting dataset of 141 items, distortions (errors) most typical for manual typing

are introduced, so that they affect the text fragments that represent the target for extraction.

The following types of situations have been selected as typical manual input errors that do not affect the word length:

- 1) replacing a single letter;
- 2) transposing a pair of neighboring letters in a word.

It should be noted that when modeling distortions of type 1, there is a natural heuristic to limit the set of letters that can be used incorrectly instead of a given correct letter. This is based on the assumption that the standard means of entering text data for computers, the keyboard, is used. In this case the most common substitutions will be the neighboring letters by the location of the keys on the keyboard. For example, for the word "Monday" one of the most frequent variants of such an error for the QWERTY layout is the replacement of the letter "d" by the letter "s" — "Monsay".

When modeling errors of type 2, a similar natural heuristic can be applied. Taking into account the blind printing method, it is logical to assume that most often a transposition error will occur for characters, which are typed by fingers of different hands [5]. For example, for the word "Sunday" a variant of such distortion in the QWERTY layout is the transposition of the letters "a" and "y" — "Sundya".

The listed types of input text distortions can be multiplied and combined: one word can contain several errors of the same type, or errors of several types simultaneously. In [5] the process of modeling each type of errors separately is described, without taking into account their combinations. According to each type of errors, the following distortions are included in the resulting set of 141 items:

- the letter "d" was replaced with the letter "s" in the word "monday" (corresponding to 2.8% of the set, the total word is contained in 3.5% of the set);
- the letters "a" and "y" in the word "sunday" have been rearranged (corresponding to 5.7% of the set, the total word is contained in 7% of the set).

When evaluating the correctness of the processing of the obtained dataset by mNevod and mMRT software modules, identical results were obtained: 91.4%. Due to the extensibility of templates in the Nevod package, rules were added to level out the corresponding erroneous situations. For letter substitution recognition, different variants of distortion of the word "monday" at the position corresponding to the letter "d" were introduced. A rule covering possible permutations of neighboring letters of the word "sunday" has been added. Taking into account the previously described heuristics, these are the variants "sundya" and "usnday". The software module using the Nevod library with the updated rule set correctly processed 100% of the test dataset. Thus, the use of the Nevod library tools allows to adapt the software module

for different variants of input data distortions by making local changes in the existing pattern sets. The application of the proposed heuristics when composing new rule sets will allow to implement the initial processing of typical input errors.

Wolfram Mathematica tools usage. To compare the functionality of the mNevod and mMRT modules when solving the problem of extracting temporal pointers in text not only from the DateExtractor test dataset, but also by forming other representative datasets, Wolfram Mathematica has developed the mDataWM service application. It contains software tools that enable you to separate the dataset to be processed from the meta-information, evaluate and compare the quality of the results of processing a modified dataset with mMRT and mNevod modules, distort any dataset, and test the performance of the libraries. The mDataWM application provides for creating test datasets in any language and analyzing the results of their processing. The tools of the mDataWM application implement the following functions:

- distort initial dataset and form a modified one;
- import/export to interface Mathematica with the mMRT and mNevod modules (handling files and separating data from meta-information);
- evaluate the quality of the results of dataset processing.

The following Mathematica kernel functions are used in mDataWM:

- *Import[source]* — imports data from *source*, returning a Wolfram Language representation of it.
- *Export[dest, expr, "format"]* — exports *data* in the specified format "*format*".
- *Map[f, expr]* — applies *f* to each element on the first level in *expr*.
- *MapIndexed[f, expr]* — applies *f* to the elements of *expr*, giving the part specification of each element as a second argument to *f*.
- *Association[key₁ -> val₁, key₂ -> val₂, ...]* — represents an association between keys and values.
- *AssociateTo[a, key -> val]* — changes the association *a* by adding the key-value pair *key -> val*.
- *SortBy[list, f]* — sorts the elements of *list* in the order defined by applying *f* to each of them.
- *KeyMemberQ[assoc, form]* — yields True if a key in the association *assoc* matches *form*, and False otherwise.
- *KeyDrop[assoc, {key₁, key₂, ...}]* — yields an association from which elements with keys *key_i* have been dropped.
- *KeyTake[assoc, {key₁, key₂, ...}]* — yields an association containing only the elements with keys *key_i*.
- *RandomSample[{e₁, e₂, ...}, n]* — gives a pseudo-random sample of *n* of the *e_i*.

- *Select*[*list*, *crit*] — picks out all elements e_i of list for which *crit*[e_i] is True.
- *Delete*[*expr*, *n*] — deletes the element at position *n* in *expr*. If *n* is negative, the position is counted from the end.
- *StringReplace*["*string*", *s* → *sp*] — replaces the string expression *s* by *sp* wherever it appears in "*string*".
- *Count*[*list*, *pattern*] — gives the number of elements in *list* that match *pattern*.

In the next examples, the original test dataset is extracted from WDR, and on its basis the correctness of temporal pointer extraction and target search pattern processing tools (mMRT tool based on MS Recognizers and mNevod tool based on Nevod) are tested. The results of the check can also be uploaded to WDR.

WDR supports the ability to work in parallel with multiple programs, nodes, clients, which allows you to organize a kind of (Mass Servicing System):

- one client uploads dataset items to the WDR;
- another client retrieves the set and runs the mNevod tool, uploading the results back into WDR;
- the third client reads the set in parallel and runs the mMRT tool, and uploads the results back into WDR.

D. Creation of a thematic block, inclusion of temporal markers analysis tools in the WDR

The upload of previously prepared data from other Information Resources into the WDR is implemented. The existing data can be modified on any computer using any tool or with Wolfram Mathematica (or WolframAlpha) toolsets. In particular, in the arsenal of tools from WM most often used functions are as follows: lists manipulation, imposition of various kinds of noise and distortions by random number generators.

WDR creation, data downloading, limitations of the free version. *CreateDatabin*[] creates a databin in the Wolfram Data Drop and returns the corresponding *Databin* object (*CreateDatabin*[*options*] creates a *databin* with the specified options). When creating a WDR, it is possible to pre-define the semantics of the data that will be contained in this databin [11]. As a result of executing the code

```
initialKb = CreateDatabin[];
```

the thematic block shown in Fig. 2 and Fig. 3 is created. When creating a new block, the system assigns an identifier to it. Using the identifier, basic operations on uploading, selecting and deleting data can be performed with the block. In our case, the obtained identifier is placed in the variable *initialKb*, with which we operate further.

The free version of WM has a number of restrictions on the use of WDRs. In particular, the size of one element

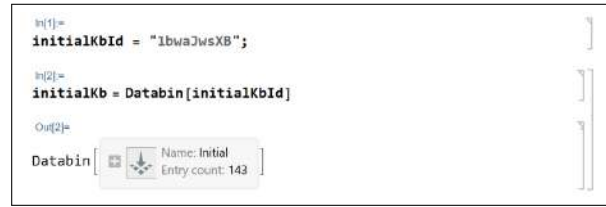


Figure 2. The result of creating a thematic block.

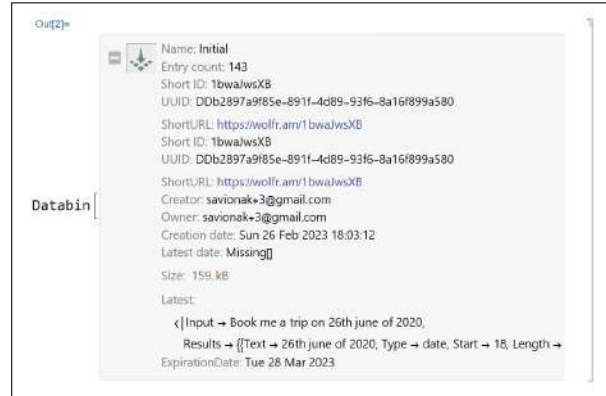


Figure 3. Detailed information about a thematic block.

in WDR cannot exceed 25 KBytes. Because of this, each element of the test dataset in the example is uploaded separately. Another restriction is applied to the frequency of uploads – no more than 60 per hour. To circumvent this limitation, the original test dataset is divided into parts of 60 or less items, and then each part is uploaded at one hour intervals. The code used to upload the test dataset into WDR is –

```
(* first batch of 60 elements: *)
specsToUpload = Take[specs, {1, 60}];
(* second batch of 60 elements: *)
specsToUpload = Take[specs, {61, 120}];
(* third batch of elements: *)
specsToUpload = Take[specs, {121, 142}];
(* upload batches, one hour interval: *)
DatabinUpload[initialKb, specsToUpload];
```

The *Take* function is used (*Take*[*list*, *n*] gives the first *n* elements of *list*; *Take*[*list*, $-n$] gives the last *n* elements of *list*; *Take*[*list*, {*m*, *n*}] gives elements *m* through *n* of *list*) to divide the dataset into parts. Each part is uploaded separately using the *DatabinUpload* function (*DatabinUpload*[*bin*, {*entry*₁, *entry*₂, ...}] bulk uploads all the entries *Subscript*[*entry*, *i*] to a *databin*; *DatabinUpload*[*bin*, *EventSeries*[...]] bulk uploads all entries in an event series to a *databin*).

In WM, it is possible to add a single item to the WDR – for this purpose the *DatabinAdd* function is used. In the following example, the last element is uploaded separately:

```
DatabinAdd[initialKb, specs[[143]]];
```


The above steps are given as one of the options that allow you to bypass the restrictions imposed, while maintaining the simplicity of retrieving the uploaded test dataset. Another option, which is more efficient in terms of storage and time spent on uploading the elements of the dataset, is to build a hierarchy. It was noticed, that each element in the previously described procedure does not exceed the size of 2 KBytes. Thus, by combining the elements in a two-level hierarchy, in batches of approximate size of 10, you can significantly reduce the upload time of the entire dataset. However, in this case, the extraction procedure becomes more complicated, because it is necessary to perform additional transformations and flatten the hierarchy, for example, using the *Flatten* function (*Flatten[list]* – flattens out nested lists; *Flatten[list, n]* – flattens to level *n*; *Flatten[list, n, h]* – flattens subexpressions with head *h*).

Extracting data from the WDR. Data extraction from the WDR is performed using the functions *Databin* (represents a databin in the Wolfram Data Drop) [12] and *Normal[expr]* (converts *expr* to a normal expression from a variety of special forms). An example of getting the full content of a thematic block is shown in Fig. 4. Examples of obtaining part of the content with a given element extraction step are shown in Fig. 5 and Fig. 6.

```

In[3]:=
initialKb = Databin[initialKbId]

Out[3]:=
Databin [Name: Initial
Entry count: 143]

In[4]:=
data = Normal[initialKb];
Length[data]

Out[5]:=
143

In[6]:=
data

Out[6]:=
{<Input -> i'll go back on 15,
Results -> {{Text -> 15, Type -> date, Start -> 16, Length -> 2}}>,
<Input -> i'll go back april 22, Results ->
{{Text -> april 22, Type -> date, Start -> 13, Length -> 8}}>,
<Input -> i'll go back jan-1, Results ->
{{Text -> jan-1, Type -> date, Start -> 13, Length -> 5}}>,
}

```

Figure 4. Extracting all the data from the thematic block.

E. Using WDR during verification of functional completeness of temporal markers extraction tools

Basic steps to check the functional completeness of temporal markers extraction tool [5] with WDR integration are as follows:

- 1) Retrieve the test dataset from the thematic block.
- 2) Start the tool to be tested (e.g. mNevod, mMRT).
- 3) Read the obtained extraction results.
- 4) Compare with the expected results from the meta-information of the test dataset.

```

In[7]:=
d7FirstKb = Databin[initialKbId, {1, 7, 1}]

Out[7]:=
Databin [Name: Initial
Total entry count: 143
Selection: entries 1 to 7 step 1]

In[8]:=
Normal[d7FirstKb]

Out[8]:=
{<Input -> i'll go back on 15,
Results -> {{Text -> 15, Type -> date, Start -> 16, Length -> 2}}>,
<Input -> i'll go back april 22, Results ->
{{Text -> april 22, Type -> date, Start -> 13, Length -> 8}}>,
<Input -> i'll go back jan-1, Results ->
{{Text -> jan-1, Type -> date, Start -> 13, Length -> 5}}>,
}

```

Figure 5. Extracting elements 1 through 7 from the thematic block.

```

In[9]:=
d7LastKb = Databin[initialKbId, -7]

Out[9]:=
Databin [Name: Initial
Total entry count: 143
Selection: latest 7 entries]

In[10]:=
Normal[d7LastKb]

Out[10]:=
{<Input -> the face amount of its 6 1/4% convertible...,
Comment -> 1/4 shouldn't recognized as date here,
Results -> {}>,
<Input -> i'll go back twenty second of june 2017,
NotSupported -> python, javascript,
Results -> {{Text -> twenty second of june 2017,
Type -> date, Start -> 13, Length -> 26}}>,
}

```

Figure 6. Extracting the last 7 elements from the thematic block.

5) Upload the results into the WDR.

An example of mNevod results is shown in Fig. 7. The form of representation is the same as that of the mMRT module: for each *Input* string, the module lists extracted temporal markers in the *Results* list in text and numeric form.

```

Out[22]:=
{<Input -> i'll go back on 15,
Results -> {{Start -> 16, Length -> 2, Text -> 15, Date -> 15.07.2022}}>,
<Input -> i'll go back april 22,
Results -> {{Start -> 13, Length -> 8, Text -> april 22, Date -> 22.04.2022}}>,
<Input -> i'll go back jan-1,
Results -> {{Start -> 13, Length -> 5, Text -> jan-1, Date -> 01.01.2022}}>,
<Input -> i'll go back jan/1,
Results -> {{Start -> 13, Length -> 5, Text -> jan/1, Date -> 01.01.2022}}>,
<Input -> i'll go back october. 2, Results ->
{{Start -> 13, Length -> 10, Text -> october. 2, Date -> 02.10.2022}}>,
}

```

Figure 7. Temporal markers extraction results by mNevod module.

It should be noted that Nevod library, due to its structure, opens an additional possibility to use WDR. Nevod is a multipurpose library designed to search for pattern matches in text. Patterns are defined independently from the library in a special language of their description, they allow to flexibly configure the search and extraction of entities from the text [13]. Previously, to solve the problem of extracting temporal pointers from text, a standard date search set from Nevod's library of basic

patterns was used. When testing the functional completeness, the disadvantages of this pattern set were revealed, it was supplemented and included as a component of the mNevod module. Taking into account independence of patterns from the library, it's expedient to place the received supplemented set of patterns in WDR. It will allow to make publicly available the actual version of the set, and at the same time will simplify the task of its subsequent correction.

III. CREATING A THEMATIC BLOCK FOR OSTIS CONFERENCE MATERIALS ANALYSIS

The purpose of this part of the work is to demonstrate the simplest WM tools to create, maintain, use Wolfram Data Repository to form a centralized repository and integration with any other platforms and systems.

Tools of sampling and placement in WDR of programs of last five OSTIS conferences, examples of intellectual analysis, in particular, selection of reports of the indicated authors are explained. It is essential, that (for the purpose of demonstration of possibilities of integration of means of different programs, packages, systems) the information processing and analysis are carried out by means of Nevod library, and also by means of WM tools.

A. Preparing material for analysis

The examples below illustrate the preprocessing and placement in WDR of information extracted from the programs of five most recent OSTIS conferences.

There are four steps involved in preparing the materials:

- 1) Extract materials from the conference website, according to the list of years, programs of meetings (plenary, breakout sessions, as well as exhibition and poster demonstrations) and generate PDF files "ostis-2018", "ostis-2019", "ostis-2020", "ostis-2021", "ostis-2022" with subsequent placement in the folder with the current WM notebook.
- 2) Convert received PDF files into plain text format.
- 3) Remove introductory explanations and summaries.
- 4) Upload prepared materials in WDR.

Step 1. Materials retrieval can be performed either manually or automatically using the WM tools for working with WWW resources, for example, described in [3].

Step 2. For each prepared file, its contents are loaded and converted into a text format for further processing. The Import function is used to specify the data format to which the file contents will be converted when loaded. To apply the function to more than one file, the /@ function, a shortened version of the Map function (*apply the function to each item in the list*), is used. The code for loading and converting files is the following:

```
dataDir = NotebookDirectory[];
files = FileNames["*.pdf", dataDir <> "pdf/"];
contents = Import[#, "Plaintext"]&/@files;
```

Step 3. The introductory explanations (introductions from organizing committee to conference schedule) and summaries (abstracts) are deleted. During this procedure, it is advisable to save the cover pages of the programs to extract the year of the conference later. The processing is done using the function *StringSplit["string", patt, n]* (*splits into substrings separated by delimiters matching the string expression patt, into at most n substrings*). The implementation uses the additional option to search for a case insensitive pattern (*IgnoreCase -> True*). The code for material cleaning is as follows:

```
getTitlePage = If[Length[#] > 1, #[[1]], ""]&
[StringSplit[#, "организационный комитет", 2,
IgnoreCase -> True]]&;
getMainBody = #[[Length[#]]]&
[StringSplit[#, "график работы конференции", 2,
IgnoreCase -> True]]&;
getClean = getTitlePage[#] <> getMainBody[#]&;
preparedContents = Map[getClean, contents];
```

The final *getClean* function is a composite of two other functions: *getTitlePage* extracts the title page and *getMainBody* extracts the main conference program text. The *getTitlePage* function is designed taking into account the fact, that some input documents were represented by the conference web-pages saved in PDF format, in which this page is missing. The *If* function is used to check the presence of the title page.

Step 4. To upload data into the WDR, WM's capabilities are applied. The creation of the thematic block using the *CreateDatabin* function is described above. At this stage peculiarities of work with text data, in particular with the Cyrillic alphabet were found out. In spite of the fact that each prepared document does not exceed the limit of 25KB per element, the final size of the loaded document increased several times and did not correspond to the specified limit. It turned out that the reason was the presence of Cyrillic letters in the materials. When converting the text to the Wolfram Language format, Cyrillic characters are represented in the wrong encoding, due to which the final size of the downloaded document increases many times. To get around this drawback, the resulting materials are compressed using WM *Compress* function. Accordingly, when extracting materials for analysis, the restoration of the original data is performed using the paired function *Uncompress*. Thus, the code of WDR creation, uploading and extraction of materials is the following:

```
initialDatabin = Databin[initialDatabinId];
DatabinUpload[initialDatabin, Compress/@
preparedContents];
downloadedContents = Uncompress/@Normal[
initialDatabin];
```

B. Examples of knowledge extraction, interpretation using the Nevod library

Preparing data for analysis by Nevod library — exporting to plain text files

The preprocessed documents of the conference programs extracted from the thematic block are saved to text files using the *Export* function. File names are generated as integers in order using the *Range* function. The path to the files is specified in the data folder in the subfolder "txt":

```
plainTextFiles = FileNameJoin[{dataDir <<
  ".txt", ToString[#] << ".txt"}]&/@Range[
  Length[downloadedContents]];
```

Then the *Export* function is applied to each element of the list using the *MapIndexed* function. Along with the contents of the element (parameter #1) it passes the sequence number of the element in the list (parameter #2 – list of one number element):

```
MapIndexed[Export[plainTextFiles[[#2[[1]]], #1]&,
  downloadedContents];
```

Preparing to launch the Nevod library

To integrate with the Nevod library, the Nevod utility module was developed. Integration is performed through intermediate files. Mandatory parameters are passed to the input of the module in the following order:

- 1) path to the file with search patterns;
- 2) path to the file to perform search in;
- 3) path to the output file for writing results in JSON format.

In the example, reports of the author whose name is specified in the template or query are selected and prepared for subsequent output. The files with search patterns for the two authors are in the data folder.

The following code generates the full paths to the files with the patterns to search the reports of the authors "Таранчук" (*patterns-taranchuk.np*) and "Головко" (*patterns-golovko.np*):

```
taranchukPatternsPath = dataDir <<
  "patterns - taranchuk.np";
golovkoPatternsPath = dataDir <<
  "patterns - golovko.np";
```

Fig. 8 shows the contents of the *patterns-taranchuk.np* file to search for the reports of the author "Таранчук". The main patterns whose matches are returned as results are "ЦелевойДоклад". (extracts the list of reports by the searched author with the report topic, full list of authors, and time) and "ГодПроведенияКонференции" (extracts the year of the conference). Other templates are internal and describe the constructs for extracting the target author ("ЦелевойАвтор"), time range ("Диапазон"), authors list ("Авторы"), and report list ("Доклад").

To start the Nevod module *RunProcess* function is used, particularly, its variant, which allows to pass a list of startup arguments. The path to the module to be launched is saved in *nevodPath*. File names for saving Nevod module results are generated beforehand from input file names by replacing the extension from "txt" to "json" and placing them in a different folder:

```
ЦелевойАвтор = "Таранчук";

#ЦелевойДоклад(Время, Тема, Авторы) = Доклад(Время:Время, Тема:Тема,
Авторы:Авторы1) @having ЦелевойАвтор;

#ГодПроведенияКонференции = Num @inside {"Программа" - "OSTIS-" + Num,
"Минск" - "БГУИР" - Num};

Время = Num(1-2) + ":" + Num(1-2);
Диапазон = Время - "-" - Время;
НеВремяДоклада = Диапазон - {"регистрация", "открытие", "заседание",
"совещание", "перерыв", "обед", "съезд"};
ВремяДоклада = Диапазон @outside НеВремяДоклада;

Автор = Alpha(TitleCase) + [0+ "-" + Alpha] + [1-2 [0+ WordBreak] +
Alpha(1, Uppercase) + ?"."];
Авторы = Автор + [0+ ", " - Автор];

Доклад(Время, Тема, Авторы1) = Время:ВремяДоклада .. Тема ..
Авторы1:Авторы;
```

Figure 8. Nevod patterns to search for the target author's reports.

```
fileBaseNames = FileBaseName/@FileNames[
  plainTextFiles];
resultTaranchukFileNames = FileNameJoin[{
  dataDir << "/json", # << " - taranchuk" <<
  ".json"}]&/@fileBaseNames;
resultGolovkoFileNames = FileNameJoin[{
  dataDir << "/json", # << " - golovko" <<
  ".json"}]&/@fileBaseNames;
```

For example, for the input text file "I.txt" the names of the output files with the search results will be (taking into account relative paths) ".json/I-taranchuk.json" and ".json/I-golovko.json" respectively.

The *MapIndexed* function runs the Nevod module separately for each input file and places the results in the corresponding output file from the *resultFileNames* list:

```
(* Таранчук *)
MapIndexed[RunProcess[{nevodPath,
  taranchukPatternsPath, #1,
  resultTaranchukFileNames[[#2[[1]]]]]&,
  plainTextFiles];
```

```
(* Головко *)
MapIndexed[RunProcess[{nevodPath,
  golovkoPatternsPath, #1,
  resultGolovkoFileNames[[#2[[1]]]]]&,
  plainTextFiles]
```

```
{<|ExitCode->0,StandardOutput->,StandardError-
->|>,<|ExitCode->0,StandardOutput->,StandardError-
->|>,<|ExitCode->0,StandardOutput->,StandardError-
->|>,<|ExitCode->0,StandardOutput->,StandardError-
->|>,<|ExitCode->0,StandardOutput->,StandardError->|>}
```

The presence of the value *ExitCode* – > 0 in the results of the *RunProcess* function indicates that the run of the module for each input file was successful.

Output the extraction results for the specified authors. The search results for each of the authors searched by Nevod are saved in JSON format. To read these files, *Import* function with this format is used. The function is applied to each file, which allows to get a list of results.

```
taranchukResultsNV = Import[#, "JSON"]&/@
resultTaranchukFileNames;
golovkoResultsNV = Import[#, "JSON"]&/@
resultGolovkoFileNames;
```

The results obtained in JSON format have a specific structure, an example of which (with line feeds saved) is shown in Fig. 9.

```
Out[31]=
{[{"ГодПроведенияКонференции" -> [{"text" -> 2018, "extractions" -> {}}],
{"ЦелевойДоклад" -> [{"text" -> 11:15-
11:40
ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ НЕЙРОННЫХ
СЕТЕЙ В АНАЛИЗЕ ГЕОДАНЫХ
Таранчук В. Б., "extractions" -> {"Время" -> {"11:15-
11:40"}, "Авторы" -> {"Таранчук В. Б."}, "Тема" -> {"
ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ НЕЙРОННЫХ
СЕТЕЙ В АНАЛИЗЕ ГЕОДАНЫХ
}}]}],
```

Figure 9. Example of Nevod module results, initial structure.

In subsequent processing with WM tools, the results are reduced to a flat representation, eliminating redundant line feeds. The functions for converting to such a representation are given below:

```
getYearNV = "text"/.
("ГодПроведенияКонференции"/.#)[[1]]&;
getReportTitleNV = StringReplace["\r\n" -> " "]
[("Тема"/.( "extractions"/.#)[[1]]]&;
getReportTimeNV = StringReplace["\r\n" -> " "]
[("Время"/.( "extractions"/.#)[[1]]]&;
getReportDetailsNV = < |
"Time" -> getReportTimeNV[#],
"Title" -> getReportTitleNV[#] > &;
getReportListNV = getReportDetailsNV/@
("ЦелевойДоклад"/.#)&;
getTargetReportsNV = If[KeyExistsQ[#,
"ЦелевойДоклад"], getReportListNV[#, {}]&;
getResultsNV = < |
getYearNV[#] -> getTargetReportsNV[#] >
&;
```

The code for getting the final results is shown below:

```
taranchukResults = getResultsNV/@
taranchukResultsNV
golovkoResults = getResultsNV/@
golovkoResultsNV.
```

The final results for each of the authors are shown in Fig. 10 and Fig. 11.

The results show that author Таранчук had no reports in 2018, two reports in 2019, one each in 2020 and 2021, and two in 2022; author Головко published in each of the five years listed.

C. Examples of knowledge extraction and interpretation using Wolfram Mathematica tools

Extracting and processing information from thematic blocks is possible with Mathematica tools [14], some of which were listed above. It should be added that any kernel and application package functions and proprietary program modules can be used. The above examples can

```
Out[32]=
{< | 2018 -> {} >,
< | 2019 -> {< | Time -> 11:15- 11:40, Title -> ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ
НЕЙРОННЫХ СЕТЕЙ В АНАЛИЗЕ ГЕОДАНЫХ > | >,
< | Time -> 11:10- 11:30, Title -> ИНФОРМАЦИОННЫЙ ПОИСК
И МАШИННЫЙ ПЕРЕВОД В РЕШЕНИИ ЗАДАЧИ
АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ЗАИМСТВОВАННЫХ
ФРАГМЕНТОВ ТЕКСТОВЫХ ДОКУМЕНТОВ > | > | >,
< | 2020 -> {< | Time -> 11:00- 11:20, Title ->
Примеры интеллектуальной адаптации цифровых полей
средствами системы ГеоБаза Данных > | > | >,
< | 2021 -> {< | Time -> 10:00- 10:30, Title ->
Интерактивные и интеллектуальные
средства системы ГеоБазаДанных > | > | >,
< | 2022 -> {< | Time -> 10:00- 10:30, Title ->
Проблемы и перспективы автоматизации
различных видов и областей человеческой
деятельности с помощью интеллектуальных
компьютерных систем нового поколения > | > | >,
< | Time -> 11:30- 12:00, Title -> Интеграция инструментов
компьютерной алгебры в приложения OSTIS > | > | > | >}
```

Figure 10. Flat results for the author Таранчук.

```
Out[32]=
{< | 2018 -> {< | Time -> 14:00- 14:20,
Title -> ИНТЕГРАЦИЯ НЕЙРОННЫХ СЕТЕЙ С БАЗАМИ ЗНАНИЙ > | >,
< | Time -> 14:20- 14:40, Title ->
ПРОЕКТИРОВАНИЕ ПРЕДПРИЯТИЯ РЕЦЕПТУРНОГО ПРОИЗВОДСТВА
В КОНТЕКСТЕ НАПРАВЛЕНИЯ INDUSTRY 4.0 > | > | >,
< | 2019 -> {< | Time -> 13:15- 13:45, Title ->
ПРИНЦИПЫ ПОСТРОЕНИЯ СИСТЕМ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ
ИНТЕГРАЦИИ НЕЙРОСЕТЕВЫХ И СЕМАНТИЧЕСКИХ
МОДЕЛЕЙ > | >, < | Time -> 14:00- 14:25,
Title -> ПРИНЦИПЫ ПОСТРОЕНИЯ СИСТЕМЫ КОМПЛЕКСНОГО
ИНФОРМАЦИОННОГО ОБСЛУЖИВАНИЯ СОТРУДНИКОВ
ПРЕДПРИЯТИЯ РЕЦЕПТУРНОГО ПРОИЗВОДСТВА > | > | >,
< | 2020 -> {< | Time -> 15:20- 15:40, Title ->
Реализация интеллектуальной системы поддержки принятия
решений для сопровождения производственного
процесса > | >, < | Time -> 15:40- 16:00,
Title -> Проблемы развития интеллектуальной робототехники
на предприятиях в контексте Industry 4.0 > | > | >,
< | 2021 -> {< | Time -> 10:30- 11:00, Title ->
Методологические проблемы современного состояния работ
в области Искусственного интеллекта > | > | >,
< | 2022 -> {< | Time -> 17:45- 18:15, Title ->
Конвергенция и интеграция искусственных нейронных
сетей с базами знаний в интеллектуальных
компьютерных системах нового поколения > | > | > | >}
```

Figure 11. Flat results for the author Головко.

be repeated with string templates, list manipulations, rearranging text phrases, and layouts. Below are a few examples and the codes for getting the results with these WM tools.

Extracting the year of the conference. To extract the conference year, *StringCases* function is used to extract patterns from strings. "WhitespaceCharacter" (including string translation), "DigitCharacter", "~" - strict following, "x..." - repeat one or more times, "a|b" - alternative choice between a and b. Additional work with nested lists: *Flatten* - flatten nested lists, *Part* - get a part of the list. Parameter *IgnoreCase* allows you to match with the template without taking into account the upper or lower case of the string. Extracting

the year from the title page of the program or in the first footer:

```
getYearWM = Part[#, 1]&@* Flatten@*
StringCases[DigitCharacter..]@* StringCases[
("Программа"~~ WhitespaceCharacter.. ~~
"OSTIS – "~~ DigitCharacter..)
|("Минск"~~ WhitespaceCharacter.. ~~
"БГУИР"~~ WhitespaceCharacter.. ~~
DigitCharacter..), IgnoreCase-> True];
```

Extracting one participant of the conference.

The "LetterCharacter" pattern describes a letter character. The first pattern below checks that the last name starts with a capital letter using the UpperCaseQ function. Longest pattern allows to select the longest match. It should be noted that in this particular case, the order of the alternatives is important. First is the longer one, second is the shorter one, since the second one is a superset of the first one; if written in reverse order, it returns the shortest match.

```
upperLetter = LetterCharacter?UpperCaseQ;
authorPattern = upperLetter ~~
LetterCharacter.. ~~ WhitespaceCharacter.. ~~
Longest[(upperLetter ~~ "."~~
WhitespaceCharacter... ~~ upperLetter ~~ ".")
|(upperLetter ~~ ".")];
```

Extracting several participants. Template "..." is used to repeat a particular part zero or more times.

```
multipleAuthorsPattern = Longest[
authorPattern ~~ WhitespaceCharacter... ~~
(","~~ WhitespaceCharacter.. ~~
authorPattern)...];
```

Extracting a time range.

```
timePattern = DigitCharacter ~~
DigitCharacter ~~ ":"~~ DigitCharacter ~~
DigitCharacter;
timeIntervalPattern = timePattern ~~ "-"~~
(WhitespaceCharacter...)~~ timePattern;
```

Extracting elements of the program schedule, except reports.

```
getNotReports = StringCases[
timeIntervalPattern.. ~~ WhitespaceCharacter..
~~"регистрация"|"открытие"|"заседание"|
"совещание"|"перерыв"|"обед"|"съезд",
IgnoreCase-> True];
```

Removing all the elements of the program schedule, except reports.

```
getTrimmedText = StringReplace[#,
getNotReports[#]-> ""]&;
```

Extracting a list of all reports.

```
getReports = StringCases[Shortest[
timeIntervalPattern ~~ __~~
multipleAuthorsPattern]]@* getTrimmedText;
```

Extracting reports with the specified author. This extraction is represented by a configurable function. The input is a string parameter, specifying the author's surname. The result is a new function that filters the list of reports in search of reports of the specified author.

```
getReportsWithAuthor = Function[{author},
Select[#, StringContainsQ[StringCases[#,
multipleAuthorsPattern][[1]], author]&]&;
```

Extracting report details: time and title.

```
getReportTime = StringCases[{
timeIntervalPattern, "\n" -> " "};
getReportTitle = StringReplace[{
(timeIntervalPattern|
multipleAuthorsPattern)-> "", "\n" -> " "};
getReportDetailsWM =< |
"Time" -> getReportTime[#[[1]],
"Title" -> getReportTitle[#[[1]]] > &;
```

Final function applied to the input file.

```
getResultsWM = Function[{author},
< |getYear[#]-> getReportDetailsWM/@
getReportsWithAuthor[author][getReports[#]]
|> &;
```

Extraction results for the specified author.

Extraction result for the author Таранчук is shown in Fig. 12, obtained with the following code:

```
taranchukResults2 = getResultsWM["Таранчук"]/@
downloadedContents.
```

```
{<|2018->{ }>, <|2019->{ <|Time->11:15- 11:40,Title-> ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ НЕЙРОННЫХ СЕТЕЙ В АНАЛИЗЕ ГЕОДАНЫХ >, <|Time->11:10- 11:30,Title-> ИНФОРМАЦИОННЫЙ ПОИСК И МАШИННЫЙ ПЕРЕВОД В РЕШЕНИИ ЗАДАЧИ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ЗАИМСТВОВАННЫХ ФРАГМЕНТОВ ТЕКСТОВЫХ ДОКУМЕНТОВ >}>, <|2020->{ <|Time->11:00- 11:20,Title-> Примеры интеллектуальной адаптации цифровых полей средствами системы ГеоБаза Данных >}>, <|2021->{ <|Time->10:00- 10:30,Title-> Интерактивные и интеллектуальные средства системы ГеоБазаДанных >}>, <|2022->{ <|Time->10:00- 10:30,Title-> Проблемы и перспективы автоматизации различных видов и областей человеческой деятельности с помощью интеллектуальных компьютерных систем нового поколения >, <|Time->11:30- 12:00,Title-> Интеграция инструментов компьютерной алгебры в приложения OSTIS >}>}
```

Figure 12. Extraction results for the author Таранчук.

IV. CONCLUSION

As an analogue of the integration of knowledge bases of the corporate OSTIS-system into the OSTIS Ecosystem, the example of integration of the local intelligent computer system based on Nevod library with the knowledge base of Wolfram Mathematica computer algebra system (Wolfram Data Repository) is presented.

The possibilities of working with Wolfram Data Repository are described by the example of creating a

thematic block of temporal markers analysis. In particular, functions for creating a thematic block, data upload and extraction from Wolfram Data Repository with the given sampling parameters are shown. The methodology for checking the functional completeness of temporal markers extraction tools from text, with focus on recognition rather than unambiguous identification, is described. This methodology is supplemented by the possibility to publish a thematic block in the public domain in order to maintain the current state of the test dataset.

The process of analyzing OSTIS conference materials for finding the reports written by a specific author is described in detail. Preparation of materials and their uploading in Wolfram Data Repository carried out by means of Wolfram Mathematica. For the purpose of demonstration of possibilities of integration with other platforms, systems, extraction and interpretation of knowledge are made independently by means of Nevod library, and by means of Wolfram Mathematica. Usage of Wolfram Mathematica tools for text manipulation and pattern search is described in detail. A step-by-step explanation of a final extraction pattern construction, as well as a brief description of main functions used to build it, is provided. The knowledge extraction results, produced by Nevod and by Wolfram Mathematica tools, are shown and uploaded to Wolfram Data Repository.

REFERENCES

- [1] V. V. Golenkov, N. A. Guliakina, D. V. Shunkevich Otkrytaja tehnologija ontologicheskogo proektirovanija, proizvodstva i jekspluatacii semanticheski sovместimyh gibridnyh intellektual'nyh komp'juternyh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems], Bestprint, 2021, P. 690
- [2] Wolfram Mathematica: Modern Technical Computing. Available at: <https://www.wolfram.com/mathematica> (accessed 2023, Feb).
- [3] V. B. Taranchuk Integration of computer algebra tools into OSTIS applications. *Open semantic technologies for intelligent systems*, 2022, no 6, pp. 369-374.
- [4] Nevod is a language and technology for pattern-based text search. Available at: <https://github.com/nezaboodka/nevod> (accessed 2022, Apr).
- [5] V. A. Savenok, V. B. Taranchuk Vozmozhnosti i sredstva biblioteki Nevod pri reshenii zadach izvlecheniya vremennykh ukazatelei v tekste [Features and tools of the Nevod library in solving problems of extracting temporal markers in the text]. *Problemy fiziki, matematiki i tekhniki [Problems of Physics, Mathematics and Technics]*, 2022, no 4, pp. 84-92, https://doi.org/10.54341/20778708_2022_4_53_84.
- [6] E. A. Suleimanova Semanticheskii analiz kontekstnykh dat [Semantic analysis of contextual dates]. *Programmnye sistemy: teoriya i prilozheniya [Program systems: theory and applications]*, 2015, vol. 6, no 4, pp. 367-399.
- [7] Microsoft.Recognizers.Text provides recognition and resolution of numbers, units, and date/time expressed in multiple languages. Available at: <https://github.com/microsoft/Recognizers-Text> (accessed 2022, Apr).
- [8] Intelligent Virtual Agents and Bots | Microsoft Power Virtual Agents. Available at: <https://powervirtualagents.microsoft.com/en-us> (accessed 2022, Apr).
- [9] Recognizers Test Cases Specs for Date Extractor. Available at: <https://github.com/microsoft/Recognizers-Text/blob/master/Specs/DateTime/English/DateExtractor.json> (accessed 2022, Apr).

- [10] D. A. Surkov, I. V. Shimko, V. A. Savenok et al. Sposob poiska v tekste sovpadenii s shablonami : pat. 037156, Belarus, MPK G06F 17/27, G06F 17/24, Evraziiskaya patentnaya organizatsiya, 2021, byul. no 2.
- [11] Data Semantics | Wolfram Datadrop Quick Reference. Available at: <https://www.wolfram.com/datadrop/quick-reference/data-semantics> (accessed 2023, Feb).
- [12] Databin — Wolfram Language Documentation. Available at: <https://reference.wolfram.com/language/ref/Databin.html> (accessed 2023, Feb).
- [13] Nevod Basic Patterns. Available at: <https://github.com/nezaboodka/nevod-patterns> (accessed 2023, Feb).
- [14] Working with String Patterns — Wolfram Language Documentation. Available at: <https://reference.wolfram.com/language/tutorial/WorkingWithStringPatterns.html> (accessed 2023, Feb).

Инструменты создания и сопровождения базы знаний путем интеграции системы Wolfram Mathematica и пакета Nevod

Таранчук В. Б., Савёнок В. А.

Одним из итогов текущего рассмотрения состояния технологий проектирования и анализа баз знаний, программных и аппаратных платформ реализации семантически совместимых интеллектуальных компьютерных систем является заключение о необходимости формулировки принципов коллективного проектирования, разработки, верификации баз знаний. Соответственно, важно не только сформулировать и обосновать теорию, формализовать требования, но и разработать инструменты представления такой формальной теории в виде (формате) базы знаний соответствующего портала научных знаний. Именно такая концепция является целью реализации и развития Экосистемы OSTIS, которая, в частности, предназначена для решения задач сближения и слияния функциональных свойств систем различных классов: расширения спектра, организационно-технического объединения, осуществления координации программных, вычислительных и телекоммуникационных средств; унификации интеллектуальных компьютерных систем. Особое место в таком объединении следует отвести решению вопросов интеграции средств Экосистемы OSTIS с системами компьютерной математики, особенно с системами компьютерной алгебры.

В данной работе приведен пример интеграции локальной интеллектуальной компьютерной системы на основе библиотеки Nevod с базой знаний системы компьютерной алгебры Wolfram Mathematica, что можно интерпретировать как аналог действий по локализации баз знаний корпоративной ostis-системы в состав Экосистемы OSTIS. Представлены и поясняются примеры использования инструментов анализа локальной базы знаний, ее перевода из статуса виртуальной в статус реальной.

Received 25.03.2023

Control Tools for Reusable Components of Intelligent Computer Systems of a New Generation

Maksim Orlov

*Belarusian State University of
Informatics and Radioelectronics*

Minsk, Belarus

Email: orlovmaksimkonst@gmail.com

Abstract—In the article, an approach to the design of intelligent systems is considered, focused on the use of compatible reusable components, which significantly reduces the complexity of developing such systems. The key means of supporting the component design of intelligent computer systems is the manager of reusable components proposed in the work.

Keywords—Component design of intelligent computer systems; reusable semantically compatible components; knowledge-driven systems; semantic networks.

I. INTRODUCTION

The main result of artificial intelligence is not the intelligent systems themselves but powerful and effective technologies for their development. The analysis of modern technologies for designing intelligent computer systems shows that along with very impressive achievements, the following serious problems occur [1], [2], [3]:

- high requirements for the initial qualifications of users and developers. Artificial intelligence technologies are not focused on the wide range of developers and users of intelligent systems and, therefore, have not received mass distribution;
- there is no general unified solution to the problem of semantic compatibility of computer systems [4]. There are no approaches that allow integrating scientific and practical results in the field of artificial intelligence, which generates a high degree of duplication of results and a lot of non-unified formats for representation of data, models, methods, tools, and platforms;
- lack of powerful tools for designing intelligent computer systems, including intelligent training subsystems, subsystems for collective design of computer systems and their components, subsystems for verification and analysis of computer systems, subsystems for component design of computer systems;
- long terms of development of intelligent computer systems and high level of complexity of their maintenance and extension;

- the degree of dependence of artificial intelligence technologies on the platforms on which they are implemented is high, which leads to significant changes in technologies when transitioning to new platforms;
- the degree of dependence of artificial intelligence technologies on subject domains in which these technologies are used is high;
- there is a high degree of dependence of intelligent computer systems and their components on each other; the lack of their automatic synchronization. The absence of self-sufficiency of systems and components, their ability to operate separately from each other without loss of expediency of their use;
- increase in the time to solve the problem with the expansion of the functionality of the problem solver and with the expansion of the knowledge base of the system [5];
- lack of methods for designing intelligent computer systems. Updating computer systems often boils down to the development of various kinds of “patches”, which eliminate not causes of the identified disadvantages of updated computer systems but only some of the consequences of these causes;
- poor adaptability of modern computers to the effective implementation of even existing knowledge representation models and models for solving problems that are difficult to formalize, which requires the development of fundamentally new computers [6];
- there is no single approach to the allocation of reusable components and the formation of libraries of such components, which leads to a high complexity of reuse and integration of previously developed components in new computer systems.

To solve these problems, it is necessary to implement a comprehensive technology for designing intelligent computer systems, which includes the following components:

- a model of an intelligent computer system [7];

- a *library of reusable components* and corresponding *tools to support component design of intelligent computer systems*;
- an intelligent integrated automation system for the collective design of intelligent computer systems, including subsystems for editing, debugging, performance evaluation, and visualization of developed components, as well as a simulation subsystem;
- methods of designing intelligent computer systems;
- an intelligent user interface;
- training subsystems for designing intelligent computer systems, including a subsystem for conducting a dialog with the developer and the user;
- a subsystem for testing and verification of intelligent computer systems, including a subsystem for testing the compatibility of the developed system with other systems;
- an information security support subsystem for the intelligent computer system.

The key component of the technology for intelligent systems design is a *library of reusable components* and the corresponding *tools for supporting component design of intelligent computer systems*. With its help, it is possible to effectively implement the typical subsystems to support the design of intelligent computer systems.

Most of the existing systems are created as self-contained software products that cannot be used as components of other systems. It is necessary to use either the whole system or nothing. A small number of systems support a component-oriented architecture capable of integrating with other systems [8], [9]. However, their integration is possible if the same technologies are used and only when designed by one development team [10].

Repeated re-development of existing technical solutions is conditioned either by the fact that known technical solutions are hardly integrated into the system being developed or by the fact that these technical solutions are difficult to be found. This problem is relevant both in general in the field of computer systems development and in the field of knowledge-based systems development, since in systems of this kind the degree of consistency of various knowledge types affects the ability of the system to solve non-trivial problems.

The development technology should allow components to be reused, integrated with other components built using both this and other technologies. It should also be open to allow using components by different development teams.

Reuse of ready-made components is widely used in many fields related to the design of various kinds of systems, since it reduces the complexity of development and its cost (by minimizing the amount of labor due to the absence of the need to develop any component), improves the quality of the systems being created, and reduces professional requirements for computer system developers [11]. Thus, there is a transition from programming

components or entire systems to their design (assembly) based on ready-made components. *Component design of intelligent computer systems* involves the selection of existing components capable of solving the problem in its entirety or the decomposition of the problem into subproblems with the allocation of components for each of them (see [12]). The designed systems according to the proposed technology have a high level of flexibility, their development is carried out in stages, moving from one complete version of the system to another. At the same time, the starting version of the system can be the core of the corresponding class of systems included in the *library of reusable components*. The technology of component design of intelligent computer systems includes a set of coordinated particular technologies that ensure the comprehensive design of computer systems. It includes the technology of component design of knowledge bases, problem solvers, interfaces, and others.

The main element of the semantic technology for component design of intelligent systems is the *library of compatible reusable components*. This allows designing intelligent systems by combining existing components, selecting the right ones from the appropriate libraries. The use of ready-made components assumes that the distributed component is verified and documented, and possible errors and limitations are eliminated or specified and known. The creation of the *library of reusable components* does not mean the re-creation of all existing modern information technology products. The technology of component design of intelligent computer systems involves the use of vast experience in the development of modern computer systems, however, it is required to create a specification of each component (both newly created and existing integrated ones) to ensure its installation and compatibility with other components and systems. Nevertheless, an effective component design technology will appear only when a “critical mass” of application system developers participating in the seeding of *libraries of reusable components* of the designed systems is formed.

The problems of implementing the component approach to the design of intelligent computer systems inherit the problems of modern technologies for designing intelligent systems and also include the following ones [13]:

- incompatibility of components developed within different projects due to the lack of unification in the principles of representing different types of knowledge within the same *knowledge base*, and, as a consequence, the lack of unification in the principles of allocation and *specification of reusable components*;
- inability to automatically integrate components into the system without manual user intervention;
- automatic updating of components leads to inconsistency of both particular modules of computer systems and the systems themselves with each other;

- lack of classification of components at different levels of detail;
- testing, verification, and analysis of the components quality are not carried out; advantages, disadvantages, limitations of components are not identified;
- development of standards that ensure the compatibility of these components is not being carried out;
- many components use the language of the developer for identification (usually English), and it is assumed that all users will use the same language. However, for many applications, this is unacceptable – identifiers that are understandable only to the developer should be hidden from end users, who should be able to choose the language for the identifiers they see;
- lack of tools to search for components that meet the specified criteria.

Component design of intelligent computer systems is possible only if the selection of components is carried out on the basis of a thorough analysis of the quality of these components. One of the most important criteria for such an analysis is the level of semantic compatibility of the analyzed components with all the components available in the current version of the library.

In addition to a powerful library of reusable semantically compatible components, an appropriate tool is needed for managing (installing into child systems, searching, updating, forming) such components. Such a tool should be built according to the same principles as intelligent computer systems of a new generation to ensure semantic compatibility of intelligent systems, their components, and their design tools.

The purpose of the work is to create a tool to support the component design of intelligent computer systems of a new generation. Such a tool is necessary to use the full potential of the infinitely extensible comprehensive *libraries of reusable components*. The fields where the technology of component design of semantically compatible intelligent systems is applied in practice have no limits.

II. ANALYSIS OF EXISTING APPROACHES TO SOLVING THE PROBLEM

At the moment, there is no comprehensive library of reusable semantically compatible components of computer systems in general, aside from intelligent ones. There are some attempts to create libraries of typical methods and programs for traditional computer systems, but such libraries do not solve the above problems.

The term “library of subprograms” was one of the first mentioned by M. Wilkes, D. Wheeler, and S. Gill as one of the forms for organizing calculations on a computer (considered in [14]). Based on what is stated in their book, a library is understood as a set of “short, pre-prepared programs for certain, frequently occurring

(standard) computing operations”. It is worth noting that the components of libraries are not only programs but also components of interfaces and knowledge bases.

Traditional solutions include package managers of programming languages and operating systems, as well as separate systems and platforms with built-in components and tools for saving created components.

Library components can be implemented in different programming languages (which leads to the fact that for each programming language, its own libraries are developed with their own solutions to various frequently occurring situations) and can also be located in different places, which leads to the fact that a tool is needed in the library to search for components and install them.

Modern package managers such as *npm*, *pip*, *apt*, *maven*, *poetry*, and others have the advantage of being able to resolve conflicts when installing dependent components, but they do not take into account the semantics of components and only install components by ID [15]. Libraries of such components are only a storage of components, which does not take into account the purpose of components, their advantages and disadvantages, scope of application, hierarchy of components, and other information necessary for the intellectualization of component design of computer systems. Searching for components in *libraries of components* corresponding to these package managers is reduced to searching by component ID. Modern package managers are only “installers” without automatic integration of components into the system. Similarly, a significant disadvantage of the modern approach is the platform dependency of components. Modern component libraries are focused only on a specific programming language, operating system, or platform.

The *pip* package manager is a package management system that is used to install packages from the Python Package Index, which is some library of such packages. Pip is often installed with Python. The pip package manager is used only for the Python programming language. It has many functions for working with packages:

- installation of a package;
- installation of a package of a specialized version;
- deletion of a package;
- reinstallation of a package;
- display of installed packages;
- search for packages;
- verification of package dependencies;
- creation of a configuration file with a list of installed packages and their versions;
- installation of a set of packages from a configuration file.

The pip package manager works well with dependencies, displays unsuccessfully installed packages, and also displays information about the required package version

```

requirements.txt
1 py==1.8.1
2 pip==19.0.3
3 Mako==1.1.1
4 MarkupSafe==1.1.1
5 six==1.14.0
6 attrs==19.3.0
7 pytest==5.3.5
8 pluggy==0.13.1
9 setuptools==40.8.0
10 parse==1.14.0
11 glob2==0.7

```

Figure 1. pip configuration file

in case of conflict with another package. An example of a pip package configuration file is shown in Figure 1.

An alternative to the pip package manager is the *poetry* package manager, which is also focused on the Python programming language. The advantage of poetry over pip is that it automatically works with virtual environments, is able to find and create them independently. The configuration file for poetry packages is more comprehensive than that of pip, it stores such information as the project name, project version, its description, license, list of authors, URL of the project, its documentation, and website, a list of project keywords, and a list of PyPI classifiers. Poetry allows configuring packages for Python projects more flexibly, the poetry configuration file is a more extensive project specification (see Figure 2), however, this specification does not allow for compatibility between components even within Python projects and is intended primarily for read-only by the developer.

```

[tool.poetry]
name = "first"
version = "0.1.0"
description = ""
authors = [".."]
readme = "README.md"

[tool.poetry.dependencies]
python = "^3.10"

[build-system]
requires = ["poetry-core"]
build-backend = "poetry-core.masonry.api"

```

Figure 2. poetry configuration file

It is impossible to automate the design of computer systems using the poetry or pip package manager, since it requires the intervention of a developer who needs to manually combine the interfaces of the installed packages. Other package managers of programming languages and operating systems are arranged according to the same

principle: there is a component storage (library), which is a set of packages of this programming language or operating system and with which the component manager interacts.

As a component approach to program design, it is possible to consider libraries of subprograms of modern programming languages, for example, *STL Library* (a library of standard C++ templates).

The *STL Library* is a set of consistent generalized algorithms, containers, means of accessing their contents, and various auxiliary functions in C++.

There are five main components of the STL Library:

- container – storage of a set of objects in memory;
- iterator – provision of access means to the contents of the container;
- algorithm – determination of the computational procedure;
- adapter – adaptation of components for providing different interface;
- functional object – privacy of a function in an object for use by other components.

The structure of the *STL Library* is shown in Figure 3.

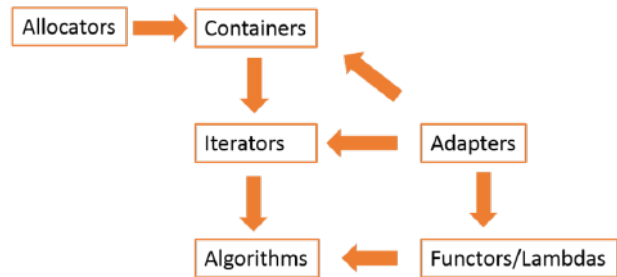


Figure 3. Structure of the STL Library

Separation allows reducing the number of components. For example, instead of writing a separate element search function for each container type, a single version is provided that works with each of them as long as the basic requirements are met.

The compatibility of components (containers) in the *STL Standard Template Library* is provided by a common interface for using these components.

The component approach to the design of computer systems can be implemented within various languages, platforms, and applications. Let us consider some of them.

The ontology implemented in *OWL* (Web Ontology Language) is a set of declarative statements about the entities of the dictionary of a subject domain (discussed in more detail in [16]). OWL assumes the concept of an “open world”, according to which the applicability of subject domain descriptions placed in a specific physical document is not limited only to the scope of this document – the contents of the ontology can be used and supplemented by other documents adding new facts about the same entities or describing another subject domain in

terms of this one. The “openness of the world” is achieved by adding a URI to each element of the ontology, which makes it possible to understand the ontology described in OWL as part of a universal unified knowledge.

WebProtege is a multi-user web interface that allows editing and storing ontologies in the OWL format in a collaborative environment [17]. This project allows not only creating new ontologies but also loading existing ontologies that are stored on the Stanford University server. The advantage of this project is the automatic error checking in the process of creating ontology objects. This project is an example of an attempt to solve the problem of accumulation, systematization, and reuse of existing solutions, however, the disadvantage of this solution is the isolation of the ontologies being developed. Each developed component has its own hierarchy of concepts, an approach to the allocation of classes and entities that depend on the developers of these ontologies, since within this approach, there is no universal model of knowledge representation, as well as formal specification of components represented in the form of ontologies. Consequently, there is a problem of their semantic incompatibility, which, in turn, leads to the impossibility of reuse of the developed ontologies in the knowledge bases design. This fact is confirmed by the presence on the Stanford University server of a variety of different ontologies on the same topics.

Based on the **Modelica** language, a large number of freely available component libraries have been developed, one of which is the **Modelica_StateGraph2** library, which includes components for modeling discrete events, reactive, and hybrid systems using hierarchical state diagrams [18]. The main disadvantage of Modelica-based systems is the lack of component compatibility and sufficient documentation.

Microsoft Power Apps is a set of applications, services, and connectors, as well as a data platform that provides a development environment for efficiently creating user applications for business. The Power Apps platform provides tools for creating a library of reusable graphical interface components, as well as pre-created text recognition models (reading visiting cards or cheques) and an object detection tool that can be connected to the application being developed [19]. The Power Apps component library is a set of user-created components that can be used in any application. The advantage of the library is that components can configure default properties that can be flexibly edited in any applications that use the components. The disadvantage lies in the lack of semantic compatibility of components, the specification of components; the problem of the presence of semantically equivalent components has not been solved; there is no hierarchy of components and means of searching for these components.

The **IACPaaS platform** is designed to support the

development, management, and remote use of applied and instrumental multi-agent cloud services (primarily intelligent ones) and their components for various subject domains [20]. The platform provides access to:

- application users (specialists in various subject domains) – to applied services;
- developers of applied and instrumental services and their components – to instrumental services;
- intelligent services managers and management services.

The IACPaaS platform supports:

- the basic technology for the development of applied and specialized instrumental (intelligent) services using the basic instrumental services of the platform that support this technology;
- a variety of specialized technologies for the development of applied and specialized instrumental (intelligent) services, using specialized platform tool services that support these technologies.

The IACPaaS platform also does not contain means for a unified representation of the components of intelligent computer systems and means for their specification and automatic integration.

Based on the analysis carried out, it can be said that at the current state of development of information technologies, there is no comprehensive library of reusable semantically compatible components of computer systems and corresponding component management tools. Thus, it is proposed to implement a library and an appropriate component management tool that will implement seamless integration of components, ensure semantic compatibility of systems and their components, and significantly simplify the design of new systems and their components.

III. PROPOSED APPROACH

Within this article, it is proposed to take an **OSTIS Technology** [21] as a basis, the principles of which make it possible to implement a semantic technology for designing intelligent systems, including a library of reusable components, component design support tools, and other components of the technology. The **OSTIS Technology** makes it possible to quickly create knowledge-driven systems using ready-made compatible components.

The systems developed on the basis of the OSTIS Technology are called *ostis-systems*. The **OSTIS Technology** is based on a universal method of semantic representation (encoding) of information in the memory of intelligent computer systems, called an *SC-code*. Texts of the *SC-code* (sc-texts) are unified semantic networks with a basic set-theoretic interpretation, which allows solving the problem of compatibility of various knowledge types. The elements of such semantic networks are called *sc-elements* (*sc-nodes* and *sc-connectors*, which, in turn, depending on orientation, can be *sc-arcs* or *sc-edges*). The *Alphabet of the SC-code* consists of five main elements, on the basis

of which SC-code constructions of any complexity are built, including more specific types of sc-elements (for example, new concepts). The memory that stores SC-code constructions is called semantic memory, or *sc-memory*.

Within the technology, several universal variants of visualization of *SC-code* constructions are proposed, such as *SCg-code* (graphic variant), *SCn-code* (nonlinear hypertext variant), *SCs-code* (linear string variant).

Within this article, fragments of structured texts in the SCn code [22] will often be used, which are simultaneously fragments of the source texts of the knowledge base, understandable to both human and machine. This allows making the text more structured and formalized, while maintaining its readability. The symbol “:=” in such texts indicates alternative (synonymous) names of the described entity, revealing in more detail certain of its features.

The basis of the knowledge base within the *OSTIS Technology* is a hierarchical system of subject domains and ontologies.

In order to solve the problems that have arisen in the design of intelligent systems and libraries of their reusable components, it is necessary to adhere to the general principles of the technology for intelligent computer systems design, as well as meet the following requirements:

- ensuring compatibility (integrability) of components of intelligent computer systems based on the unifying representation of these components;
- clear separation of the process of developing formal descriptions of intelligent computer systems and the process of their implementation according to this description;
- clear separation of the development of a formal description for the designed intelligent system from the development of various options for the interpretation of such formal descriptions of the systems;
- availability of an ontology for component design of intelligent computer systems, including (1) a description of component design methods, (2) a model of a *library of reusable components*, (3) a model of a *specification of reusable components*, (4) a complete *classification of reusable components*, (5) a description of means for interaction of the developed intelligent computer system with *libraries of reusable components*;
- availability of *libraries of reusable components of intelligent computer systems*, including component specifications;
- availability of means for interaction of the developed intelligent computer system with libraries of reusable components for installation of any types of components and their management in the created system. The installation of a component means not only its transportation to the system (copying sc-elements and/or downloading component files) but also the

subsequent execution of auxiliary actions so that the component can operate in the system being created.

Based on this, in order to solve the problems set within this article, it is proposed to develop the following system of subject domains and corresponding ontologies:

Subject domain of reusable ostis-systems components

⇒ *private subject domain**:

Subject domain of a library of reusable ostis-systems components

Subject domain of the manager of reusable ostis-systems components

The *Subject domain of reusable ostis-systems components* describes the concept of a reusable component, the classification of components, and their general specification. This subject domain allows creating new and specifying existing components to add them to the library.

As a *reusable ostis-systems component*, a component of some ostis-system that can be used within another ostis-system is understood (see [13]). This is a component of the ostis-system that can be used in other ostis-systems (*child ostis-systems*) and contains all those and only those sc-elements that are necessary for the functioning of the component in the child ostis-system. In other words, it is a component of some *maternal ostis-system*, which can be used in some child ostis-system. To include a reusable component in some system, it must be installed in this system, that is, all the sc-elements of the component should be copied into it and, if necessary, auxiliary files, such as the source or compiled component files. *Reusable components* must have a unified specification and hierarchy to support compatibility with other components. The compatibility of *reusable components* leads the system to a new quality, to an additional extension of the set of problems to be solved when integrating components.

reusable ostis-systems component

- := [typical ostis-systems component]
- := [reused ostis-systems component]
- := [reusable OSTIS component]
- := [ostis-systems ip-component]
- := *frequently used sc-identifier**:
[reusable component]
- ⊂ *ostis-system component*
- ⊂ *sc-structure*

The requirements for *reusable ostis-systems components* inherit the common requirements for the design of software components and also include the following ones [23]:

- there is a technical possibility to embed a reusable component into a child ostis-system;

- a reusable component should perform its functions in the most general way, so that the range of possible systems in which it can be embedded is the widest;
- compatibility of a reusable component: the component should strive to increase the level of negotiability of ostis-systems in which it is embedded and be able to be automatically integrated into other systems;
- self-sufficiency of components, that is, their ability to operate separately from other components without losing the appropriateness of their use.

In the *Subject domain of the library of reusable ostis-systems components*, the most common concepts and principles are described, which are valid for any *library of reusable components*. This subject domain allows building many different libraries, each of which will be semantically compatible with any other built according to the proposed principles. Such libraries store components and their specifications for use in child ostis-systems. An example of a specification of a reusable ostis-systems component is shown in Figure 4.

library of reusable ostis-systems components

- ⇒ *frequently used sc-identifier**:
[library of ostis-systems components]
- ⇒ *frequently used sc-identifier**:
[library of components]
- := [library of compatible reusable components]
- := [comprehensive library of reusable semantically compatible ostis-systems components]
- := [library of reusable and compatible components of intelligent computer systems of a new generation]
- := [library of typical ostis-systems components]
- := [library of reusable OSTIS components]
- := [library of reused OSTIS components]
- := [library of intelligent property ostis-systems components]
- := [library of ostis-systems ip-components]
- ⊃ *typical example'*:
OSTIS Metasystem Library
- := [Distributed library of typical (reusable) ostis-systems components as part of the *OSTIS Metasystem*]
- := [Library of reusable ostis-systems components as part of the *OSTIS Metasystem*]
- ⊃ *typical example'*:
OSTIS Metasystem Library
- ⇒ *frequently used sc-identifier**:
[OSTIS Library]
- := [Library of reusable and compatible components of intelligent computer systems of a new generation]
- := [Library of typical components of intelligent computer systems of a new generation]

:= [Distributed library of typical (reusable) ostis-systems components as part of the OSTIS Ecosystem]

:= [Library of reusable ostis-systems components as part of the *OSTIS Ecosystem*]

- ⇐ *combination**:
{
 - *library of reusable components of ostis-systems knowledge bases*
 - *library of reusable components of ostis-systems problem solvers*
 - *library of reusable components of ostis-systems interfaces*
 - *library of embedded ostis-systems*
 - *library of ostis-platforms*
}

First versions for the full contents of the *Subject domain of reusable ostis-systems components* and the *Subject domain of the library of reusable ostis-systems components* are represented in the work [24].

The **manager of reusable ostis-systems components** is the main means of supporting component design of intelligent computer systems built by the *OSTIS Technology*. It allows installing reusable components in ostis-systems and controlling them. The *Subject domain of the manager of reusable ostis-systems components* contains the full specification for the manager of ostis-systems components, the requirements for the component manager, its functionality, the specification of the implementation option for the manager of ostis-systems components, including the sc-model of the knowledge base, the problem solver, and the interface.

Before considering the model of the *manager of reusable ostis-systems components*, let us consider the general model of any *library of reusable ostis-systems components*, with which the component manager interacts, and the most important classes of reusable components. Next, we will consider in more detail the fragments for sc-models of the *Subject domain of the manager of reusable ostis-systems components*.

IV. LIBRARY OF REUSABLE OSTIS-SYSTEMS COMPONENTS

The basis for the implementation of the component approach within the *OSTIS Technology* is the **OSTIS Metasystem Library**. The *OSTIS Metasystem* is focused on the development and practical implementation of methods and tools for component design of semantically compatible intelligent computer systems, which provides an opportunity to quickly create intelligent systems for various purposes. The *OSTIS Metasystem* includes the *OSTIS Metasystem Library*. The scope of practical application for the technology of component design of semantically compatible intelligent systems does not have any limits.

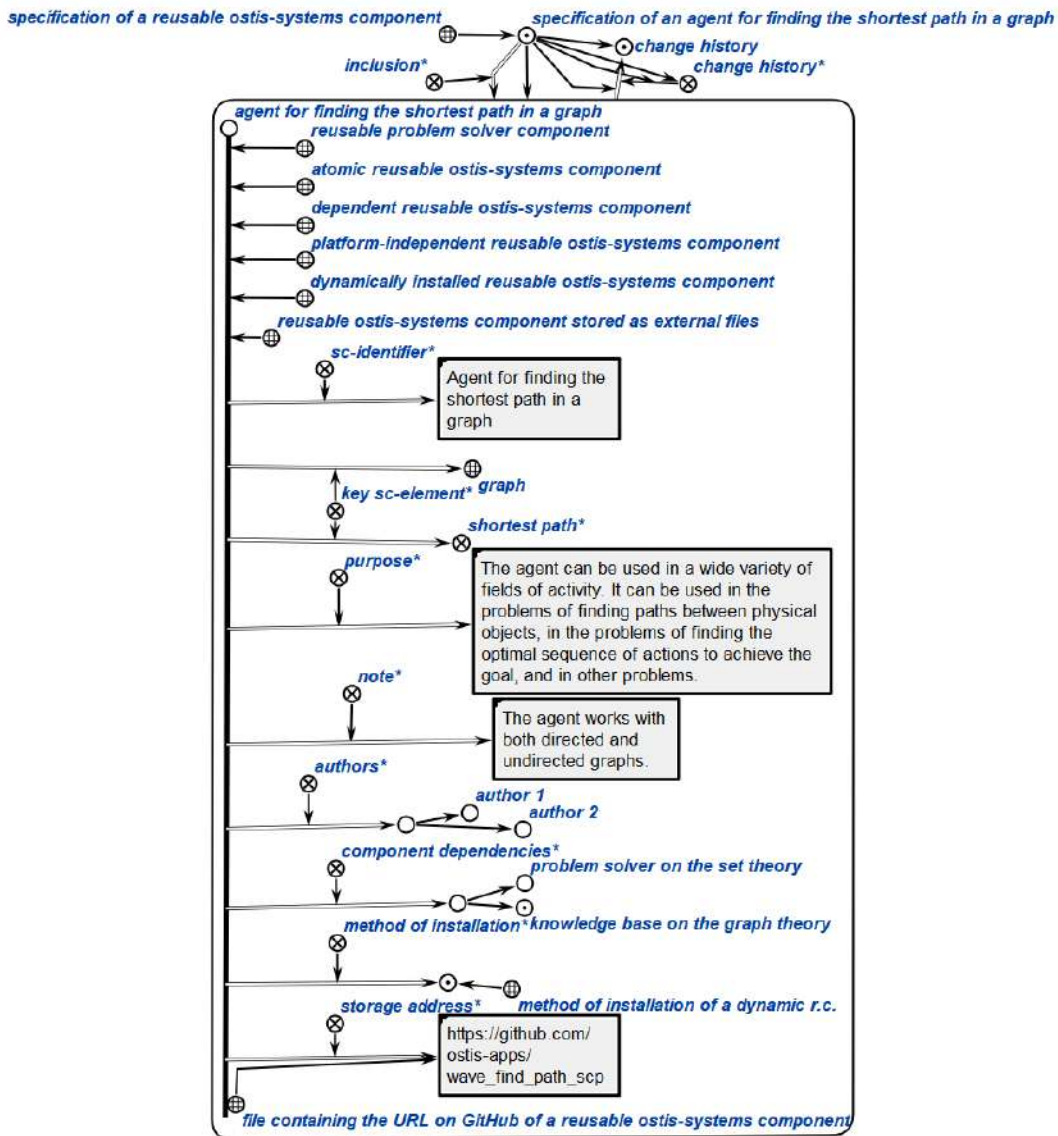


Figure 4. An example of a specification of a reusable ostis-systems component

The *OSTIS Metasystem* acts as the *maternal system* for all ostis-systems being developed, since it contains all the basic components for their development (Figure 5).

Functionality of any *library of reusable ostis-systems components* (see [24]):

- storage of reusable ostis-systems components and their specifications. At the same time, some of the components specified within the library may be physically stored elsewhere due to the peculiarities of their technical implementation (for example, the source texts of the ostis-platform may be physically stored in some separate storage, but they will be specified as a component of the corresponding library). In this case, the specification of the component within

the library should also include a description of (1) the where the component is located and (2) the scenario of its automatic installation in a child ostis-system;

- viewing available components and their specifications, as well as searching for components by fragments of their specification;
- storage of information about components use statistics. For example, in which *child ostis-systems* which of the library components and which version are used (are downloaded). This is necessary to take into account the demand for a particular component, to assess its importance and popularity;
- systematization of *reusable ostis-systems components*;
- provision of versioning of *reusable ostis-systems*

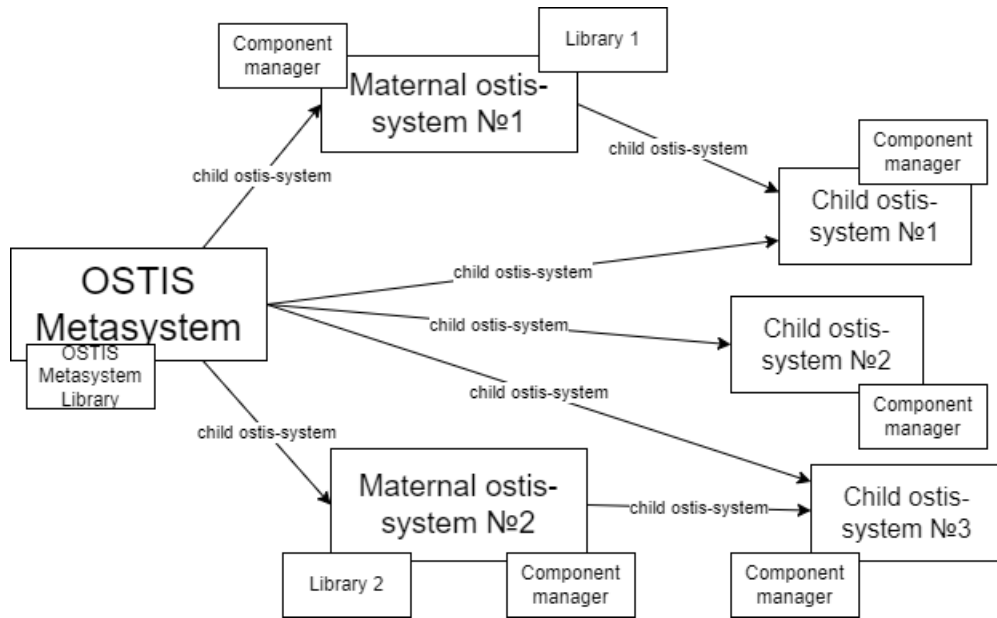


Figure 5. Architecture of the OSTIS Ecosystem in terms of libraries of reusable components

components;

- search for dependencies between reusable components within the library of components;
- provision of automatic updating of components borrowed into child ostis-systems. This function can be turned on and off at the request of the developers of the child ostis-system. Simultaneous updating of the same components in all systems using it should not in any context lead to inconsistency between these systems. This requirement may turn out to be quite complicated, but without it the work of the *OSTIS Ecosystem* is impossible.

The *library of reusable ostis-systems components* is an embedded ostis-system. It has its own knowledge base, its own problem solver, and its own interface. However, not every ostis-system is required to have a library of components.

Let us consider the most important classes of reusable ostis-systems components from the point of view of the *manager of reusable components*.

reusable ostis-systems component

⇒ *subdividing**:

- *atomic reusable ostis-systems component*
- *non-atomic reusable ostis-systems component*

The typology of ostis-systems components by atomicity. An atomic reusable ostis-systems component is a component that in the current state of the ostis-systems library is considered as indivisible, that is, does not contain

other components in its structure. A non-atomic reusable component in the current state of the ostis-systems library contains other atomic or non-atomic components in its structure and does not depend on its own parts. Without any part of the non-atomic component, its purpose restricts. The *manager of reusable ostis-systems components* allows installing both a whole non-atomic component and a selected subset of its constituent components. At the same time, there can be no inconsistency of such a combined component due to semantic compatibility of *reusable ostis-systems components*.

reusable ostis-systems component

⇒ *subdividing**:

- *dependent reusable ostis-systems component*
- *independent reusable ostis-systems component*

The typology of ostis-systems components by dependency. A dependent reusable ostis-systems component depends on at least one other component of the ostis-systems library, i.e. it cannot be embedded in a child ostis-system without the components on which it depends. The independent component does not depend on any other component of the ostis-systems library. When installing dependent components, the component manager installs all its dependencies, otherwise the component cannot operate in a child ostis-system. If any component, which is a dependency of another, is not installed and it is not possible to install any equivalent component, then the

installation of the dependent component in the current state of the ostis-system and the component libraries used by it is impossible.

reusable ostis-systems component

- ⇒ subdividing*:
- {• reusable ostis-systems component stored as external files
 - reusable ostis-systems component stored as an sc-structure
- }

reusable ostis-systems component stored as external files

- ⇒ subdividing*:
- {• reusable ostis-systems component stored as source files
 - reusable ostis-systems component stored as compiled files
- }

The typology of ostis-systems components by their storage method. A reusable component stored as an sc-structure is integrated into child systems in the simplest and most convenient way. The installation of such components takes place by copying the sc-elements of the structure from one ostis-system to another. When storing reusable components as external files, not all components can be dynamically installed. At this stage of development of the *OSTIS Technology*, it is more convenient to store components in the form of source texts.

reusable ostis-systems component

- ⇒ subdividing*:
- {• platform-dependent reusable ostis-systems component
 - platform-independent reusable ostis-systems component
- }

The typology of ostis-systems components depending on the ostis-platform. A platform-dependent reusable ostis-systems component is a component partially or fully implemented with the help of any third-party means from the point of view of the *OSTIS Technology*. The disadvantage of such components is that the integration of such components into intelligent systems may be accompanied by additional difficulties depending on the specific means of implementing the component. As a potential advantage of platform-dependent reusable ostis-systems components, it is possible to allocate them, as a rule, higher performance due to their implementation at a level closer to the platform one. In general, a platform-dependent reusable ostis-systems component can be supplied either as a set of source codes or compiled.

The process of integrating a platform-dependent reusable ostis-systems component into a child system developed using the *OSTIS Technology* strongly depends on the implementation technologies of this component and in each case may consist of various stages. Each platform-dependent reusable ostis-systems component must have the appropriate detailed, correct, and understandable instructions for its installation and implementation in the child system using the component manager. A platform-independent reusable ostis-systems component is a component that is entirely represented in the *SC-code*. The process of integrating a platform-dependent reusable ostis-systems component into a child system developed using the *OSTIS Technology* is significantly simplified by using a common unified formal basis for knowledge representation and processing.

The most valuable are platform-independent reusable ostis-systems components.

reusable ostis-systems component

- ⇒ subdividing*:
- {• dynamically installed reusable ostis-systems component
 - := [reusable component, the installation of which does not require a restart of the system]
 - reusable component, the installation of which requires a restart of the system
- }

dynamically installed reusable ostis-systems component

- ⇒ decomposition*:
- {• reusable component stored as compiled files
 - reusable knowledge base component
- }

The typology of ostis-systems components according to the dynamics of their installation. The process of integrating components of different types at different stages of the ostis-systems life cycle can be different. The most valuable components are those that can be integrated into a working system without stopping its functioning. Some systems, especially control ones, cannot be stopped, but components need to be installed and updated.

An **embedded ostis-system** is a non-atomic reusable component that consists of a knowledge base, a problem solver, and an interface.

embedded ostis-system

- ⊂ ostis-system
 - ⊂ non-atomic reusable ostis-systems component
- ⇒ decomposition*:
- {

- *reusable component of ostis-systems knowledge bases*
 - *reusable component of ostis-systems problem solvers*
 - *reusable component of ostis-systems interfaces*
- }

As such systems, for example, an intelligent interface (including a natural language interface), an environment for collective design of knowledge bases, a manager of reusable ostis-systems components, a training system, a system for testing and verification of intelligent systems, a visual web-oriented editor of sc.g-texts, and others can act.

The peculiarity of *embedded ostis-systems* is that the integration of entire intelligent systems involves the integration of the knowledge bases of these systems, the integration of their problem solvers, and the integration of their intelligent interfaces. When integrating embedded ostis-systems, the knowledge base of the embedded system becomes part of the knowledge base of the system into which it is embedded. The problem solver of the embedded ostis-system becomes part of the problem solver of the system into which it is embedded. And the interface of the embedded ostis-system becomes part of the interface of the system into which it is embedded. From the point of view of the component manager, this is equivalent to installing a non-atomic reusable ostis-systems component, that is, a separate installation of all its components. At the same time, the embedded system is integral and can function separately from other ostis-systems, unlike other reusable components.

Embedded ostis-systems are often subject-independent reusable components. Thus, for example, an embedded ostis-system in the form of a knowledge base design environment can be integrated both into a system from the subject domain of geometry and into an agricultural facilities management system.

The *embedded ostis-system*, like any other reusable ostis-systems component, should support semantic compatibility of ostis-systems. Both the embedded ostis-system itself and all its components must be specified and coordinated. Components of embedded ostis-systems can be replaced with others having the same purpose, for example, a natural language interface can have different versions of the knowledge base depending on the natural language supported by the system, different interface options, depending on the requirements and convenience of users, and also various options for implementing a problem solver for natural language processing, which can use different models but solve the same problem. The *manager of reusable components* allows flexibly selecting certain components of *embedded ostis-systems*, while maintaining their integrity and overall functionality. The embedded ostis-system connects with the system in

which it is embedded using the *embedded ostis-system** relation.

V. MANAGER OF REUSABLE OSTIS-SYSTEMS COMPONENTS

The *manager of reusable ostis-systems components* is a subsystem of the ostis-system, through which interaction with the *library of reusable ostis-systems components* takes place.

manager of reusable ostis-systems components

- ⊂ *embedded ostis-system*
- ⊂ *platform-dependent reusable ostis-systems component*
- := *frequently used sc-identifier**:
[manager of reusable components]
- := *frequently used sc-identifier**:
[component manager]
- ⇒ *generalized decomposition**:
 - *knowledge base of the manager of reusable ostis-systems components*
 - *problem solver of the manager of reusable ostis-systems components*
 - *interface of the manager of reusable ostis-systems components*
- }
- ⊃ *Implementation of the manager of reusable ostis-systems components*
- ⇒ *component address**:
[<https://github.com/ostis-ai/sc-component-manager>]

The *knowledge base of the component manager* contains all the knowledge that is necessary to install reusable components in the *child ostis-system*. Such knowledge includes knowledge about the specification of reusable components, methods of installing components, knowledge about the libraries of ostis-systems with which interaction occurs, classification of components, and others.

The *problem solver of the manager of reusable ostis-systems components* interacts with the *library of reusable ostis-systems components* and allows installing and integrating reusable components into a child ostis-system, as well as searching, updating, publishing, deleting components, and other operations with them. At a minimum, the component manager should provide the following functionality:

- **Search for reusable ostis-systems components.**
The set of possible search criteria corresponds to the *specification of reusable components*. As such criteria, the component classes, its authors, identifiers, a fragment of a note, purpose, belonging to a subject domain, the type of knowledge of the component, and others can serve.

- **Installation of a reusable ostis-systems component.**

The installation of a reusable component takes place regardless of the type, installation method, and location of the component. A necessary condition for the possibility of installing a reusable component is the availability of the *specification of a reusable ostis-systems component*. Before installing a reusable component into a child system, all dependent components must be installed. Also, for platform-dependent components, it may be necessary to perform additional steps for component installation, depending on the specific implementation of the component. After the component has been successfully installed, an information construction is generated in the knowledge base of the child system, indicating the fact that the component has been installed into the system using the *installed components** relation.

- **Addition and removal of library components controlled by the manager.** The component manager contains information about a variety of sources for installing components, the list of which can be supplemented. By default, the component manager monitors the *OSTIS Metasystem Library*, however, it is possible to create and add optional ostis-systems libraries.

Based on the specified minimum functionality, the *problem solver of the manager of reusable ostis-systems components* represents the following hierarchy of abstract sc-agents:

problem solver of the manager of reusable ostis-systems components

```
⇒ decomposition of an abstract sc-agent*:
{
• Abstract sc-agent for searching for reusable ostis-systems components
• Abstract sc-agent for installing reusable ostis-systems components
• Abstract sc-agent for managing library components controlled by the manager
}
```

Abstract sc-agent for managing library components controlled by the manager

```
⇒ decomposition of an abstract sc-agent*:
{
• Abstract sc-agent for adding a library controlled by the component manager
• Abstract sc-agent for deleting a library controlled by the component manager
}
```

Using minimal functionality, the component manager can install components that will extend its functionality. Components that implement the extended functionality of the component manager are part of the *OSTIS Metasystem*

Library. The extended functionality includes:

- **Specification** of a reusable ostis-systems component. The component manager allows specifying the components that are part of the ostis-systems library, as well as specifying new components that are being created, which will be published to the ostis-systems library. In this case, the specification can occur both automatically and manually. For example, the component manager can update the specification of the component used in accordance with which new ostis-systems have installed it, update the specification of the authorship of the component when editing it in the ostis-systems library or the specification of errors detected during the operation of the component, etc.
- **Creation** of a reusable ostis-systems component according to a template with specified parameters. When installing a template for a reusable ostis-systems component, the component manager allows creating a specific component based on it. To do this, the user is asked to determine the values of all sc-variables in the template to form a specific component from a certain subject domain. For example, to form a reusable component of knowledge bases, which is a semantic neighborhood of some relation (see Figure 6), it is necessary to determine the values of all variables, except for the variable that is the key sc-element of this structure.
- **Publication** of a reusable ostis-systems component to the ostis-systems library. When a component is published to the ostis-systems library, verification takes place based on the component specification. The publication of a component can be accompanied by the assembly of a non-atomic component from existing atomic ones. It is also possible to update the version of the published component by the team of its developers.
- **Update** of an installed reusable ostis-systems component.
- **Deletion** of an installed reusable component. As in the case of installation, after deleting a reusable component from the ostis-system, the fact of deleting the component is established in the knowledge base of the system. This information is an important part of the operational history of the ostis-system.
- **Edition** of a reusable component in the ostis-systems library.
- **Comparison** of reusable ostis-systems components.

The *interface of the manager of reusable ostis-systems components* provides convenient use of the component manager for the user and other systems. The minimal interface of the component manager is console-based and allows accessing the functionality of the component manager using commands. The minimal interface is available for both users and other ostis-systems. The

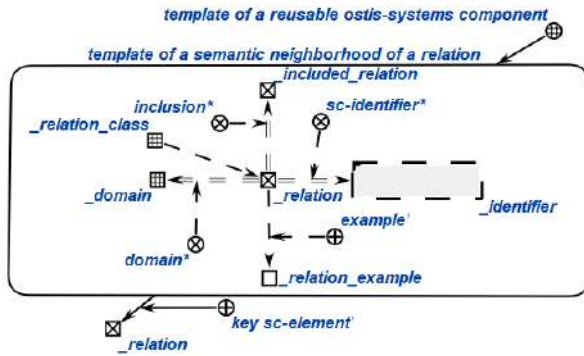


Figure 6. Example of a template for a semantic neighborhood of a relation

extended interface is graphical and is part of the *OSTIS Metasystem Library*. It can be installed using the reusable component manager. Thus, the component manager allows developing not only the systems in which it is embedded but also itself.

In order to create a new ostis-system “from scratch” using the *ostis-platform*, it is necessary to install some *Software implementation option of the ostis-platform* using third-party tools. As such tools, (1) platform source code storages, for example, cloud storages, such as a GitHub repository, with an appropriate set of platform installation instructions, or (2) installation tools for a pre-compiled software implementation of the platform, for example, an apt software installation tool, can serve. Next, the installation of reusable components in the ostis-system (regardless of the type of components) is carried out using the component manager. When installing platform-dependent components, the component manager must manage the appropriate means of assembling such components (CMake, Ninja, npm, grunt, and others).

In order to store reusable ostis-systems components, some storage is needed. In addition to the external files of the component, its specification must be located in the storage. As such a storage, (1) a *library of reusable ostis-systems components* or (2) files in some cloud storage can act. In the case when a component is stored in a library, to install it, the component manager copies all the sc-elements, which represent the component, to a child ostis-system. In the case when the component is stored as files in the cloud storage, the component manager downloads the component files and installs them according to the specification. The addresses of component specification storages should be stored in the knowledge base of the component manager in order to have access to component specifications for their subsequent use (search, installation, etc.). In Figure 7, there is an example for a fragment of the knowledge base of the component manager, which describes where the specifications of the components available for installation are stored. Such a storage is a set consisting of two sets: (1) a set of addresses of

component specifications and (2) a set of addresses of specifications for other storages. Thus, a tree-like structure is formed in accordance with the hierarchy of the maternal ostis-systems and their corresponding libraries.

When specifying the address for the root of the specification address storage tree, the component manager gets access to all the specifications of the child storages. When processing such a specification tree, the component manager immerses the specifications of the components available for installation into sc-memory but not these components themselves.

The *manager of reusable ostis-systems components* is an optional subsystem of the ostis-platform. However, a system with a component manager can install components not only into itself but also into other systems if it has an access. Thus, one system can replace the ostis-platform of another system, while leaving the *sc-model of a cybernetic system*. In the same way, some ostis-system can generate other ostis-systems using a component approach.

Inclusion of the component in the *child ostis-system* generally consists of the following steps:

- search for a suitable component in a set of available libraries;
- allocation of a component in a form convenient for transportation to a child ostis-system with an indication of the version and modification, if necessary (for example, selection of the available component storage, selection of the optimal implementation option for the component taking into account the composition of the child system);
- installation of a reusable component and its dependencies (indicating the version and modification, if necessary);
- integration of a component into a child system;
- search and elimination of errors and contradictions in the child system.

From the point of view of the user, this process does not depend on the type of the component and the specifics of its implementation.

VI. INTEGRATION OF REUSABLE OSTIS-SYSTEMS COMPONENTS

Let us consider in more detail the process of integrating reusable ostis-systems components as a process of immersing sc-constructions into the knowledge base. The most valuable components are those that can be integrated into a working system without stopping its functioning. Such components include, for example, knowledge base components.

The problem of integration of reusable ostis-systems components is solved by the interaction of components through a common knowledge base. Components can only use common key nodes (concepts) in the knowledge base (see [25] and [26]). Integration of reusable ostis-systems components is reduced to matching the key nodes and

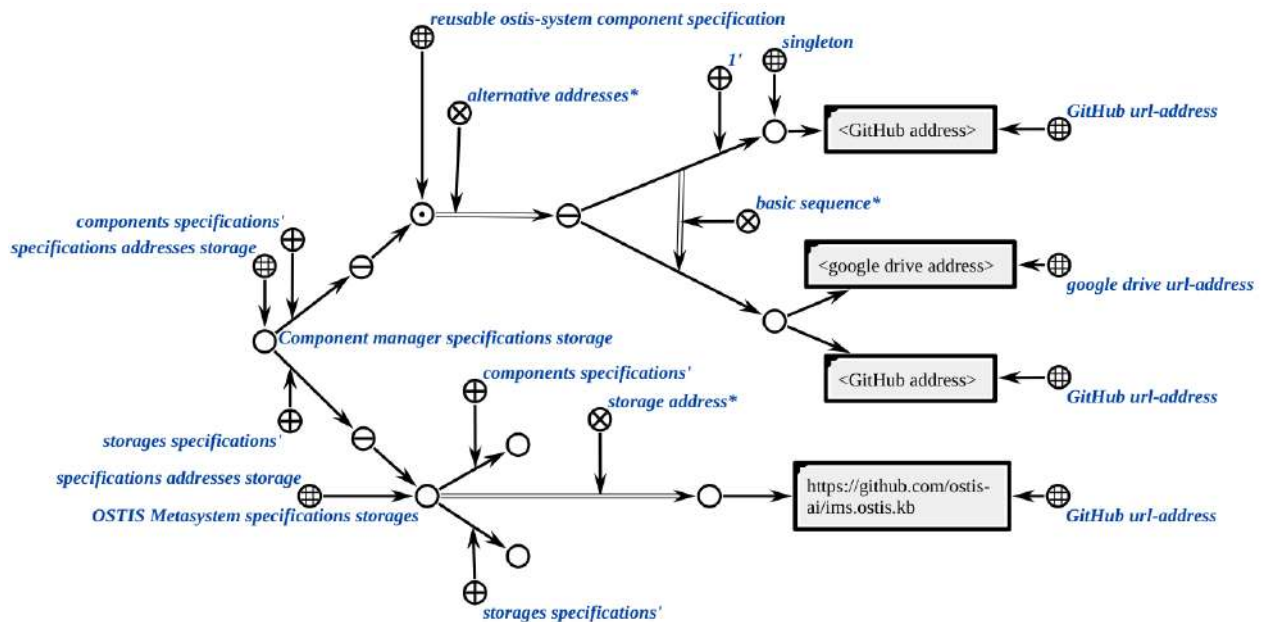


Figure 7. Example for the structure of the address storage of component specifications

eliminating possible duplications and contradictions based on the specification of the component and its contents. This way of integrating components allows them to be developed in parallel and independently of each other, which significantly reduces the time for design.

The process of integrating a reusable component into an ostis-system is a system of the following interacting processes:

- reducing a component to a unified form, that is, representing it in the SC-code;
- coordination of the key nodes and ontologies of the component and the knowledge base of the system;
- allocation of such sc-elements in the integrated component that have global (unique) identifiers;
- allocation of sc-elements in the integrated component that have local identifiers together with the scope of each such identifier. The scope of the local identifier is such a fragment of the knowledge base within which different sc-elements having this local identifier are considered synonymous;
- gluing together sc-elements having the same global identifiers;
- gluing together sc-elements having the same local identifiers if each of these sc-elements belongs to the scope of its local identifier and the scope of the local identifier of another sc-element;
- gluing together sc-elements based on the unambiguity of the algebraic operations used;
- gluing together sc-elements based on logical statements about the existence of uniqueness;
- gluing together multiple connectives belonging to relations in which multiple connectives are absent

either always or under certain clearly specified conditions.

Thus, the integration of components, that is, the process of immersion (understanding) of one semantic network into another, is a non-trivial process of reasoning aimed at identifying pairs of synonymous elements of a semantic network based on certain knowledge available in the knowledge base of an intelligent system. Based on these stages, it is important to clearly structure and fill the knowledge base of the ostis-system into which the component is integrated, as well as qualitatively specify components and the contents of these components.

Let us consider the simplest example of integrating a reusable component of problem solvers, which is an sc-agent for finding the shortest path in a graph, into a logistics process management system. Let us assume that the logistics process management system is able to solve problems related to cost optimization in the process of creating and warehousing of goods. An example for a fragment of a logistics system model is shown in Figure 8.

At the same time, the knowledge base of the system describes what problems the system should solve and the corresponding actions. For example, “action. find the shortest path in the graph”, which is currently difficult for the system to perform.

The graph theory problem-solving system has a rich knowledge base and a problem solver, including an sc-agent for finding the shortest path in a graph, which is specified as a reusable component and can be used in a logistics system (see Figure 9).

As a result of installing a *reusable component* in the

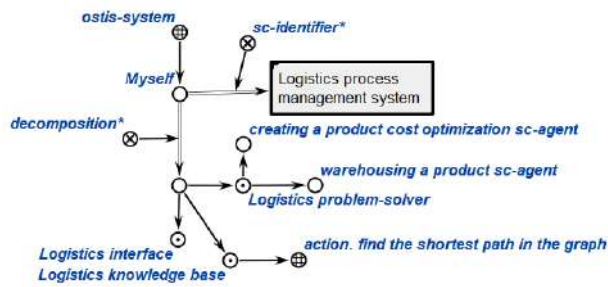


Figure 8. Structure of the logistics process management system

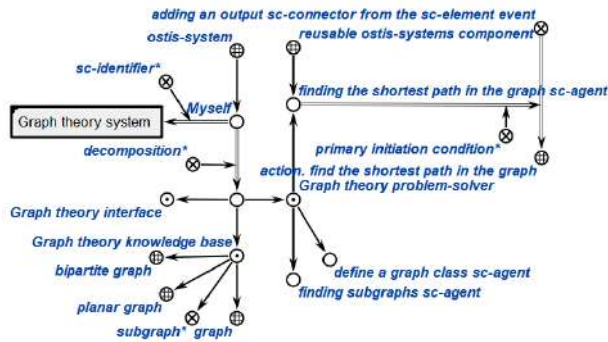


Figure 9. Structure of the graph theory problem-solving system

form of an sc-agent for finding the shortest path in the graph, its entire sc-model is immersed in a logistics problem-solving system. When integrating a reusable component, which is an sc-agent for finding the shortest path in a graph, into a logistics system, the key node of the logistics system “*action. find the shortest path in the graph*” is matched with the same node from the installed component from the graph theory system. Thus, when solving logistical problems, the system will be able to interpret the action of finding the shortest paths using an integrated component.

Integration of any ostis-systems components occurs automatically, without developer intervention. This is achieved through the use of the *SC-code* and its advantages, the unification of the specifications of reusable components, and the careful selection of components in libraries by the expert community responsible for this library.

VII. CONCLUSION

The component approach is key in the technology of designing intelligent computer systems. At the same time, the technology of component design is closely related to the other components of the technology of designing intelligent computer systems and ensures their compatibility, producing a powerful synergetic effect when using the entire complex of private technologies for designing intelligent systems. The most important principle in the implementation of the component approach

is the semantic compatibility of reusable components, which minimizes the participation of programmers in the creation of new computer systems and the improvement of existing ones.

To implement the component approach, in the article, a library of reusable compatible components of intelligent computer systems based on the *OSTIS Technology* is proposed, classification and specification of reusable ostis-systems components is introduced, a component manager model is proposed that allows ostis-systems to interact with libraries of reusable components and manage components in the system, the architecture of the ecosystem of intelligent computer systems is considered from the point of view of using a library of reusable components.

The results obtained will improve the design efficiency of intelligent systems and automation tools for the development of such systems, as well as provide an opportunity not only for the developer but also for the intelligent system to automatically supplement the system with new knowledge and skills.

REFERENCES

- [1] K. Yaghoobirafi and A. Farahani, “An approach for semantic interoperability in autonomic distributed intelligent systems,” *Journal of Software: Evolution and Process*, vol. 34, p. 18, 02 2022.
- [2] N. N. Skeeter, N. V. Ketko, A. B. Simonov, A. G. Gagarin, and I. Tislenkova, “Artificial intelligence: Problems and prospects of development,” *Artificial Intelligence: Anthropogenic Nature vs. Social Origin*, p. 12, 2020.
- [3] O. Yara, A. Brazheyev, L. Golovko, L. Golovko, and V. Bashkatova, “Legal regulation of the use of artificial intelligence: Problems and development prospects,” *European Journal of Sustainable Development*, p. 281, 2021.
- [4] J. Waters, B. J. Powers, and M. G. Ceruti, “Global interoperability using semantics, standards, science and technology (gis3t),” *Computer Standards & Interfaces*, vol. 31, no. 6, pp. 1158–1166, 2009.
- [5] M. Wooldridge, *An introduction to multiagent systems*, 2nd ed. Chichester : J. Wiley, 2009.
- [6] X. Shi, Z. Zheng, Y. Zhou, H. Jin, L. He, B. Liu, and Q.-S. Hua, “Graph processing on GPUs,” *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–35, Nov. 2018. [Online]. Available: <https://doi.org/10.1145/3128571>
- [7] P. Lopes de Lopes de Souza, W. Lopes de Lopes de Souza, and R. R. Ciferri, “Semantic interoperability in the internet of things: A systematic literature review,” in *ITNG 2022 19th International Conference on Information Technology-New Generations*, S. Latifi, Ed. Cham: Springer International Publishing, 2022, pp. 333–340.
- [8] I. Ashvin, *Component Design for Relational Databases*, 12 2021, pp. 143–156.
- [9] B. Ford, R. Schiano-Phan, and J. Vallejo, *Component Design*, 11 2019, pp. 160–174.
- [10] A. Donatis, *OOP in Component Design*, 01 2006, pp. 3–31.
- [11] Ryndin, Nikita and Sapegin, Sergey, “Component design of the complex software systems, based on solutions’ multivariant synthesis,” *International Journal of Engineering Trends and Technology*, vol. 69, pp. 280–286, 12 2021.
- [12] D. Shunkevich, “Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems,” in *Open semantic technologies for intelligent systems*, ser. 2, V. Golenkov, Ed. BSUIR, Minsk, 2018, pp. 119–132.

Средства управления многократно используемыми компонентами интеллектуальных компьютерных систем нового поколения

Орлов М. К.

Важнейшим этапом эволюции любой технологии является переход к компонентному проектированию на основе постоянно пополняемой библиотеки многократно используемых компонентов. В работе рассматривается подход к проектированию систем, управляемых знаниями, ориентированный на использование совместимых многократно используемых компонентов, что существенно сокращает трудоемкость разработки таких систем.

Received 13.03.2023

- [13] D. V. Shunkevich, I. T. Davydenko, D. N. Koronchik, I. I. Zukov, and A. V. Parkalov, "Sredstva podderzki komponentnogo projektirovaniya sistem upravlyaenih znaniyami [support tools knowledge-based systems component design]," *Open semantic technologies for intelligent systems*, pp. 79–88, 2015. [Online]. Available: <http://proc.ostis.net/proc/Proceedings%20OSTIS-2015.pdf>
- [14] M. Wilkes, *The Preparation of Programs for an Electronic Digital Computer: With Special Reference to the EDSAC and the Use of a Library of Subroutines*, ser. Addison-Wesley mathematics series. Addison-Wesley Press, 1951. [Online]. Available: <https://books.google.by/books?id=PzVVAAAAMAAJ>
- [15] J. Blähser, T. Göller, and M. Böhmer, "Thine — approach for a fault tolerant distributed packet manager based on hypercore protocol," in *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2021, pp. 1778–1782.
- [16] V. Torres da Silva, J. S. dos Santos, R. Thiago, E. Soares, and L. Guerreiro Azevedo, "OWL ontology evolution: understanding and unifying the complex changes," *The Knowledge Engineering Review*, vol. 37, p. e10, 2022.
- [17] A. Memduhoglu and M. Basaraner, "Possible contributions of spatial semantic methods and technologies to multi-representation spatial database paradigm," *International Journal of Engineering and Geosciences*, pp. 108 – 118, Oct. 2018. [Online]. Available: <https://doi.org/10.26833/ijeg.413473>
- [18] P. Fritzon, *Modelica Library Overview*, 2015, pp. 909–975.
- [19] S. Prakash Pradhan, *Working with Microsoft Power Apps*. Berkeley, CA: Apress, 2022, pp. 79–131. [Online]. Available: https://doi.org/10.1007/978-1-4842-8600-5_3
- [20] V. Gribova, L. Fedorishev, P. Moskalenko, and V. Timchenko, "Interaction of cloud services with external software and its implementation on the IACPaaS platform," pp. 1–11, 2021.
- [21] V. Golenkov, N. Guliakina, and D. Shunkevich, *Otkrytaya tekhnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheskii sovmestimyykh gibridnykh intellektual'nykh komp'yuternykh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*. Minsk: Bestprint [Bestprint], 2021, (In Russ.).
- [22] (2023, September) IMS.ostis Metasystem. [Online]. Available: <https://ims.ostis.net>
- [23] G. Sellitto, E. Iannone, Z. Codabux, V. Lenarduzzi, A. De Lucia, F. Palomba, and F. Ferrucci, "Toward understanding the impact of refactoring on program comprehension," in *29th International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2022, pp. 1–12.
- [24] M. K. Orlov, "Comprehensive library of reusable semantically compatible components of next-generation intelligent computer systems," in *Open semantic technologies for intelligent systems*, ser. Iss. 6. Minsk : BSUIR, 2022, pp. 261–272.
- [25] C. Acharya, D. Ojha, R. Gokhale, and P. C. Patel, "Managing information for innovation using knowledge integration capability: The role of boundary spanning objects," *International Journal of Information Management*, vol. 62, p. 102438, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0268401221001316>
- [26] V. Ivashenko, "Application of an integration platform for ontological model-based problem solving using a unified semantic knowledge representation," in *Open semantic technologies for intelligent systems*, ser. Iss. 5. Minsk : BSUIR, 2022, pp. 179–186.

Integration of Third-Party Functional Services on a Unified Semantic Basis

Alexandr Zagorskiy

Belarusian State University of Informatics and Radioelectronics

Minsk, Belarus

Email: alexandr.zagorskiy.research@gmail.com

Abstract—In the article, a digital ecosystem structure that utilizes semantic representation of information and investigates the integration principles for heterogeneous services is proposed. An example of this approach is demonstrated using the OSTIS Technology. The results obtained highlight the potential benefits of using semantic representation to enhance interoperability and adaptability of digital ecosystems. This research has significant implications for the design of more efficient systems that better match the user needs and preferences.

Keywords—Digital ecosystem, interoperability, integration, semantics

I. INTRODUCTION

To increase the level of automation of more and more broad types of human activities, a qualitative transition to the development of whole complexes of independently interacting intelligent computer systems is required [1]. The central problem of the next stage of information technology development is the problem of semantic interoperability of computer systems and their components [2]. The solution to this problem requires a transition from traditional computer systems and modern intelligent systems to semantically compatible computer systems [3], [4].

Semantic computer systems are next-generation computer systems that eliminate many of the shortcomings of modern computer systems. Nevertheless, the mass development of such systems requires appropriate technology, which should include:

- methods and design tools for semantic computer systems;
- methods and tools for continuous improvement of the technology itself.

The purpose of the work is to develop methods and tools for integrating various services into intelligent systems as one of the stages for designing a unified ecosystem of semantic computer systems. In the article, the problems of forming digital ecosystems using traditional approaches are considered, an approach to creating a digital ecosystem of semantically compatible computer systems is proposed, and an example of integration of functional services with the existing intelligent dialog system is demonstrated.

II. PROBLEM DEFINITION

The concept of a *digital ecosystem* is a complex and dynamic system that consists of many components, including technologies, processes, users, enterprises, etc. In the context of digital technologies, the transition to the *digital ecosystem* is a key aspect for achieving business and societal goals through more flexible and sustainable management: “Society 5.0”, “Industry 4.0”, “Smart Home”, “Smart City”, “Knowledge Market” [5].

The concept of the *digital ecosystem* can be defined as a set of digital products and services that interact with each other and with the external environment, forming a single habitat. Implementation of the *digital ecosystem* is strongly related to the formation of a *distributed system*. This principle of implementation has both advantages (high level of adaptability, stability, connectivity) and disadvantages (non-optimality, uncontrollability, unpredictability of behavior) [6], [7], [8], [9].

In traditional approaches to the problem of forming a *digital ecosystem*, there are problems associated with the low level of *interoperability* of such systems [10]. Traditional approaches to solving this problem are often ineffective, since each of the systems has its own specialized application interface and data format for interaction. This leads to additional costs to eliminate the shortcomings of such problems. Life-cycle support and modification of existing systems may also require additional time and resources.

The use of open standards and interoperability protocols can greatly simplify the problem of ensuring *interoperability* between different systems. This improves the efficiency and cost-effectiveness of digital transformation projects and reduces the time and cost for developing and maintaining the *digital ecosystem* [11].

Thus, in order to create a successful *digital ecosystem*, many problems need to be solved to ensure a high level of interoperability between independently operating systems. The Semantic Web idea aims to solve these problems [12], but it also raises other problems or limitations that need to be considered [13], [14], [15].

In *integration* of *services* into such a *digital ecosystem*, several problems arise that can hinder the *integration*

process and reduce the efficiency of the ecosystem [10], [16]. Some of these problems may include:

- different data formats and exchange protocols that can lead to errors in *information exchange*, which makes it difficult to communicate between *services*;
- *incompatibility* of application versions, which can lead to conflicts in *information exchange*;
- different security levels, which can lead to leaks of confidential information;
- lack of a single control point, which makes it difficult to monitor and manage the integration process;
- lack of mechanisms for analyzing and managing information, which makes it difficult to control *integration* processes.

These problems significantly complicate the development of the *services* themselves and lead to a significant increase in time and material costs. To solve these problems, various approaches and technologies are used to integrate *digital ecosystems* with various *services* and *resources* [17], [18]. Some of them may include:

- using standard protocols and data exchange formats, such as XML, JSON, and others, to make *information exchange* more reliable and universal;
- development of a unified data schema and access rules to make *integration* simpler and more manageable;
- implementation of mechanisms for automatic error and conflict handling, which reduces errors and improves reliability of the *digital ecosystem*;
- use of tools and technologies for information analysis and management, such as business analysis and information management systems, which allows controlling *information exchange* processes and optimizing their performance.

A possible solution is to move to universal communities of individual *intelligent cybernetic systems*, which are integrated into *multi-agent systems*, thus forming a single digital ecosystem. The general principles for *integration* of modern *services* with such a digital ecosystem may look as follows [10], [16]:

- *standardization* and *compatibility*, which is achieved by using standardized protocols and data exchange formats;
- openness and accessibility of the digital ecosystem to various stakeholders through open and user-friendly interfaces;
- security and confidentiality, which is achieved through the use of cryptographic methods of data protection and access control to resources;
- *automation* and *scalability*, which will ensure the efficiency and productivity of the *digital ecosystem* when dealing with a large number of *services* and *resources*;
- analysis and data management to help determine the effectiveness of *integration* and improve it further.

III. PROPOSED APPROACH

Within this article, it is suggested to use the OSTIS Technology as the foundation [19]. The OSTIS Technology comprises a range of technologies that facilitate the creation, management, operation, and improvement of intelligent computer systems aimed at automating various human activities. This Technology is built on semantic representation and ontological systemization of information and agent-oriented knowledge processing.

The principles of the OSTIS Technology can be summarized as follows:

- Focus on developing highly intelligent and socialized computer systems called *ostis-systems*.
- Focus on automation of various types and areas of human activities by creating a network of interacting and coordinating *ostis-systems*, referred to as the OSTIS Ecosystem.
- Implementation of the OSTIS Technology as a network of *ostis-systems*, which includes the OSTIS Metasystem as a key *ostis-system*. The Metasystem implements the OSTIS Technology Core, which consists of subject-independent methods and tools for designing and producing *ostis-systems*, along with built-in support subsystems for operation and reengineering. Other specialized *ostis-systems* within the network use various *ostis-technologies* to automate different areas and types of human activities.
- Focus on developing next-generation computers capable of efficiently interpreting the logical-semantic models of *ostis-systems*, represented by the knowledge bases of these systems with semantic representation.

Within this article, fragments of structured texts in the SCn-code [20] will be used, which are simultaneously fragments of the source texts of the knowledge base, understandable both to human and machine. This allows making the text more structured and formalized, while maintaining its readability.

IV. OSTIS ECOSYSTEM: A NETWORK FOR INTERACTING AND COORDINATING INTELLIGENT SYSTEMS

The *OSTIS Technology* is a powerful tool for designing and implementing digital ecosystems that can integrate with various services and information resources. The realization of such a universal community of interoperable *intelligent cybernetic systems* can be implemented as a global *OSTIS Ecosystem* [21]. The *OSTIS Ecosystem* is a socio-technical ecosystem, which is a collective of interacting *semantic computer systems*, providing permanent support for evolution and semantic compatibility of all its constituent systems, throughout their entire life cycle.

A system built in accordance with the requirements and standards of the *OSTIS Technology* is defined as an *ostis-system*.

The *OSTIS Ecosystem* is a collective of:

- *ostis-systems* themselves;
- users of those *ostis-systems* (both end-users and developers);
- some *computer systems* that are not *ostis-systems* but are considered by them as additional *information resources* or *services*.

Within the *OSTIS Ecosystem*, the *ostis-systems* are able to communicate with each other and form specialized collectives to jointly solve complex problems. This approach not only increases the intelligence level of each *individual cybernetic system* but also ensures more effective interaction between them within a single *digital ecosystem*. This provides a significant development for a number of properties of each *computer system*, allowing a significant increase in the *level of intelligence* (and, above all, their *level of learnability* and *level of socialization*).

The members of the *OSTIS Ecosystem* are characterized as:

- *semantically compatible*;
- constantly evolving individually;
- constantly maintaining its compatibility with other members in the course of its individual evolution;
- capable of decentralized coordination.

The *OSTIS Ecosystem* is a transition from *single ostis-systems* to *collective ostis-systems*, that is, to distributed *ostis-systems*.

ostis-system

⇒ *subdividing**:

- { • *stand-alone ostis-system*
- *build-in ostis-system*
- *collective of ostis-systems*
- }

The purpose of the *OSTIS Ecosystem* is to ensure the *compatibility of computer systems* that are part of the *OSTIS Ecosystem* both during their development and operation. One of the problems is the implementation of various changes to the systems, that are part of the *OSTIS Ecosystem*, during their operation, which can cause compatibility to be compromised.

OSTIS Ecosystem

⇒ *objectives**:

- *operational implementation of all agreed changes of the ostis-systems standard (including changes to the systems of concepts used and their corresponding terms)*
- *permanent support for a high level of mutual understanding of all systems included in the OSTIS Ecosystem and all their users*
- *corporate solution of various complex problems requiring coordination of*

several ostis-systems and, possibly, some users

The *OSTIS Ecosystem* is the next stage in the development of human society, providing a significant increase in the level of public (collective) intelligence by transforming human society into an ecosystem consisting of humans and semantically compatible intelligent systems. The *OSTIS Ecosystem* is a proposed approach to the implementation of a smart society, or Society 5.0, built on the basis of the *OSTIS Technology*.

The super-purpose of the *OSTIS Ecosystem* is not just a comprehensive automation of all types of human activities (only those activities whose automation is appropriate) but also a significant increase in the level of intelligence of various human (more precisely, human-machine) communities and the entire human society as a whole.

V. SUPPORT OF COMPATIBILITY BETWEEN OSTIS-SYSTEMS THAT ARE PART OF THE OSTIS ECOSYSTEM

Each system that is part of the *OSTIS ecosystem* must:

- train intensively, actively, and purposefully, either with the help of teacher-developers or independently;
- inform all other systems about proposed or finally approved changes in ontologies and, in particular, in the set of concepts used;
- accept proposals from other *ostis-systems* for changes in ontologies, including the set of concepts used, for agreement or approval of these proposals;
- implement approved changes in ontologies stored in its knowledge base;
- ensure maintaining a high level of semantic compatibility not only with other *ostis-systems* that are part of the *OSTIS Ecosystem* but also with its users (train them, inform them about changes in ontologies).

The *stand-alone ostis-system* that is part of the *OSTIS Ecosystem* has special requirements:

- it must have all the necessary knowledge and skills for messaging and purposeful interaction with other *ostis-systems* that are part of the *OSTIS Ecosystem*;
- in the context of constant change and evolution of the *ostis-systems* that build-up the *OSTIS Ecosystem*, it must itself monitor its compatibility (consistency) with all other *ostis-systems*, that is, it must maintain this compatibility by coordinating all changes with other *ostis-systems*, which require coordination in itself and in other systems.

To ensure high operational efficiency and high rate of evolution of the *OSTIS Ecosystem*, it is necessary to constantly improve the level of information compatibility (level of mutual understanding) not only between the computer systems that are part of the *OSTIS Ecosystem* but also between these systems and their users.

One of the ways to ensure such compatibility is the striving to ensure that the knowledge base, the picture of the world of each user, becomes part of the unified knowledge base of the *OSTIS Ecosystem*. This means that each user must know how the structure of each scientific and technical discipline is arranged (objects of research, subjects of research, definitions, patterns, etc.), how different disciplines can be interconnected.

The interoperability support requirements were taken into account when designing the *OSTIS Ecosystem* structure [21].

VI. PRINCIPLES FOR INTEGRATING THE *OSTIS ECOSYSTEM* WITH HETEROGENEOUS SERVICES

In the first stages of the transition to Society 5.0, there is no need to convert all modern automation systems of some types and areas of human activities into *ostis-systems*. Systems based on the principles of the *OSTIS Technology* can take on a coordinating and interconnecting role due to their high level of *interoperability*. Such systems must learn either to perform the mission of an active interoperable superstructure over various modern automation tools or to task modern automation tools with feasible problems for them, ensuring their direct participation in solving complex problems and organizing the interaction management for various automation tools in the process of collectively solving complex problems.

In the context of integration of the *OSTIS Ecosystem* with heterogeneous *services*, the *services* refer to applications, programs, *web-services*, and other *information systems* that provide a certain functionality, a mechanism for transforming information according to a given function. Frequently, such an application can provide an *application interface* that can be used with a certain format of inputs to which certain formats of outputs will correspond.

The *integration* of the *OSTIS Ecosystem* with the *service* should be understood as the ability to use the functionality of the *service* to change the internal state of the *knowledge base* of the *OSTIS Ecosystem*. Several levels of *integration* are distinguished, which allow interaction with various *information resources* and *services*.

Full *integration* implies executing the function of the *service* on a platform-independent level, where the entire program is executed in the *knowledge base* of the *OSTIS Ecosystem*. That is, the problem of integration of such a *service* is reduced to allocation of the graph structure processing algorithm and its implementation within the *knowledge base*. Within the *OSTIS Technology*, interpretation of such a processing program can be implemented with the help of the SCP Language [22]. As a result of such a full integration, the need to use a third-party *service* is eliminated.

Partial *integration* involves the implementation of interaction and state changes for the *knowledge base*

of the *OSTIS Ecosystem* at the stages of executing the function of the *service*. The degree of *integration* depth can vary: in some cases the *service* may refer to the *knowledge base* for additional information, in others — for immersing intermediate results of execution into the *knowledge base* of the system. In the simplest case, changing the *knowledge base* of the *OSTIS Ecosystem* can take place once, after receiving the result of executing the function of the *service*. The interaction with the *knowledge base* in this case will be performed by a special *sc-agent* using a third-party *service*.

To ensure the *integration* of a functional *service*, the following minimum requirements must be met:

- specification of the input construction in the system *knowledge base*: defining the structure in the system *knowledge base* that will be converted to a data format compatible with the *service*;
- specification of the output construction in the system *knowledge base*: defining the structure in the system *knowledge base*, which will be formed from the original structure after the transforming the data of the *service* into knowledge;
- implementation of the *sc-agent*, which converts the construction of the *knowledge base* into a format that can be used in the *service*, as well as immerse the results of the *service* operation back into the system *knowledge base* in accordance with the specification.

Meeting these requirements will allow for effective *integration* of the functional *service*, which, in turn, will allow the data and functionality of the *service* to be used in various *ostis-systems*.

The generalized algorithm of the *sc-agent* using a third-party *service* for *integration* of functionality can be described as follows:

- extracting the necessary knowledge structures from the *knowledge base*, that meet the requirements of a functional *service*;
- converting the extracted knowledge into the format required for input to the functional *service*;
- forming a request for a functional *service* and waiting for its response;
- forming knowledge structures based on the data received from the functional *service*;
- immersing new knowledge structures in the *intelligent system knowledge base* in order to ensure their further use.

It should be noted that during the *integration* of functional *services*, it may be necessary to carry out additional data processing and transformation, for example, to ensure their *compatibility* with data formats or to ensure the security of data transmission and storage.

Thus, the introduction of the possibility to use a third-party *service* in the *OSTIS Ecosystem* assumes the following steps:

- analysis of the requirements for the integrated *service*, determining the necessary functionality, input and output data formats, and other characteristics of the service;
- development of the *integration* specification, which will define data formats and rules of interaction between the *knowledge base* of the *OSTIS Ecosystem* and a third-party *service*;
- development of an *sc-agent* that will provide interaction between the *knowledge base* and the third-party *service* in accordance with the *integration* specification;
- testing and debugging;
- embedding in the *OSTIS Ecosystem*, which will allow using the capabilities of the integrated third-party *service* in various *ostis-systems*.

The use of the described approach is considered in the work [23] on the example of the integration of computer vision services with the dialog *ostis-system* [24]. Similarly, other data processing models can be implemented: natural language processing systems [25], large language models such as ChatGPT [26], etc.

The most important feature when developing *ostis-systems* is that the development of an *ostis-system* actually boils down to the development of its knowledge base. When developing the components of the problem solver and the interface, their features are taken into account [27], however, the general mechanism for making any changes to the *ostis-system* becomes unified [28]. It follows from this that integrated tools can also affect both the system interface and the algorithm for solving certain problems.

Various approaches are used to enable the *sc-agent*, one of which is to connect the *sc-agent* within the already existing, main *ostis-system*. From the point of view of scalability when implementing this approach, the *monolithic architecture* of the resulting *ostis-system*, which simplifies the process of introducing new *services* and *sc-agents* into the *ostis-system*, should be noted.

The advantage of this approach is a simpler and more convenient introduction of new *services* and *sc-agents* into the *ostis-system*, as well as simplification of the dependency management process. Using the *ostis-system* implementation with *monolithic architecture* can be applied in cases where the functional *service* does not need frequent modification and has a fairly simple structure. In addition, *monolithic architecture* may be more convenient in cases where access to the *service* occurs via an internal network and requires low latency and high performance. In this way, both *services* for obtaining knowledge from external sources (obtaining weather forecasts, processing statistical information, etc.) and functional *services* (processing audio information and immersing the processing results in the *knowledge base*, obtaining a syntactic analysis of the sentence) can be integrated. An alternative implementation option is the

implementation of a separate *ostis-system*, within which the function of the *service* will be integrated. This allows switching to the use of *microservice architecture*, which is characterized by distributed interaction of *ostis-systems*.

The advantages of this approach are greater *flexibility* and *scalability*. The approach is characterized by a high degree of distribution, decentralization, and availability of new functionality. The system goes beyond technical limitations, the functionality can be distributed on different hardware. The resulting functionality can be used by various *ostis-systems* within the *OSTIS Ecosystem* to achieve their goals. The disadvantages of such a system are the complexity of development, as well as the increase in time spent on communication of systems with each other over network protocols.

It is preferable to use the *microservice architecture* of *ostis-systems* in cases where the functional *service* has a complex structure, as well as in cases where scaling and flexibility of the entire system is required. An example is a *service*, which interacts with external data sources and may be prone to frequent changes. As examples of *integration* based on *microservice architecture*, *services* for reading the emotional state of the user, natural language processing, classification, and identification problems, etc. can serve.

Thus, the selection of the approach to *integration* of functional *services* depends on the specific requirements and conditions of the project. Using the *OSTIS Technology* allows creating flexible and efficient systems that can be adapted to different needs and requirements of users.

VII. CONCLUSION

To increase the level of automation of human activities, it is necessary to automate more complex problems. The solution of such problems is reduced to the requirement to increase the level of intelligence of individual cybernetic systems. However, the individual intelligence of cybernetic systems has its limitations. It is possible to achieve a significant increase in the level of intelligence by forming collectives of cybernetic systems, i.e. by moving to multi-agent systems [29]. Within such systems, the possibility of communication of each agent with each one is provided, as well as the possibility of forming specialized collectives for the solution of complex problems.

The possibility of *integration* of traditional *services* and *information resources* with a single platform of semantically compatible computer systems is an essential element for promoting technological innovations in the modern world. As the demand for efficient, reliable, and affordable systems continues to grow, it is very important to have a comprehensive and integrated platform capable of meeting various needs.

The *OSTIS Technology* provides the basis for the development of complex systems and their seamless *integration* with existing *services* and *resources*. Systems

developed within an integrated ecosystem together provide opportunities to use a wide variety of functionality: information management, data analysis, decision-making, automation, etc.

ACKNOWLEDGMENT

The author would like to thank the research group of the Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics for its help in the work and valuable comments.

REFERENCES

[1] V. Golenkov, N. Guliakina, I. Davydenko, and A. Ereemeev, "Methods and tools for ensuring compatibility of computer systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 25–52, 2019.

[2] F. W. Neiva, J. M. N. David, R. Braga, and F. Campos, "Towards pragmatic interoperability to support collaboration: A systematic review and mapping of the literature," *Information and Software Technology*, vol. 72, pp. 137–150, 2016.

[3] P. Lopes de Lopes de Souza, W. Lopes de Lopes de Souza, and R. R. Ciferri, "Semantic Interoperability in the Internet of Things: A Systematic Literature Review," in *ITNG 2022 19th International Conference on Information Technology-New Generations*, S. Latifi, Ed. Cham: Springer International Publishing, 2022, pp. 333–340.

[4] Hamilton, Gunther, Drummond, and Wideregren, "Interoperability – A Key Element for the Grid and DER of the Future," in *2005/2006 IEEE/PES Transmission and Distribution Conference and Exhibition*, 2006, pp. 927–931.

[5] M. Tsujimoto, Y. Kajikawa, J. Tomita, and Y. Matsumoto, "A review of the ecosystem concept — towards coherent ecosystem design," *Technological Forecasting and Social Change*, vol. 136, pp. 49–58, 2018.

[6] K. Kelly, *Out of Control: The New Biology of Machines, Social Systems, and the Economic World*. USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[7] H. Boley and E. Chang, "Digital ecosystems: Principles and semantics," in *2007 Inaugural IEEE-IES Digital EcoSystems and Technologies Conference*, 2007, pp. 398–403.

[8] G. Briscoe and P. D. Wilde, "Digital ecosystems: Self-organisation of evolving agent populations," *CoRR*, vol. abs/0803.2675, 2008. [Online]. Available: <http://arxiv.org/abs/0803.2675>

[9] T. Burstrom, T. Lahti, V. Parida, M. Wartiovaara, and J. Wincen, "Software ecosystems now and in the future: A definition, systematic literature review, and integration into the business and digital ecosystem literature," *IEEE Transactions on Engineering Management*, 2022.

[10] W. Li, Y. Badr, and F. Biennier, "Digital ecosystems: challenges and prospects," pp. 117–122, 2012.

[11] M. Mohseni, M. Sohrabi, and M. Dorrigiv, "A model-driven approach for semantic web service modeling using web service modeling languages," *Journal of Software: Evolution and Process*, vol. 33, 06 2021.

[12] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001.

[13] S. Kirrane, S. Villata, and M. d'Aquin, "Privacy, security and policies: A review of problems and solutions with semantic web technologies," *Semantic Web*, vol. 9, no. 2, pp. 153–161, 2018.

[14] R. McCool, "Rethinking the semantic web. part 2," *IEEE Internet Computing*, vol. 10, no. 1, pp. 93–96, 2006.

[15] H. Nacer and D. Aissani, "Semantic web services: Standards, applications, challenges and solutions," *Journal of Network and Computer Applications*, vol. 44, pp. 134–151, 2014.

[16] O. Valdez-De-Leon, "How to develop a digital ecosystem: A practical framework," *Technology Innovation Management Review*, vol. 9, no. 8, 2019.

[17] E. G. Caldarola, A. Picariello, and A. M. Rinaldi, "An approach to ontology integration for ontology reuse in knowledge based digital ecosystems," pp. 1–8, 2015.

[18] D. Bork, R. Buchmann, D. Karagiannis, M. Lee, and E.-T. Miron, "An open platform for modeling method conceptualization: the omilab digital ecosystem," 2019.

[19] V. Golenkov, N. Guliakina, and D. Shunkevich, *Otkrytaya tekhnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheski sovmestimyykh gibridnykh intellektual'nykh komp'yuternyykh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*. Minsk: Bestprint [Bestprint], 2021.

[20] (2022, Nov) Ostis Metasystem. [Online]. Available: <https://ims.ostis.net>

[21] A. Zagorskiy, "Principles for implementing the ecosystem of next-generation intelligent computer systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 347–356, 2022.

[22] V. V. Golenkov, N. A. Guliakina, and O. E. Elisееva, *Opisanie yazyka SCP: Materialy po matematicheskomu obespecheniu EVM [Description of the SCP language: Computer Software Materials]*. Minsk : Instityt tech. kibernetiki Akademii nauk Belarusi, 1995, (In Russ.).

[23] A. Kroschchanka, V. Golovko, E. Mikhno, M. Kovalev, V. Zahariev, and A. Zagorskiy, "A Neural-Symbolic Approach to Computer Vision," in *Open Semantic Technologies for Intelligent Systems*, V. Golenkov, V. Krasnoproshein, V. Golovko, and D. Shunkevich, Eds. Cham: Springer International Publishing, 2022, pp. 282–309.

[24] V. Zahariev, D. Shunkevich, S. Nikiforov, and E. Aarov, "Intelligent voice assistant based on open semantic technology," in *Open semantic technologies for intelligent systems*, V. Golenkov, V. Krasnoproshein, V. Golovko, and E. Azarov, Eds. Cham: Springer, 2020, pp. 121–145.

[25] A. Goylo and S. Nikiforov, "Means of formal description of syntax and denotational semantics of various languages in next-generation intelligent computer systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 99–118, 2022.

[26] "Introducing ChatGPT," Available at: <https://openai.com/blog/chatgpt>, (accessed 2023, March).

[27] D. Shunkevich, "Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 119–132, 2018.

[28] V. V. Golenkov, D. V. Shunkevich, I. T. Davydenko, and N. A. Grakova, "Principles of organization and automation of the semantic computer systems development," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 53–90, 2019.

[29] A. Zagorskiy, "Factors that determine the level of intelligence of cybernetic systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 13–26, 2022.

Интеграция сторонних функциональных сервисов на унифицированной семантической основе

Загорский А. Г.

В работе представлены принципы, модели и средства интеграции различных сервисов в цифровую экосистему, основанную на смысловом представлении информации. Продемонстрирован пример реализации данного подхода с использованием Технологии OSTIS. Полученные результаты свидетельствуют о потенциальных преимуществах использования семантического представления информации для увеличения интероперабельности и адаптивности компьютерных систем в рамках единой цифровой экосистемы.

Received 29.03.2023

Integration of Large Language Models with Knowledge Bases of Intelligent Systems

Kseniya Bantsevich and Mikhail Kovalev
Vitaly Tsishchanka and Natallia Malinovskaya
*Belarusian State University of
Informatics and Radioelectronics*
Minsk, Belarus

Email: ksusha.bantsevich@gmail.com, michail.kovalev7@gmail.com
vitalik.tsishchanka@gmail.com, natasha.malinovskaya.9843@gmail.com

Aliaksei Andrushevich
Belarusian State University
Minsk, Belarus
andrushevich@bsu.by

Abstract—In the article, an approach for integration of large language models with knowledge bases of intelligent systems is proposed. The described approach aims at overcoming the disadvantages of large language models. A dialog system implementing the proposed approach is described.

Keywords—knowledge, knowledge base, conversational interface, dialogue, large language model, intelligent system, integration.

I. INTRODUCTION

With the development of *Artificial intelligence*, natural language text generation models are becoming increasingly popular and accessible. By now, neural network models have made significant progress in this area [1], [2]. In addition, *context-dependent information systems* have become widespread, using the concept of context [3].

Currently, one of the most popular approaches to natural language text generation is the use of *large language models* [4], [5], which are models consisting of neural networks with many parameters trained on a large amount of unlabeled text.

The main purpose of creating such models is to improve the interaction between users and computer systems by allowing the user to ask questions and get natural and understandable answers in their own language.

Such models have a number of advantages, such as:

- providing specific expanded responses to user requests;
 - almost an unlimited ability to make repeated changes to the user request until they are satisfied with the answer;
 - memorizing the context of user interaction and providing information related to those responses, using that context;
 - the ability to change the style and nature of responses depending on instructions;
 - a large training sample that allows giving the most plausible answers;
 - the ability to form a response in the selected language.
- Despite the above advantages of models of this type, they also have a number of disadvantages:
- Generation of unreliable answers.
Due to the lack of verification of the sources used and the lack of a coherent knowledge base, a model can respond seriously and in great detail to completely meaningless questions, have errors in reasoning, and draw incorrect conclusions, even though the resulting text in the model output may look coherent and convincing.
 - There is no mechanism for constantly updating the model with the latest knowledge.
At present, there is still no automatic learning mechanism for constantly updating the model with the latest knowledge.
 - There is the lack of ability to work with context.
Despite the fact that such models are context-aware systems and take context into account when generating answers, models still do not always have a sufficiently complete understanding of the context and, as a result, may produce incorrect answers.
 - Sensitivity to formulations.
In order to get the right answer to the question, it is necessary to clearly and concisely formulate the request, since even small changes in the message can lead to drastically different answers.
 - The influence of natural languages (English, German, Russian, etc.) on responses.
The authors' recent practical experiments have shown that responses to the same user queries in different natural languages can vary significantly in contents, form, and layout.
 - Inability to explain the answers.
The represented type of models uses a large amount of data, which it "remembers" from the training set. However, it is not clear how this data was used to make a decision and which rules were applied to

generate the answer; the models themselves cannot explain their answer.

The modern development of *Artificial intelligence* follows the path of building intelligent computer systems of a new generation [6] that are capable, among other things, not only of solving problems from various subject domains but also of explaining their solutions. The disadvantages represented above do not allow such systems to be built based only on *large language models*. On the contrary, intelligent systems of a new generation are built on the basis of unified representation in a single knowledge base of problems, subject domains of these problems, and methods solving these problems. Thus, *large language models* can and must become one of the powerful problem-solving methods for intelligent systems [7] but in no way replace the entire intelligent system [8].

From the perspective of intelligent systems of a new generation, dialog is the process of solving a large number of problems, which require a wide variety of *knowledge* and problem-solving methods. From the perspective of such systems, the context is the *knowledge* needed to solve the problems.

The purpose of this work is to develop an approach for integrating *large language models* with *knowledge bases* of intelligent systems. The implementation of the proposed approach is considered on the example of a specific intelligent dialog system.

II. PROPOSED APPROACH

To preserve context and provide better interaction with the user, the use of *knowledge bases* is one of the most effective approaches. A *knowledge base* can be a source of reliable *knowledge*, which can be used to give context to *large language models*.

However, to give the correct context with the *knowledge base* in a free dialog, the following is necessary:

- the ability to quickly localize the *subject domains* within which the dialog takes place;
- extracting the necessary *knowledge* from *subject domains*;
- placing the *knowledge* extracted from the dialog into their corresponding *subject domains*;
- create new *subject domains*, in case there are no suitable ones in the used *knowledge base* [9];
- etc.

Thus, in order to solve the problem, it is required to have a well-structured *knowledge base* [10] with a set of tools for processing and extracting *knowledge* from it.

The proposed approach implies the use of the *OSTIS Technology* and its basic principles [11]. Intelligent systems developed with the use of the *OSTIS Technology* are called *ostis-systems*. Any *ostis-system* consists of a *knowledge base*, a *problem solver*, and a *user interface*.

The approach to the development of *knowledge bases* of *ostis-systems* is based on the following basic principles of the *OSTIS Technology*:

- the basis of the *knowledge base* in the *OSTIS Technology* is a hierarchical system of *subject domains and ontologies*;
- orientation toward a semantic representation of *knowledge*;
- unification of representation and consistency of different types of *knowledge* and *problem-solving methods*;
- integration and convergence of various *problem-solving methods* in a single *knowledge base* in order to ensure consistency of the semantics for this set of methods.

As noted earlier, *knowledge bases* can be used as a source of context for systems built on the basis of large language models. Consequently, the use of systems built using the *OSTIS Technology* will allow building a context-aware dialog system due to the availability of well-structured *knowledge bases*.

To transfer context to such systems, *prompts* are used — an explicit indication of how and under what conditions the used model should respond. Modern software engineering has a separate direction dedicated to the development of methods and algorithms for the effective setting of prompts and context transfer — Prompt engineering [12], [13].

In this work, we propose to implement an *ostis-system*, the task of which is to select *prompts* based on knowledge about:

- the subject domains involved in the process of the dialog (i.e., the history of the dialog);
- the participants of the dialog;
- the place of the dialog;
- etc.

According to this approach, the following subject domain is developed:

Subject domain of prompts

- ∈ *subject domain*
- ⇐ *private subject domain**:
- Subject domain of statements*
- ⊃ *maximum class of explored objects'*:
prompt
- ⊃ *non-maximum class of explored objects'*:
 - *static prompt*
 - *dynamic prompt*
 - *atomic prompt*
 - *non-atomic prompt*
- ⊃ *explored relation'*:
 - *sc-structure equivalent to the statement**

prompt

- ⊃ *statement*
- ⇒ *explanation**:
[**prompt** — statement describing instructions, inputs for the natural language model]
- ⇒ *subdividing**:
 - *static prompt*
 - *dynamic prompt*
- ⇒ *subdividing**:
 - *atomic prompt*
 - *non-atomic prompt*

static prompt

- ⇒ *explanation**:
[**static prompt** is a prompt, whose natural language translation does not have an sc-structure equivalent to the prompt.]

dynamic prompt

- ⇒ *explanation**:
[**dynamic prompt** is a prompt, whose natural language translation has an sc-structure equivalent to the prompt.]

atomic prompt

- ⇒ *explanation**:
[**atomic prompt** is a prompt, which does not include other prompts.]

non-atomic prompt

- ⇒ *explanation**:
[**non-atomic prompt** is a prompt, which includes other prompts.]

sc-structure equivalent to the statement*

- ⇒ *first domain**:
statement
- ⇒ *second domain**:
sc-structure

In Figure 1, an example of a *dynamic prompt* with an sc-structure equivalent to this prompt is shown. This structure is a construction in the *knowledge base*, and the statement is a translation of this construction into a pseudo-natural language.

In the proposed system, the *prompts* are used to solve two problems:

- receiving a reply to a user message;
- replenishing the knowledge base with the knowledge gained through the dialog process.

In order to get a response to the user message, it is necessary to form a *static prompt* — instructions on how and in what conditions to respond to the received message, and it is also necessary to form a *dynamic*

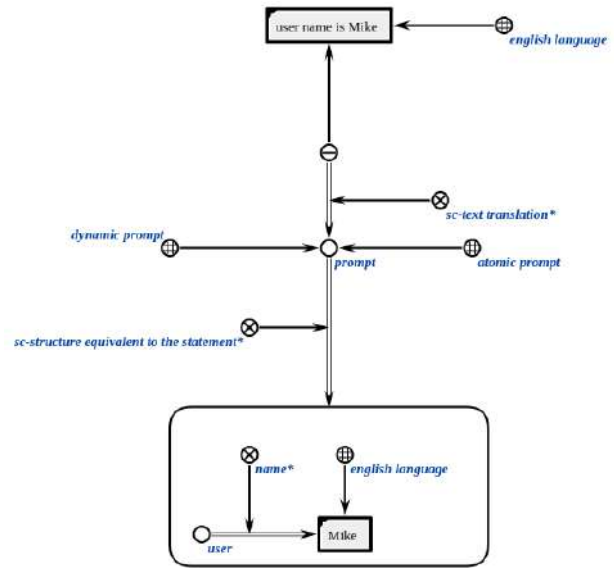


Figure 1. Dynamic prompt with an sc-structure equivalent to this prompt

prompt — knowledge from the *knowledge base*, which will be useful to respond to the message.

Therefore, for the problem of getting an answer to the message, two actions are highlighted:

- action. form a static prompt;
- action. form a dynamic prompt.

action. form a static prompt implies forming a *non-atomic static prompt* from all *atomic prompts* in the *knowledge base* of the system. Thus, the resulting *non-atomic static prompt* may depend on the particular user of the system, based on their preferences and current requirements.

action. form a dynamic prompt implies searching for all the *knowledge* in the *knowledge base* of the system that corresponds to the context within which the dialog is conducted. After obtaining all the necessary knowledge, these constructions are translated into a pseudo-natural language.

In general, a *dynamic prompt* is formed based on the *full semantic neighborhood* of the entity, which is the argument of the described action.

In order to constantly replenish the *knowledge base* in the process of the dialog, it is necessary to extract *knowledge* from the user messages and load them into the *knowledge base*. For this purpose, within the problem solver of the described system, an action is implemented whose task is to transform the natural-language text of the messages into *knowledge base* constructions.

To translate natural language text messages into a *knowledge base* construction (i.e., to extract meaning from the message), within the proposed approach, *large language models* are used. The essence of this approach lies in representing the key entities used in the message,

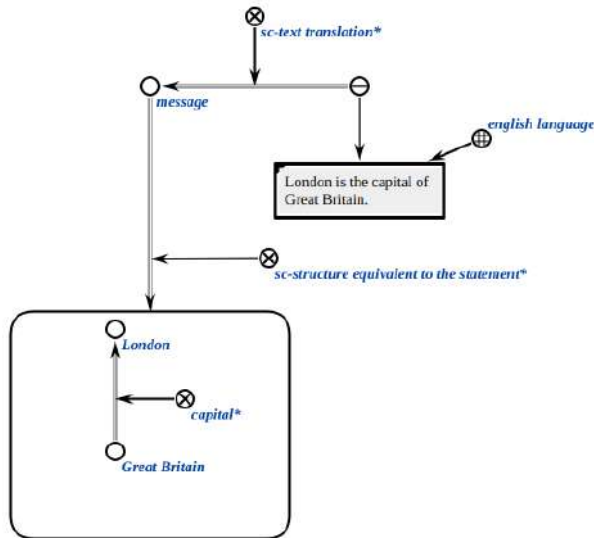


Figure 2. Example of a message with its equivalent construction

as well as the relations between such entities, using constructions in the form of subject-verb-object [14].

In Figure 2, an example message and its equivalent construction in the *knowledge base* are shown.

Thus, any natural language text can be transformed into *knowledge base* constructions.

The represented hierarchy of actions corresponds to the collective of agents [15] within the proposed *ostis-system*.

In Figure 3, an example of a dialog between the system and the user is demonstrated.

In Figure 4, the user *semantic neighborhood* obtained from the dialog above is shown.

The constantly expanding *semantic neighborhood* of the user, as well as all the entities that are raised during the dialog, allows storing in the *knowledge base* all the information obtained during the dialog.

As stated earlier, the *dynamic prompt* is formed on the basis of the *full semantic neighborhood* of the entity that is the argument of the described action. Based on the received meaning of the message, the *dynamic prompt* will contain all the *knowledge* about the topic of the dialog. This solution allows the system to better “memorize” the contexts of the dialog and use them for a more accurate and coherent response, taking into account the previously mentioned entities and understanding how they are related to the current topic of communication. This makes the dialog with the system more productive and natural, similar to a conversation with a human who remembers all the details of previous conversations, without the user having to remind the system about the context of the conversation.

III. DEMONSTRATION OF RESULTS

Contextual dialog control was tested as follows:

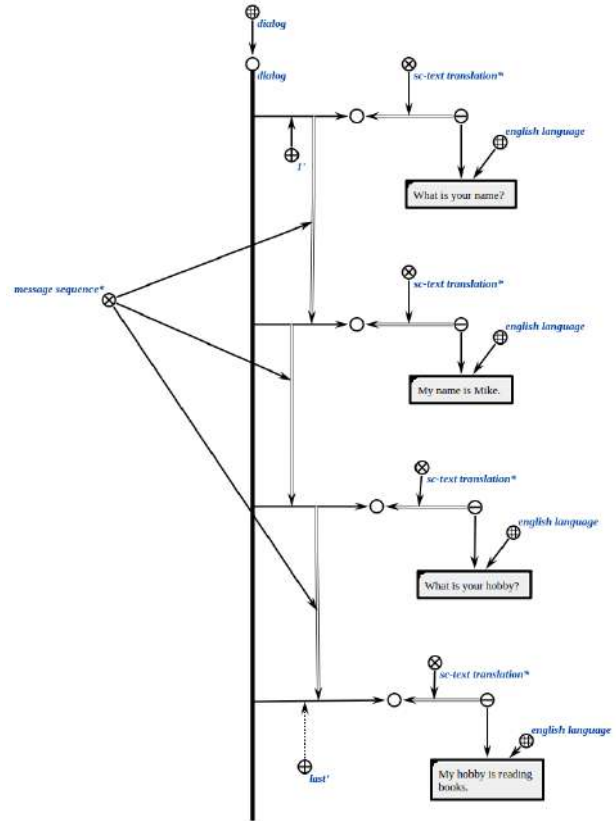


Figure 3. Example of a dialog

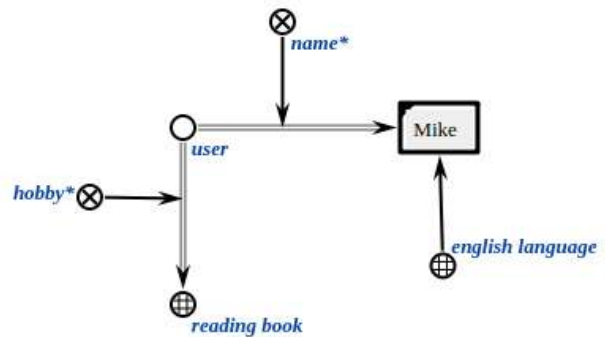


Figure 4. Obtained semantic neighborhood of the user

- Inputting any specific information about the user at the beginning of the dialog (name, age, location, etc.).
- Conducting a dialog on a variety of topics over a period of time. Since the ability of large language models to hold a dialog context depends on the configuration and available memory on the server where the model is operating, the time interval after which the model may “forget” the given information varies.
- Asking a question about the information mentioned in the dialog from the first item.

Let us consider the example of a communication between a user and:

- a system using a large language model [5];
- a system using a large language model powered by the knowledge base.

In Table I, the results of a user communication with a system using a large language model are shown.

Table I
RESULTS OF CONVERSATION USING A LARGE LANGUAGE MODEL

Nº	User message	System response
1	Hello. My name is Mike and I am from Suriname.	Hello, Mike! It's nice to meet you. How may I assist you today?
2	I don't know what to do.	I'm sorry to hear that you're feeling unsure about what to do. Can you tell me a bit more about your situation? What's been on your mind lately?
...
n-1	I can't remember my name.	If you are unable to recall your name, it may be helpful to take a moment to try to relax and clear your mind. You could also try focusing on other details about yourself, such as your age, where you are located, or what you were doing before you forgot your name. If you continue to have difficulty remembering your name, it may be a good idea to seek medical assistance from a healthcare professional who can help you determine the cause of your memory loss and provide appropriate treatment.
n	What is my name?	As an AI language model, I don't have access to personal information such as your name. I apologize for the confusion earlier. Is there anything else I can help you with?

As can be seen from Table I, in the process of a dialog, the represented system could not remember the username that was mentioned at the beginning of the dialog. This fact leads to the need for constant repetition of information, the exact wording of queries, which makes it difficult to qualitatively and smoothly get answers to the queries. In Table II, the results of a user communication with a large language model powered by a knowledge base are demonstrated.

As per the proposed approach, since any information mentioned during the dialog was placed in the *knowledge base*, the system was able to easily answer the posed question, as seen in Table II. The presence of the *knowledge base* allows the system to provide more accurate and concise answers, and also allows the user to form their queries in various ways, as such a system

Table II
RESULTS OF A USER COMMUNICATION WITH A LARGE LANGUAGE MODEL POWERED BY A KNOWLEDGE BASE

Nº	User message	System response
1	Hello, my name is Mike and I am from Suriname.	Great to meet you, Mike! What brings you to our platform today?
2	I don't know what to do.	I understand. Can you tell me a bit more about what's been going on and why you're feeling unsure about what to do?
...
n-1	I can't remember my name.	Is there anything I can do to help you remember your name?
n	What is my name?	Your name is Mike. You mentioned it in your previous response. Is there anything else you would like to know or talk about?

is not susceptible to incorrect responses due to imprecise phrasing of words in a sentence.

IV. CONCLUSION

The implementation of the described approach for integrating *large language models* with *knowledge bases* of intelligent systems allows significantly improving the quality of a dialog with the user in such systems. This occurs due to overcoming the drawbacks of *large language models*, such as the inability to take into account the context of the dialog on a long-term basis and sensitivity to the user formulations.

The solution is achieved by transferring the dialog context from the natural-language form represented by the direct message history to the formalized form represented in a *knowledge base* structured on the basis of the ontological approach. In such a *knowledge base*, the system stores the dialog context, and the system problem solver makes it possible to localize this context and transform it into the form required by *large language models*.

Moreover, since intelligent systems consider a dialog as a process of problem solving and a *large language model* is one of the methods of its solution, it becomes possible to use *large language models* to solve various problems, not necessarily related to the dialog. Thus, the proposed approach for integration adds another powerful problem-solving method to the arsenal of intelligent systems.

The proposed approach is tested in the described intelligent dialog system, which demonstrates a higher level of context-awareness than systems using only large language systems without a *knowledge base*.

A further development of this work will among others consist in solving the problem of verification of the answers given by *large language models* in the *knowledge base* of the intelligent system.

ACKNOWLEDGMENT

The authors would like to thank the research groups of the Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Department of Software Engineering of the Belarusian State University for their help in the work and valuable comments, in particular, Vladimir Golenkov and Daniil Shunkevich.

REFERENCES

- [1] C. Dong, Y. Li, H. Gong, M. Chen, J. Li, Y. Shen, and M. Yang, "A Survey of Natural Language Generation," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–38, Aug. 2023. [Online]. Available: <https://dl.acm.org/doi/10.1145/3554727>
- [2] Y. Kim, H. Lee, J. Shin, and K. Jung, "Improving Neural Question Generation Using Answer Separation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 6602–6609, Jul. 2019. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/4629>
- [3] A. Andrushevich and I. Vojteshenko, "Towards semantic representation of the iot ecosystem and smart home applications," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, vol. 6, p. 405–410, 2022.
- [4] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, "Large language models in machine translation," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, url = "<https://aclanthology.org/D07-1090>", Jun. 2007, pp. 858–867.
- [5] A. Tamkin, M. Brundage, J. Clark, and D. Ganguli, "Understanding the capabilities, limitations, and societal impact of large language models," *CoRR*, vol. abs/2102.02503, 2021. [Online]. Available: <https://arxiv.org/abs/2102.02503>
- [6] V. V. Golenkov and N. A. Gulyakina, "Next-generation intelligent computer systems and technology of complex support of their life cycle," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 27–40, 2022.
- [7] M. V. Kovalev, "Convergence and integration of artificial neural networks with knowledge bases in next-generation intelligent computer systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 173–186, 2022.
- [8] (2023, March) Opinion | Noam Chomsky: The False Promise of ChatGPT. [Online]. Available: <https://www.nytimes.com/2023/03/08/opinion/noam-chomsky-chatgpt-ai.html>
- [9] R. Patra and S. K. Saha, "A hybrid approach for automatic generation of named entity distractors for multiple choice questions," *Education and Information Technologies*, vol. 24, no. 2, pp. 973–993, Mar. 2019. [Online]. Available: <http://link.springer.com/10.1007/s10639-018-9814-3>
- [10] K. Bantsevich, "Structure of knowledge bases of next-generation intelligent computer systems: a hierarchical system of subject domains and their corresponding ontologies," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 87–98, 2022.
- [11] V. Golenkov, N. Gulyakina, I. Davydenko, and D. Shunkevich, "Semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem i semanticheskie associativnye komp'yutery [Semantic technologies of intelligent systems design and semantic associative computers]," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 42–50, 2019, (In Russ.).
- [12] (2023, March) Introduction to prompt engineering: Key concepts and use cases. [Online]. Available: <https://www.mlq.ai/introduction-to-prompt-engineering/>
- [13] E. Jiang, K. Olson, E. Toh, A. Molina, A. Donsbach, M. Terry, and C. J. Cai, "Promptmaker: Prompt-based prototyping with large language models," in *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: <https://doi.org/10.1145/3491101.3503564>
- [14] D. Crystal, *The Cambridge Encyclopedia of Language*, 2nd ed. Cambridge: Cambridge University Press, 1997.
- [15] D. Shunkevich, "Agentno-orientirovannye reshateli zadach intellektual'nyh sistem [Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems]," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, pp. 119–132, 2018.

Интеграция больших языковых моделей с базами знаний интеллектуальных систем

Банцевич К. А., Ковалев М. В.,
Тищенко В. Н., Малиновская Н. В.,
Андрюшевич А. А.

В данной работе предлагается подход к интеграции больших языковых моделей с базами знаний интеллектуальных систем. Описанный подход направлен на преодоление недостатков больших языковых моделей. Описана диалоговая система, реализующая предлагаемый подход.

Received 13.03.2023

OSTIS Technology Integration with Third-party NLP Services

Alena Cherkas, Aliaksandr Kupo

Francisk Skorina

Gomel State University

Gomel, Belarus

Email: alenacherkas.iss@gmail.com, kupo@gsu.by

Abstract—This article is devoted to the integration of OSTIS Technology with third-party services based on neural networks. As an example of integration, the task of keeping minutes of meetings is considered, the solution of which is based on transcription and summarization. Transcription is carried out using the Whisper service, summarization — retext.ai.

Keywords—OSTIS Technology integration; neural networks; transcription task; summarization task; Natural Language Processing

I. INTRODUCTION

After the appearance of ChatGPT, its analogues and other services based on neural networks in the general availability, a large number of opportunities appeared in the professional activities of representatives of different professions, scientific activities, and personal life. In particular, the scope of application of OSTIS technology has expanded [1].

Integration of OSTIS Technology with third-party services based on neural networks allows you to create universal components adapted to reuse. Thus, for example, when using ChatGPT, it becomes possible not only to get an answer to a given question, but also to save it by formalization, that is, writing on SC-code [2], thereby replenishing the global knowledge base. After that, the system using this knowledge base will be able to process the information received to solve other tasks [3].

Thus, when integrated with ChatGPT or its analogues, the Nika dialog platform, operating on the basis of OSTIS, can access the advantages of dialog systems representing a neural network, while maintaining the advantages of OSTIS.

ChatGPT can also generate insufficiently correct responses. If the user does not have enough knowledge in the field of the question being asked, then it is almost impossible to distinguish truth from lies in the neural network response. This reduces the quality of work with the service and its capabilities. The integration of ChatGPT with OSTIS allows you to identify incorrect information in the responses of the service.

The task of integrating ChatGPT with OSTIS, namely the Nika dialog system, has already been solved. However, OSTIS Technology can be integrated with services of

completely different orientation [1], working on the basis of neural networks. Thus, it is possible to optimize their work, expand functionality, and reduce the number of shortcomings [4].

II. OVERVIEW OF EXISTING APPROACHES

The possibility of integration is demonstrated by the example of creating a meeting protocol. To do this, the tasks of transcription and summarization will be solved. To convert an audio recording into a text format, there are many services based on neural networks. One of the available services is Whisper from OpenAI.

Whisper is an Automatic Speech Recognition (ASR) system trained on more than 650 thousand hours of multilingual and multitasking controlled data collected from the Internet. The developers of the service note that using such a large and diverse data set leads to increased resistance to accents, background noise and technical language. In addition, Whisper allows you to perform transcription in several languages, as well as translation from these languages into English.

Whisper is built on the transformer architecture, stacking encoder blocks and decoder blocks with the attention mechanism propagating information between both. It will take the audio recording, split it into 30-second chunks and process them one by one. For each 30-second recording, it will encode the audio using the encoder section and save the position of each word said, and leverage this encoded information to find what was said using the decoder.

The decoder will predict what user call tokens from all this information, which are basically each words being said. Then, it will repeat this process for the next word using all the same information as well as the predicted previous word, helping it guess the next one that would make more sense.

The overall architecture is a classic encoder-decoder that is similar to GPT-3 and other language models. OpenAI open-sourced their code and everything instead of an API, so everyone can use Whisper as a pre-trained foundation architecture to build upon and create more powerful models for his necessities.

retaining the most important information and meaning. The goal is to produce a summary that accurately represents the content of the original text in a concise form.

During training, the neural network learns to identify important phrases, sentences, and ideas in the full-length text, and generate a shorter summary that accurately captures the main points.

There are different approaches to text summarization, including extractive methods that identify and extract important sentences or phrases from the text, and abstractive methods that generate new text based on the content of the original text.

One approach for text summarization is to use an encoder-decoder architecture with an attention mechanism. The encoder takes in the full-length text and encodes it into a fixed-length vector representation. The decoder then generates the summary based on this encoded representation. The attention mechanism allows the decoder to focus on the most important parts of the encoded representation while generating the summary.

A major limitation of this network is its incapacity to extract significant contextual connections from extended semantic sentences. When a lengthy text contains contextual relationships within its substrings, the basic seq2seq model cannot recognize those contexts. As a result, the model's performance is somewhat compromised, leading to reduced accuracy.

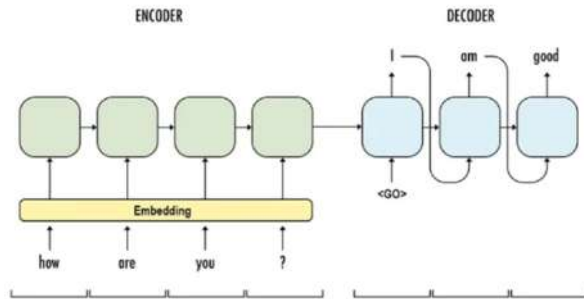


Figure 5. Encoder-decoder model [7]

The encoder network extracts or obtain features from input data and products internal state vector or context vectors, which summarizes the input sequence. In STM networks, these vectors are referred to as hidden state and cell state vectors. The decoder network interprets the context vector produced by the encoder and generates the output sequence. The final call of the encoder produces the context vector that serves as the input to the first call of the decoder network. The decoder then use this context vector along with the initial states to start generating the output sequence, and the outputs are considered for future predictions.

Attention is a type of enhancement to the current sequence-to-sequence network that addresses its limitations. It derives its name from its ability to highlight important parts of a sequence. By considering only a few relevant items in the input sequence, the output sequence becomes conditional and is guided by weighted constraints. These constraints are the contexts that receive attention, which are subsequently used for training and predicting the desired results.

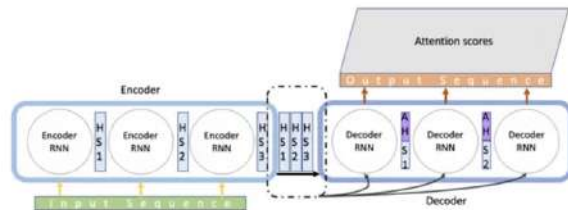


Figure 6. Attention architecture [7]

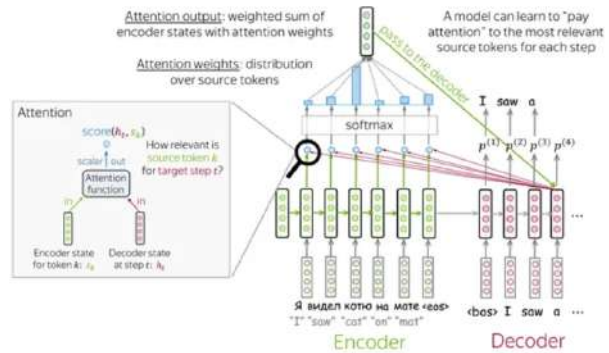


Figure 7. Attention model [7]

The attention-based sequence-to-sequence model requires considerable computational power, but its performance is superior to that of the traditional sequence-to-sequence model. Additionally, the model can demonstrate how attention is focused on the input sequence when predicting the output sequence. This is helpful in comprehending and identifying the specific input-output pairs and the extent to which the model considers them.

The model aims to create a context vector that is tailored for each output time step by selectively filtering the input sequence [7].

Another approach is to use reinforcement learning, where the neural network is trained to generate summaries that maximize a reward function based on the quality of the summary. This approach can be useful when there is not enough labeled data available for supervised learning [8].

The key to neural network text summarization is in finding a balance between generating a summary that is short and concise while still accurately conveying the main ideas of the full-length text.

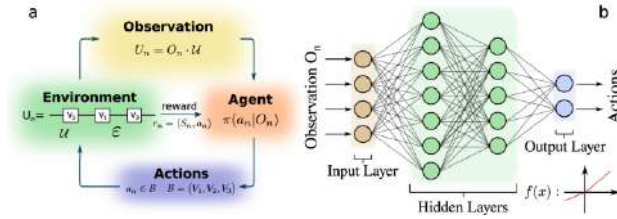


Figure 8. Rainforest learning model [8]

It doesn't matter which service to choose for summarization. But one of the most convenient services — ReText.AI. ReText.AI's summarization feature can help shorten text content while preserving its meaning. It can create shorter descriptions or summaries, simplify complex information for easier comprehension, and adapt the text to social networks, saving authors time and increasing productivity.

III. PROPOSED APPROACH

The purpose of the work is to expand the functionality, increase the possibilities, ways of using the meeting protocol, find the best solution and optimize the process of finding errors and inaccuracies of the result.

One of the options for achieving the goals mentioned above is integration with OSTIS Technology. This way the meeting is presented in the knowledge base, its specification necessarily includes information about the participants, the start and end time of the meeting, the topic of the meeting, as well as an audio recording of the meeting is stored in the form of an ostis-system file.

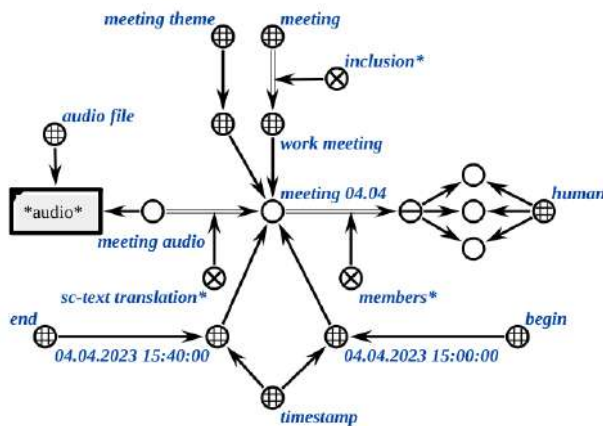


Figure 9. Text format of the meeting recording in SCg

To solve the problem under consideration, the OSTIS-system problem solver applied a strategy for decomposing the problem into subtasks. Further, problem solver analyzed that for summarization it is necessary to obtain a natural-language representation of the meeting in a text format and then use a summarization agent for it. Therefore, the speech-to-text translation agent is called.

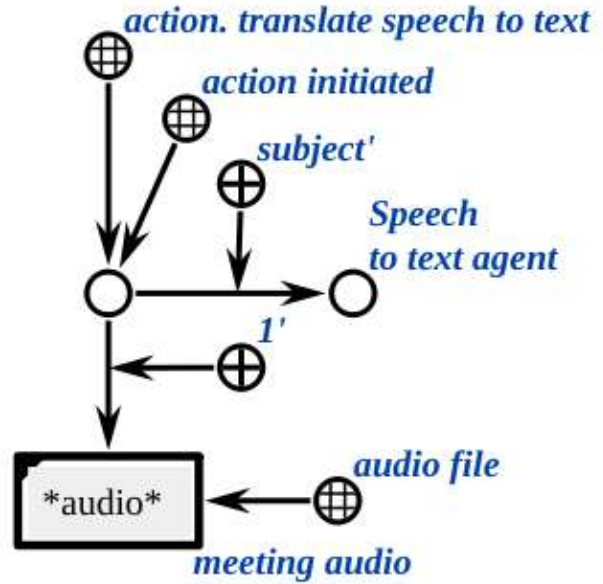


Figure 10. Speech-to-text translation agent input

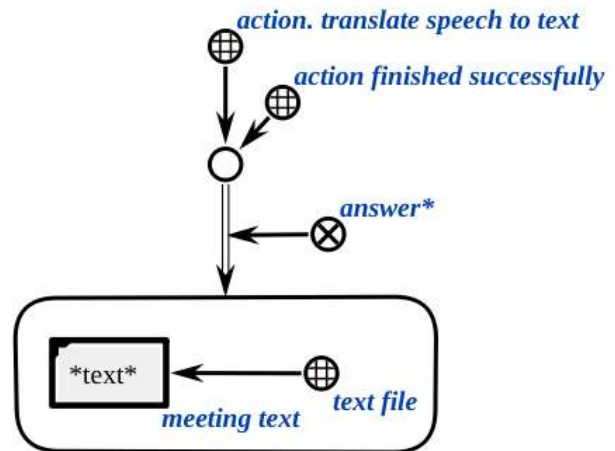


Figure 11. Speech-to-text translation agent output

The audio recording of the meeting is transcribed into a text format using the Whisper service. Next, a structure is formed in the knowledge base that connects the same meeting sign with its different external representations, that is, audio and text. Thus, from the audio format at the input, a text format at the output is obtained.

Thus, the task of transcription is solved. Further, to obtain the minutes of meetings, it is necessary to solve the summarization problem. To do this, the summarization agent is called. Summarization agent is the code that accesses the service and immerses its response into the knowledge base of the OSTIS system. The input for the service is also taken from the knowledge base, that is, the input is the result of solving the task of transcribing using the Whisper service and translating the audio recording

of the meeting into the text format.

The text received from the audio recording is fed to the input of the summarization agent. And the result of the agent's work is a text reduced to the specified number of words. Which, at the same time, retains its meaning.

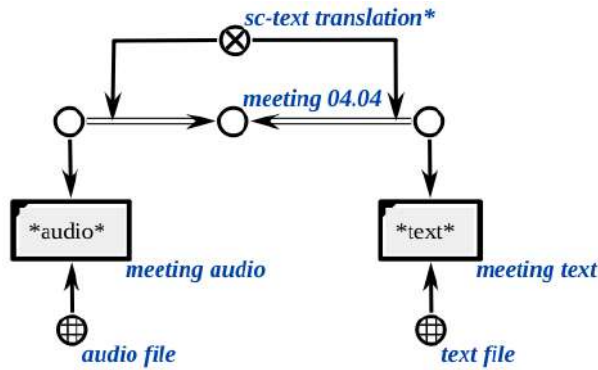


Figure 12. Speech-to-text translation agent

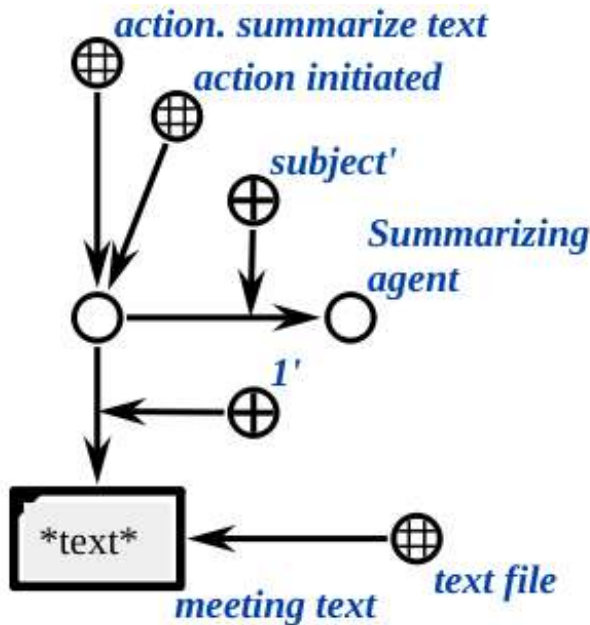


Figure 13. Summarization agent input

In fact, the above actions occur with knowledge base constructs, and text and sound are done only by NLP agents. NLP agents are natural language processing agents, computer programs that are designed to understand and respond to natural language input. These agents use a combination of techniques from computer science, linguistics, and artificial intelligence to analyze and interpret human language. NLP agents can be used in a wide range of applications, such as chatbots, virtual assistants, language translators, speech recognition systems, and sentiment analysis tools. These agents can help automate tasks, provide personalized recommendations, and improve the

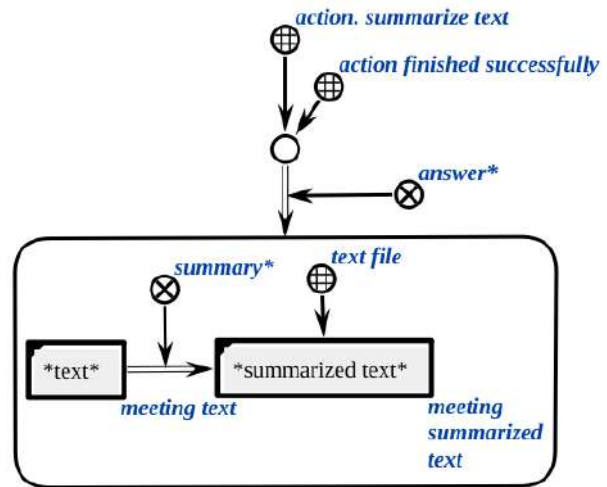


Figure 14. Summarization agent output

overall user experience by allowing humans to interact with technology in a more natural way.

NLP agents typically involve a combination of several components, including speech recognition, natural language understanding, natural language generation, and dialogue management. These components work together to enable the agent to understand and respond to human language input in a meaningful way.

Everything that is developed using OSTIS technology should be presented, first of all, in the form of knowledge base constructions, that is, described in SC-code. Including the meaning of the text and sound. And only then, in order to display the already processed information to the user, a natural language interface is used, that is, NLP. The text-to-speech translation agent is a classic NLP task, as is text-to-speech translation.

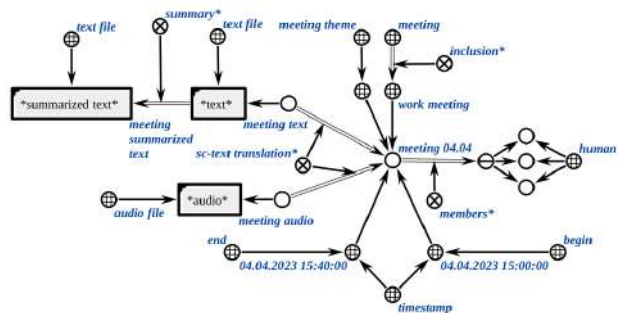


Figure 15. SCg-result of making a call protocol

OSTIS technology in the creation of a meeting protocol allows user to integrate various problem-solving models, achieve compatibility, effectively use knowledge bases to solve transcription and summarization tasks. OSTIS makes it possible not only to summarize the results of the meeting, but also to highlight the meaning of what was said during the meeting, process the context, analyze

the content, identify problem areas and contradictions, and offer solutions to them [9].

This will bring the use of the developed meeting protocol to a new level, because the data will be stored in a single OSTIS memory [10]. Thus, any OSTIS application will have access to them, will be able to use and explain. Such a system will be easy to integrate with other neural networks, expand the functionality by using and reusing existing developments.

IV. CONCLUSION

The paper proposes one of the solutions for keeping minutes of meetings based on solving the problems of transcription and summarization. The article describes the existing approaches to solving such problems, and also offers a solution for integrating existing approaches with OSTIS Technology, the advantages of such a solution are marked.

The results obtained will allow not only to get a brief conclusion from the meeting, but also to create an intelligent system based on the knowledge base. The ready-made system allows to highlight the meaning of what was said during the meeting, process the content, analyze the content, identify problem areas and contradictions, and offer solutions to them.

After the project is implemented, OSTIS allows you to reuse the received solutions in other tasks and projects. And also, integrate the system with other applications. As a result, all applications developed on the basis of OSTIS technology have access to a common knowledge base.

REFERENCES

- [1] V. Golenkov, N. Guliakina, and D. Shunkevich, *Otkrytaya tekhnologiya ontologicheskogo proektirovaniya, proizvodstva i ekspluatatsii semanticheski sovmestimyykh gibridnykh intellektual'nykh komp'yuternyykh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*. Minsk: Bestprint [Bestprint], 2021, (In Russ.).
- [2] (2021, Jun) Ostis applications. [Online]. Available: <https://github.com/ostis-ai/sc-web>
- [3] O. Yara, A. Brazheyev, L. Golovko, L. Golovko, and V. Bashkatova, "Legal regulation of the use of artificial intelligence: Problems and development prospects," *European Journal of Sustainable Development*, vol. 10, p. 281, 02 2021.
- [4] (2022, September) IMS.ostis Metasystem. [Online]. Available: <https://ims.ostis.net>
- [5] (2015-2023) Whisper. [Online]. Available: <https://openai.com/research/whisper>
- [6] (2015-2023) Robust speech recognition via large-scale weak supervision. [Online]. Available: <https://cdn.openai.com/papers/whisper.pdf>
- [7] (2021, February) Understanding encoders decoders with attention based mechanism. [Online]. Available: <https://medium.com/data-science-community-srm/understanding-encoders-decoders-with-attention-based-mechanism-c1eb7164c581>
- [8] (2021, August) Quantum compiling by deep reinforcement learning. [Online]. Available: <https://www.nature.com/articles/s42005-021-00684-3>
- [9] (2021, Jun) Sc-machine. [Online]. Available: <https://github.com/ostis-ai/sc-machine>
- [10] (2021, Jun) OSTIS. [Online]. Available: <https://github.com/ostis-ai>

Интеграция Технологии OSTIS со сторонними NLP сервисами

Черкас Е. О., Купо А. Н.

Данная статья посвящена интеграции Технологии OSTIS со сторонними сервисами, работающими на базе нейронных сетей. В качестве примера интеграции рассматривается задача ведения протокола совещаний, решение которой основано на транскрибировании и суммаризации. Транскрибация осуществляется с помощью сервиса Whisper, суммаризация — retext.ai.

Received 13.03.2023

Examples of Integrating Wolfram Mathematica Tools into OSTIS Applications

Valery B. Taranchuk
*Department of Computer Applications
and Systems
Belarusian State University
Minsk, Republic of Belarus
taranchuk@bsu.by*

Vladislav A. Savionok
*Department of Software
for Information Technologies
Belarusian State University
of Informatics and Radioelectronics
Minsk, Republic of Belarus
v.savenok@bsuir.by*

Abstract—Within the concept of convergence and unification of intelligent computer systems of the new generation, technical solutions are discussed, examples of development and modernization, integration of Ecosystem OSTIS tools with Wolfram Mathematica (WM) computer algebra system (CAS) are provided.

On the example of integration with specialized complex of intellectual educational resource for the discipline “Computer Systems and Networks” the possibilities of using WM tools in ostis-system are discussed when solving problems related, in particular, to topology of info-communication networks. The application of WM tools for visualization of network topology, as well as emulation of the search for the optimal route for data transmission is shown.

Keywords—OSTIS technology, ostis-system, computer algebra systems, Wolfram Mathematica, Computer Systems and Networks, graph visualization

I. INTRODUCTION

Following the assessment of the current state of work in the field of Artificial Intelligence (AI), it is possible to affirm active local development of various directions (non-classical logics, formal ontologies, artificial neural networks, machine learning, soft computing, multi-agent systems, etc.), however, a comprehensive increase in the intelligence of modern intelligent computer systems does not occur [1].

The key reasons of methodological problems of current state of Artificial Intelligence, as well as a number of actions required to solve them are outlined in [1]. What actions are needed to improve the current state? First of all, it is necessary to converge and integrate all directions of Artificial Intelligence and corresponding construction of a general formal theory of intelligent computer systems (ICS), the transformation of modern variety of frameworks for development of different ICS components into a single technology of complex design and support of the full life cycle of these systems, which guarantees the compatibility of all developed components, as well as the compatibility of the ICS as independent subjects, interacting between each other. Convergence and unification of new generation of in-

tellectual computer systems and their components is required.

Convergence and unification of new generation intelligent computer systems and their components is necessary. At the same time, convergent solutions basically mean optimized complexes that include everything necessary to solve AI tasks, organized and configured for efficient use of information resources, simplification of implementation processes, meeting the requirements of maximum performance, availability of intelligent interface, simple and understandable for all categories of users.

Supporting the outlined concepts, we note that such problems can be effectively solved by developing, improving, regularly updating the content of intelligent systems by incorporating CAS means. Below are a few methodological and technical solutions for integration of different types of knowledge, implemented by inclusion of Wolfram Mathematica functions in the ostis-system of support and maintenance of the teaching process of one of the basic disciplines in high school, illustrated by examples.

II. CONCRETIZATION AND VARIANTS FOR INTEGRATING THE ECOSYSTEM OSTIS WITH CAS

Integration of the Ecosystem OSTIS with any service means the ability to use the functionality of the service to change the internal state of the system’s knowledge base. Within Ecosystem OSTIS full and partial integration levels are acceptable.

According to the Technology, full integration of the Ecosystem OSTIS with any service implies the possibility of executing service’s function at platform-independent level using SCP language. That is, the task of integration of such a service is reduced to allocation of a graph structure processing algorithm and its implementation within a system’s knowledge base. As a result of such integration there is no need to use a third-party service, in fact, an Ecosystem component is used.

Partial integration means changing the state of the system’s knowledge base at the stages of service function

execution. The depth of integration can vary. In some cases, a service can refer to the knowledge base to get additional information or to record intermediate results. In the simplest case, a knowledge base can change only once, after a result of the service's function is received. In case of partial integration it is supposed that particular ostis-systems are to play the role of system integrators of included resources and services of other computer systems, as the level of intelligence of ostis-systems allows them to specify the computer systems being integrated to a sufficient degree of detail and, consequently, to "understand" adequately what each of them knows and/or can do.

Separately, let us note that following the Technology, ostis-systems are capable (and it should be used) to coordinate the activities of a third-party resource and service sufficiently well, to provide a "relevant" search for the required component. The systems themselves can also perform the role of intelligent help-systems – assistants and consultants for efficient operations with functional capabilities, when the user interface is implemented with non-trivial semantics in the unique tasks of complex subject areas. Such help systems can be made intelligent intermediaries between the relevant computer systems and their users.

The systems themselves can also act as intelligent help systems. Their relevance is dictated by the high complexity of subject areas and the non-triviality of some unique tasks. Such conditions require the design of appropriate unique and nontrivial user interfaces with additional information support for their use. Such help systems can be made intelligent intermediaries between the relevant computer systems and their users, and the homogeneity of the technologies used ensures seamless integration with the existing system.

Solving the issues of data format coordination. A tedious problem of functional service integration when forming a digital ecosystem of multiple interacting services is the difference in data formats that participants of this Ecosystem work with. Two services, which imply data processing from one subject area, are likely to have different data formats. The problem of coordinating the data format of different services significantly complicates the development of the services themselves and leads to an increase in time costs. Such issues can be effectively solved using CAS import and export functions, such as in Wolfram Mathematica, which supports more than 100 data formats, including graphics, video, and more.

At the current stage of development and usage of the Ecosystem OSTIS, one of the priority areas appears to be the integration of the capabilities of computer algebra systems and intelligent learning systems constructed in ostis-application. The importance of this is due to the relevance, the requirements of the intellectualization of educational resources on the one hand,

and on the other – contents of computer algebra systems, which have an undoubted advantage and great opportunities for solving problems relevant to educational systems for virtually all natural-science and technical disciplines, involving the use of complex mathematical apparatus.

It can be stated that, despite the popularity of topics related to automation and intellectualization of educational activities in science disciplines and the development of appropriate computer systems, at the moment the market is practically lacking tested intellectual educational systems capable of self generating and solving various problems, and verifying the correctness of the solution provided by the user. As prototypes, there are some systems that consider non-trivial problems, such as geometry [2], [3] and graph theory [4]. But, to be fair, it should be noted that there is no intelligence in the mentioned systems (in fact, only a specific set of actions is implemented, the tasks are not generated in the applications themselves), there is no means of verification of solutions with even minor deviations of the design rules.

One of the variants for interaction between Ecosystem OSTIS and CAS can be approaches similar to the integration of artificial neural networks in ostis-systems (see [5]).

Developing the aforementioned implementations, the following methodological and technical solutions can be considered:

- Black-box integration, when the knowledge base of the ostis-system contains the specification of the used kernel function of the computer algebra system, as well as the specification of the method of calling this function (for example, specifying through which software interface the interaction with this external system is performed). This integration variant is the easiest to implement and generally has the advantages listed below. At the same time, this variant has a disadvantage that the ostis-system does not contain means of analysis and explanation of how a certain step of solving a problem that is realized by a used CAS function was taken.
- A tighter integration, in which a particular function is still a part of a third-party CAS, when not only the result of its performance is loaded to the knowledge base of ostisystem, but also all possible specification of it, e.g. explanation of the problem solution step, indication of particular algorithms and formulas which can be involved in the solution, description of possible alternative solution variants, evaluation of solution efficiency and so on. In this variant of integration, the ostis-system gets more opportunities of analysis and explanation of the problem solution process. (Note that this doesn't apply specifically to CAS Wolfram Mathematica, because it always has detailed explanations for all solutions, and allows

for step-by-step execution.)

- Full integration, which translates computer algebra functions in use from this system's internal language into the ostis-system. This variant is the most labor-intensive and complicated in terms of updating the capabilities of computer algebra systems in the corresponding ostis-systems taking into account their constant development. At the same time this integration variant, in comparison with the two previous ones, has an important advantage: it ensures platform-independent solution and allows using all the advantages of the approaches proposed within the OSTIS Technology in solving a concrete problem, in particular the possibility of multi-user parallel knowledge processing and the possibility to optimize a problem solution plan or its fragments directly during the solution.

The approach to solving the problems of intellectualization of educational activity, based on the integration of ostis-systems and computer algebra systems, has several advantages:

- When developing ostis-systems, the need to program many functions that have already been implemented and tested in CAS is eliminated. This is fundamental because computer algebra systems are developed by highly qualified specialists in the relevant fields, the implementation of similar functions in ostis-systems may require significant financial and time expenditures.
- A concrete ostis-system using individual functions of CAS, due to the approach to the development of hybrid problem solvers in OSTIS Technology, gets the possibility to *self plan* the course of problem solving provided that some of its steps are implemented by means of the attached functions. From the point of view of the approach proposed within the framework of OSTIS Technology, each function of the computer algebra systems, computer mathematics systems (CMS) becomes a *method* for solving problems of some class. This class of problems is described in the knowledge base of the ostis-system and allows it, when solving a concrete problem, to independently draw a conclusion about the expediency of applying one or another CAS function. Such integration with ostis-systems will make it possible to eliminate a possible disadvantage of individual computer algebra systems noted earlier (determined by which CAS are used – explained below in the overview of computer mathematics systems, conditions of their application and access to individual components).

We emphasize that these integration variants are not mutually exclusive and can be combined. In addition, integration can be deepened step by step taking into account the above advantages and disadvantages as well

as the relevance of using certain functions of computer algebra systems in solving specific tasks within the Ecosystem OSTIS and corresponding ostis-systems.

In general case, step-by-step integration of CAS with the Ecosystem OSTIS implies, as a minimum, description of the specification of the basic functions of the selected computer algebra system by means of the OSTIS Technology, in other words – development of the ontology of external functions. In case of Wolfram family systems, the process of developing such ontology can be automated due to the presence of the Wolfram Language formal language and good documentation of system functions.

Summarizing the above, we state: the integration of educational systems developed on the basis of OSTIS Technology and computer algebra systems will allow us to create systems with intelligent properties in a shorter time, and with the use of carefully developed (mathematically, algorithmically) and repeatedly tested tools.

III. FUNDAMENTALS, TERMINOLOGY. COMPUTER MATHEMATICS SYSTEMS, COMPUTER ALGEBRA SYSTEMS

In the mid-twentieth century, at the junction of mathematics and computer science, a fundamental scientific trend, computer algebra, the science of efficient algorithms for calculating mathematical objects, emerged and intensively developed. Synonyms for the term “computer algebra” are: “symbolic calculations”, “analytical calculations”, “analytical transformations”, and sometimes “formal calculations”. The field of computer algebra is represented by theory, technology, software tools. Applied results include developed algorithms and software for solving problems using a computer in which the original data and results take the form of mathematical expressions, formulas. The basic product of computer algebra became software computer algebra systems (CAS). The range of mathematical problems solvable with the help of CAS is constantly expanding. Considerable effort is devoted to developing algorithms for computing topological invariants of varieties, nodes, algebraic curves, cohomology of different mathematical objects, and arithmetic invariants of rings of integers in the fields of algebraic numbers. Another direction of modern research is quantum algorithms, which sometimes have polynomial complexity, while existing classical algorithms have exponential complexity.

Research and development of theoretical foundations and technologies for implementing methods and software implementations of computer algebra tools continues. Terms, definitions, names in descriptions of functions and tools of these systems also undergo changes, some formulations earlier given in separate manuals, reviews of tool capabilities are being not only refined, but also

changed. This is normal for new scientific directions and technologies. The reader should not be surprised if in other sources they meet different formulations, terms.

A detailed description of the functionality of CAS and CMS can be found in [6], the current state, an overview of the systems on the market can be clarified in [7], a brief outline (basic features) for beginners can be found in [8].

IV. COMPUTER ALGEBRA TOOLS

Regarding the classification of CAS. A fairly complete list with the functionality of symbolic computation systems and the platforms they operate on can be found in [7]. Classification attributes of CAS are: functional purpose, type of architecture, means of implementation, fields of application, integral quality assessments.

A. Basic functionality of CAS

CAS allow computer-assisted implementation of analytical and numerical methods for solving problems, presenting the results in mathematical notation, providing graphic visualization, design of the results, and preparation for publication. Using CAS and a computer, it is possible to perform in analytical form the following computations:

- simplifying expressions or reducing to the standard form;
- substituting symbolic and numeric values into expressions;
- extraction of common factors and divisors;
- exponentiation of products and powers, factorization;
- decomposition into simple fractions;
- finding limits of functions and sequences;
- operations with series;
- differentiation in full and partial derivatives;
- finding undetermined and definite integrals;
- continuity analysis of functions;
- finding extremes of functions and their asymptotes;
- operations with vectors;
- matrix operations;
- finding solutions to linear and nonlinear equations;
- symbolic solution of optimization problems;
- algebraic solution of differential equations;
- integral transformations;
- direct and inverse fast Fourier transform;
- interpolation, extrapolation and approximation;
- statistical computations;
- machine theorem proving.

If a problem has an accurate analytical solution, the user of CAS can get this solution in explicit form (of course, we are talking about problems for which the algorithm of solution construction is known).

Also, most of the CAS provide:

- numerical operations of arbitrary precision;

- integer arithmetic for large numbers;
- calculation of fundamental constants with arbitrary precision;
- support for number theory functions;
- editing mathematical expressions in two-dimensional form;
- graphing of analytically defined functions;
- function graphing using table values;
- plotting function graphs in two or three dimensions;
- animation of the plots of various types;
- use of special-purpose extension packages;
- programming in the built-in language;
- automatic formal verification;
- program synthesis.

CAS in the modern implementation are not only applicable for the study of various mathematical and scientific and technical problems using built-in and additional functions, but also contain all the components of programming languages – de facto are problem-oriented high-level programming languages.

Mathematica and Maple are the leaders of CAS – these are powerful systems with their own kernels, equipped with an advanced user interface and with a variety of graphical and editorial capabilities. CAS are also widely used today: Derive, Maxima, Axiom, Reduce, MuPAD. Computer mathematics systems MATLAB, MathCad occupy a special place.

B. Noncommercial general-purpose CAS

A distinctive feature of the current state of IT is that commercial software products in many cases can be fully or partially replaced by non-commercial software, analogues with open source – free software. This includes software products which, with or without modification, have no restrictions on use, copying or transferring to other users, whether for a fee or for free. The following is a reference to software released under the GPL.

Maxima. Maxima is a free, full-featured computer algebra system, a descendant of Macsyma, which was developed as part of the Artificial Intelligence Project at the Massachusetts Institute of Technology from 1968 to 1982 (development stages and leaders of the development teams of the main sections listed in [9]). Experts note that Maxima, unlike Mathematica and Maple, is mainly oriented toward applied mathematical calculations. In this connection, the system lacks or reduces sections related to theoretical methods, such as number theory, group theory, algebraic fields, and mathematical logic. At the same time, numbers in mathematical expressions in the system are assumed to be real by default. This allows to get analytical solutions for many computations encountered in applied problems (such as algebraic transformations and simplifications, integration, solution of differential equations), for which solutions do not exist in the complex domain. Maxima itself is a console

program; it “draws” all mathematical formulas with regular text characters. This has some advantages. For example, you can use Maxima itself as a kernel to build various graphical special interfaces on top of it. There are several examples to date. We can recommend the following textbook as a basic introduction to Maxima CAS, available in electronic form [10].

Axiom. Axiom is a free computer algebra system [11]. It consists of an interpreter environment, a compiler, and a library describing a strict, mathematically correct type hierarchy. Its development was begun in 1971 by a group of IBM researchers, led by Richard Dimick Jenks. The original name of the system was Scratchpad. Originally the project was seen as a research platform for developing new ideas in computational mathematics. It was sold to the Numerical Algorithms Group (NAG) in the 1990s, named Axiom, and became a commercial product, but was not a commercial success and was withdrawn from the market in October 2001. NAG made Axiom free software and opened the source code under a modified BSD license. Development of the system continues, with releases of new versions [11]. In 2007, Axiom had two open-source forks: OpenAxiom and FriCAS.

OpenAxiom [12] released version 1.4.2 in April 2013. The main changes implemented in this version relate to the work of the compiler. The aforementioned system for preparing and editing documents with GNU TeXmacs mathematical notation can be used as an OpenAxiom interface.

Another branch of Axiom that is being actively developed is FriCAS [13], version 1.3.8 (version 22/06/2022) is currently in use. FriCAS favourably differs from other general-purpose CAS by the developed type hierarchy corresponding to real mathematical structures. Axiom and the named branches are inferior to Maxima at this stage in the pace of development. It is better for beginners to focus on Maxima.

The above information about Maxima, Axiom CAS is specifically given because their codes are open, can be used in the Ecosystem OSTIS.

C. Proprietary CMS, CAS

MATLAB programming system (short for Matrix Laboratory) was developed by The MathWorks, Inc. It is one of the oldest, thoroughly developed and time-tested systems for automating mathematical calculations, built on an extended representation and application of matrix operations. Nowadays the system has gone far beyond specialized matrix and has become one of the most powerful universal integrated CMSs. MATLAB includes tools for developing complex programs with an advanced graphical interface, is an effective environment for conducting research, creating models, solving natural science and engineering problems [14]. The system has become de-facto one of the world standards in the field of modern mathematical and scientific and technical software. First

of all, CMS is focused on numerical calculations, with matrix algebra being particularly prominent. The effectiveness of the system is primarily due to its orientation to work with multidimensional arrays, large and sparse matrices with software emulation of parallel calculations and simplified tools for setting cycles. Recent versions of the system support 64-bit microprocessors and multi-core microprocessors such as Intel Core 2 Duo and Quad. System functionality is provided by a rich command library and its own programming language. MATLAB is the largest of all PC-oriented systems due to large number of extension packages it comes with. Its file size exceeds 3 GB. MATLAB works on most modern operating systems, including Linux, macOS, Solaris (support for Solaris is discontinued since R2010b) and Windows. There are many publications describing the system and its components – in Russian [15] can be noted.

Versions history of MATLAB can be traced in [16]. Focusing only on notable items in terms of AI, machine learning, and data mining for versions after 2012 (code R2012* means the 2012 version):

- MATLAB 8.2 R2013b – added table data type, Java runtime updated to version 7;
- MATLAB 8.4 R2014b – added improved user toolbar, new functions and packages, such as py (to use Python), web page counter, histograms, TCP client, and others;
- MATLAB 8.6 R2015b – new runtime mechanism (LXE) and new classes, such as graphs and or-graphs, have been added to handle graphs;
- MATLAB 9.1 R2016b – official MATLAB engine for JAVA, new encoding and decoding functions for JSON, new "string" data type added; algorithms for handling non-memory data, including algorithms for dimensionality reduction, descriptive statistics, k-means clustering, linear regression, logistic regression and discriminant analysis; Bayesian optimization to automatically adjust machine learning algorithm parameters, component neighborhood analysis (NCA) to select machine learning model functions;
- MATLAB 9.5 R2018b – implemented graph axis interaction, which provides efficient data analysis with panning, zooming; added functions: removing outliers in an array, table or schedule; setting a local environment about each element in the input data;
- MATLAB 9.6 R2019a – Added Functions to specify the location of a missing value, detect outliers using percentiles; Implemented improvements for artificial intelligence and analytics;
- MATLAB 9.7 R2019b – includes updates on artificial intelligence (new features allow users to train advanced network architectures using custom learning cycles, automatic differentiation, common weights, and custom loss functions; users can create generative adversarial GANs, Siamese networks,

variational autoencoders, and attention networks; Deep Learning Toolbox can also now export to ONNX format networks that combine CNN and LSTM layers, and networks that include 3D CNN layers);

- MATLAB 9.11 R2021b – added: a set of tools for statistics and machine learning (signal and image analysis, preprocessing and parameter extraction using wavelet methods and interactive applications for artificial intelligence models), k-means clustering in real problems;
- MATLAB 9.13 R2022b includes updates on artificial intelligence, a set of system identification tools – create nonlinear state-space models based on deep learning using neural ordinary differential equations (ODEs); machine learning and deep learning techniques can also represent nonlinear dynamics in nonlinear ARX and Hammerstein-Wiener models.

MATLAB is a commercial system; there are non-commercial versions of its type that are compatible in basic language constructs, but not compatible in library functions. For example, Scilab, Maxima, Euler Math Toolbox and Octave.

MATLAB includes a command interpreter, graphical shell, editor-debugger, profiler, compiler, symbolic kernel Maple for analytical calculations, mathematical libraries and Toolboxes libraries, designed to work with special classes of tasks.

MATLAB language. MATLAB system is both an operating environment and a programming language. MATLAB language is a high-level interpreted programming language. Programs written in MATLAB are of two types: functions and scripts. MATLAB programs, both console and GUI, can be compiled using MATLAB Compiler component into independent executable applications or dynamic libraries. MATLAB Builder programs extend the capabilities of MATLAB Compiler and allow you to create independent Java, .NET or Excel components.

Basic extension packages. A feature of MATLAB CMS is the ability to create special toolboxes. MathWorks supplies more than 80 toolboxes that are used in many areas. In recent releases, the company classifies them into three families – MATLAB, SIMULINK and Polyspace [14], as well as partner products.

Maple. Mathematica and Maple are the leaders of CAS, and they are often compared. This seems counterproductive. Each of the systems named has its own characteristics, and they have their own strengths and weaknesses; constantly competing with each other, they are evolving and improving. Most CAS users have experienced several other systems before choosing their primary system. The exchange of opinions, the analysis of publications, and presentations at conferences and seminars allow us to state that each system has its own adherents, and it is useless to convince specialists who have been using CAS for quite a long time that a system

other than the one they prefer is somehow better than the others. In most cases, the main factor in using a particular CAS is the user's habit. However, many note that having mastered any of the systems, it is easy to work with others.

In terms of completeness and interface solutions, Mathematica and Maple tools for symbolic and numerical computation are perfect, not the lack of any functions or tools, but the skills of the users. It is impossible to give a complete overview of the capabilities of Maple, as well as Mathematica. It is unlikely that any of the authors of even specialized publications with a book focus on a particular class of problems can present all of the tools of the named CAS from the spectrum they cover. This material can only be regarded as an introduction to the capabilities of the system, mentioning classes of problems of interest to undergraduates, graduate students, postgraduates, researchers, and programmers. Again, the functionality of Mathematica and Maple in almost everything related to mathematics, applied mathematics, and computer science is not only sufficient, but also redundant. Since the main list of features of CAS has already been given above, and in Maple they are implemented, here we will note what in a number of sources is either omitted or called by other terms.

There are many books devoted to the Maple system around the world, a list of which can be found on the developer's section [17]. Publications in Russian can be tracked by [18], [19]. Despite its fundamental nature and focus on the most serious mathematical calculations, Maple-class systems are needed by a fairly wide category of users: university students and teachers, engineers, graduate students, researchers, and even students of mathematics classes of general education and special schools. Maple is a typical integrated software system [19]–[21].

The main Maple document is the Worksheet, which is similar to editing in a text editor. Text can be formatted at the paragraph level, with different styles, or as symbols. The content of the document can be structured into sections, sub-sections, etc., all the way down to cells.

Like most CAS, the Maple interface combines text and command processor functions. Since version 8, Maplets have been added to the system to support visually-oriented dialog.

Maple (like Mathematica) integrates three languages: communication language, implementation, and programming.

The Maple kernel and all of its components improve from version to version. Many of the functions built into the system, like the kernel functions, can be used without any declaration, others need to be declared. There are several problem-oriented packages (packages) that cover many branches of classical and modern mathematics. The total number of functions in Maple, including those built

into the kernel and placed in packages, exceeds 5500. The kernel (not in full composition) uses MATLAB and MathCad (starting from version 14, symbolic kernel MuPAD is used).

The main stages of development, additions in the Maple versions can be traced in [20], but it must be stated that machine learning, artificial intelligence, and data mining are not yet priorities for the system.

Wolfram Mathematica. Mathematica – is a computer algebra system developed by Wolfram Research company. It is one of the most powerful and widely used integrated multimedia-technology software package [6], [22]–[25]. Mathematica is recognized as a fundamental advancement in computer-aided design. It is one of the world’s largest software packages by volume of modules and contains many new algorithms, as well as many unique developments. Mathematica lets users use virtually every analytic and numeric option, and it supports databases, graphics, and sound. Mathematica lets you work, analyze, manipulate, and graph almost any function of pure and applied mathematics. The system provides calculations with any specified accuracy; construction of two- and three-dimensional graphs, their animation, drawing geometric figures; importing, processing, exporting images and sound.

Mathematica has evolved from a program used primarily for mathematical and technical calculations to a tool widely used in various other areas [22], [23]. It is recognized among specialists as a development platform that fully integrates computation into the workflow from start to end, seamlessly guiding the user from initial ideas to deployed custom and industrial solutions.

Mathematica has a built-in Wolfram Language, including tools for creating programs and user interfaces, connecting external dlls, and parallel computing. The system’s programming language is a typical interpreter; it’s not designed to create executable files, but it incorporates the best of such programming languages as BASIC, Fortran, Pascal, and C. The Mathematica programming language supports all known paradigms: functional, structural, object-oriented, mathematical, logical, recursive, and more. It also includes visual-oriented programming tools based on the use of mathematical symbol templates, such as integral, summation, product, etc.; this language exceeds the usual general-purpose programming languages in its ability to perform mathematical and scientific computations.

Like all computer algebra, Mathematica is a type of software tool designed to manipulate mathematical formulas. Its main purpose is to automate the often tedious and in many cases difficult algebraic transformations. User works in the system with notebooks - NB documents, each document contains at least one section (cell). An explanation of the preference adopted here is a comparison with MS Excel, where the term cell is

steadily and universally used. Those who have experience with Excel and Mathematica understand the difference and that in MS Excel it is cells, and in Mathematica notebooks it is more general objects.

NB documents can be opened, viewed, edited, saved, executed in their entirety, or individual cells. Notebook’s interface contains many palettes (menus) and graphical tools for creating, editing, viewing documents, sending and receiving data to and from the core. Notebook includes one or several cells that can be grouped together as needed. Each cell contains at least one line of text or formulas, a digital audio or video object. Notebooks can be edited as text in any editor or in the Mathematica interface. The kernel performs the computations and can be run on the same computer as the interface, or on another computer connected through a network. Typically, the kernel is started when the computation begins. Cells in Mathematica can be roughly divided into input and result (output) cells. In the input cells, the user enters or places commands, comments, multimedia objects, and they can be executable or otherwise; the executable cells are processed – the system returns results and displays them in output cells.

All versions of Mathematica include a powerful reference database, and the built-in Help, Documentation Center is an example of an NB document in itself. Without interrupting work on modules, you can clarify any function, option, directive, or service word; explore the capabilities of “live” examples to get and document results; and embed examples or code snippets from examples in your own code.

From the chronology of Mathematica versions. The first release of Mathematica was in June 1988, the basic concept being to create once and for all one system for different computations in a consistent and unified way. The basis for this was the creation of a new symbolic computer language for controlling, with a minimum number of inputs, the large number of objects involved in technical computation. Since its inception, all Wolfram Research Inc. developments have been regularly ranked first among IT achievements, highlighted by the media.

Release dates, additions, and updates to Mathematica are fully reflected in a number of publications and websites, e.g. [22], [23]. Experts note that the list of updates to Wolfram Mathematica reflects many completely new advances that have found application, development in other systems, and information technology. Wolfram Research Inc. developments are mostly characterized by interface continuity and the ability to use source code from previous versions.

About Mathematica features. A complete list of capabilities would require several times as much space as this presentation allows. For example, manual [24] has over 600 pages of content, but in fact it only outlines

the basic functions of the CAS. Originally published in 1988 and updated in Mathematica 5, “The Mathematica Book by Stephen Wolfram”, Fifth Edition, 2003, is 1,488 pages long. A list of books by S. Wolfram’s books can be viewed at [25].

Help system in Mathematica. The Documentation Center, Function Navigator, and Virtual Book are part of the Help system. These modules provide all necessary information to guide users through the language and functionality of Mathematica. Built-in documentation contains more than 150,000 representative and illustrative examples of Wolfram code. All documents are fully interactive; they are Mathematica notebooks in which the user can try out their own code and modify the examples directly in the Help system.

The theses outlined above are important from the position of developers of computer systems for artificial intelligence to understand the current state in closely related fields, in particular because computer algebra systems, which implement intelligent computations with the help of a computer, are also one of the (and quite successfully developed) areas to adopt artificial intelligence.

V. EXAMPLES OF INTEGRATING WM TOOLS INTO OSTIS APPLICATIONS

A. Wolfram Mathematica. Current state

Building on over thirty years of research, development, and use around the world, Mathematica and Wolfram are geared for the long term and especially successful in computational mathematics. The roughly 6,000 functions (symbols) built into Wolfram allow the user to represent and manipulate a huge variety of computational objects – from special functions to graphics and geometric regions.

In addition, the Wolfram knowledge base [26] and its associated entity structure [27] allows to explain, interpret, and formalize hundreds of specific “things” (facts, situations, objects). For example: people, cities, food, structures, planets, etc. appear as objects that can be manipulated and counted.

B. Wolfram knowledge base. Coverage areas

The growing Wolfram Data Repository (WDR), based on Wolfram Alpha and the Wolfram Language, is now the world’s largest repository of computable knowledge. Covering thousands of fields, the WDR contains carefully selected expert knowledge obtained directly from primary sources. It includes not only trillions of data elements, but also a huge number of algorithms that encapsulate methods and models from virtually every domain. The Wolfram Knowledge Base is based on Wolfram’s three decades of accumulated computable knowledge. All data in the Wolfram KB can be used immediately for Wolfram computations. Every millisecond of every day,

the Wolfram Knowledge Base is updated with the latest data.

Major coverage areas of WDR [26] are shown in Fig. 1.



Figure 1. Coverage areas of WDR.

In [28] typical options for working with WDR in Education are outlined, as well as examples of interaction with Wikipedia.

With extensive statistics on hundreds of thousands of educational institutions around the world, Wolfram|Alpha can calculate answers to complex questions about education. For example, you can query what academic degrees students receive at prestigious universities, average enrollment figures by year for selected majors. In the examples [28] illustrations of the response to the query about the number of students in the Republic of Belarus, quantitative indicators for the leading universities of BSU and BSUIR are given. Ways to present knowledge and access to it are described. It is noted that access to the Wolfram knowledge base is deeply integrated in Wolfram Language (WL). Free-form linguistics makes it easy to identify many millions of entities and many thousands of properties, and automatically generates accurate Wolfram Language representations suitable for extensive further computation. WL also supports custom entity stores that allow you to perform the same computations as the built-in knowledge base and can be linked to external relational databases.

People interact with each other through speech and text, and this is called natural language. Computers understand people’s natural language using Natural Language Processing (NLP). NLP is the process of manipulating human speech and text with artificial intelligence so that computers can understand them. In [28] the basic NLP tools implemented in Wolfram Mathematica are noted. In particular: Speech recognition; Voice assistants and chatbots. Auto-substitution and auto-prediction. Email filtering. Sentiment analysis. Divertissements for the target audience. Translation. Social media analytics. Recruitment (staffing). Text summary (abstracting). Several representative examples with explanations of the functions of WL groups Structural Text Manipulation, Text Analysis, Natural Language Processing are mentioned.

Also in [28] examples of knowledge extraction, entities from Wikipedia articles are discussed. Wikipedia data uses the MediaWiki API to extract article and category content and metadata from Wikipedia. An article can be specified as a string or Wolfram Language object. Retrieving articles associated with language entities is provided by the WM TextSentences feature, in particular, it is possible to work with Wikipedia resources. Presented are specific results of the TextSentences function, with parameters WikipediaData, Entity, "Person", "AlexeiLeonov".

These examples of working with knowledge bases using WM tools, since the system kernel functions can be used in programs developed on other platforms, can be interpreted as proposals for the innovative improvement of existing tools, components of any intelligent computer systems, and of course the Ecosystem OSTIS.

VI. EXAMPLE OF INTEGRATING WOLFRAM MATHEMATICA WITH EDUCATIONAL OSTIS-SYSTEM PROTOTYPE FOR DISCIPLINE "COMPUTER SYSTEMS AND NETWORKS"

Here is an illustration of the combined use of WM and OSTIS-prototype for discipline "Computer Systems and Networks" ostis-system for working with computer network topologies. The results below show the possibilities of using the visualization performed in WM in the ostis-system. Moreover, implementations are available using an appropriate programming interface (it is possible to execute WL code hosted in the Wolfram cloud within a user program, such as Python or C++ [29]) or import, export tools. According to Mathematica *\$ImportFormats* and *\$ExportFormats* functions, it supports more than 100 formats, the list of formats is as follows:

3DS, ACO, Affymetrix, AgilentMicroarray, AIFF, ApacheLog, ArcGRID, AU, AVI, Base64, BDF, Binary, Bit, BMP, BSON, Byte, BYU, BZIP2, CDED, CDF, Character16, Character8, CIF, Complex128, Complex256, Complex64, CSV, CUR, DAE, DBF, DICOM, DIF, DIMACS, Directory, DOT, DXF, EDF, EML, EPS, ExpressionJSON, ExpressionML, FASTA, FASTQ, FCS, FITS, FLAC, GenBank, GeoJSON, GeoTIFF, GIF, GPX, Graph6, Graphlet, GraphML, GRIB, GTOPO30, GXL, GZIP, HarwellBoeing, HDF, HDF5, HIN, HTML, HTTPRequest, HTTPResponse, ICC, ICNS, ICO, ICS, Ini, Integer128, Integer16, Integer24, Integer32, Integer64, Integer8, JavaProperties, JavaScriptExpression, JCAMP-DX, JPEG, JPEG2000, JSON, JVB, KML, LaTeX, LEDA, List, LWO, M4A, MAT, MathML, MBOX, MCTT, MDB, MESH, MGF, MIDI, MMCIF, MO, MOL, MOL2, MP3, MPS, MTP, MTX, MX, MXNet, NASACDF, NB, NDK, NetCDF, NEXUS, NOFF, OBJ, ODS, OFF, OGG, OpenEXR, Package, Pajek, PBM, PCAP, PCX, PDB, PDF, PGM, PHPIni, PLY, PNG, PNM, PPM, PXR, PythonExpression, QuickTime, Raw,

RawBitmap, RawJSON, Real128, Real32, Real64, RIB, RLE, RSS, RTF, SCT, SDF, SDTS, SDTSDEM, SFF, SHP, SMA, SME, SMILES, SND, SP3, Sparse6, STL, String, SurferGrid, SXC, Table, TAR, TerminatedString, TeX, Text, TGA, TGF, TIFF, TIGER, TLE, TSV, UBJSON, UnsignedInteger128, UnsignedInteger16, UnsignedInteger24, UnsignedInteger32, UnsignedInteger64, UnsignedInteger8, USGSDEM, UUE, VCF, VCS, VTK, WARC, WAV, Wave64, WDX, WebP, WNet, WMLF, WXF, XBM, XHTML, XHTMLMathML, XLS, XLSX, XML, XPORT, XYZ, ZIP.

In the example below, the initial data (a particular graph of the topology of a computer network) is imported from a teaching ostis-system for the "Computer Systems and Networks" discipline, visualized by WM graphics, then solved the typical problem and the preferred final results are exported back to the teaching ostis-system for the discipline. Initial data, the specific graph of the network topology used next is shown in the Fig. 2.

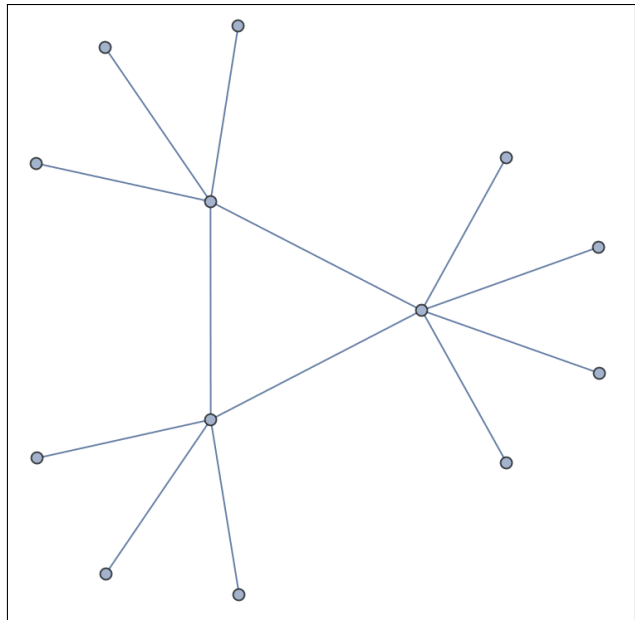


Figure 2. Network topology graph, nodes and connections (in the ostis-system).

The following illustrations are generated in WM.

For the imported graph in WM, you can get general information such as: number of vertices (network nodes), list of edges (connections between nodes), and visualize it. Fig. 3 shows the output of the vertex list (VertexList), the number of edges (EdgeCount), and the edges list (EdgeList).

The three output layouts are shown below for an example visualization. Fig. 4 shows vertices and edges with their weights. This form of representation is preferable for visualization of logical network topology, in which the directions of data flows are indicated. Edges with

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}
13
{1 ↔ 2, 2 ↔ 3, 3 ↔ 1, 1 ↔ 4, 1 ↔ 5, 1 ↔ 6,
 2 ↔ 7, 2 ↔ 8, 2 ↔ 9, 3 ↔ 10, 3 ↔ 11, 3 ↔ 12, 3 ↔ 13}
```

Figure 3. Network topology graph, general information (output in Wolfram Mathematica).

no arrows are bidirectional, which is analogous to two opposing edges with the same weight.

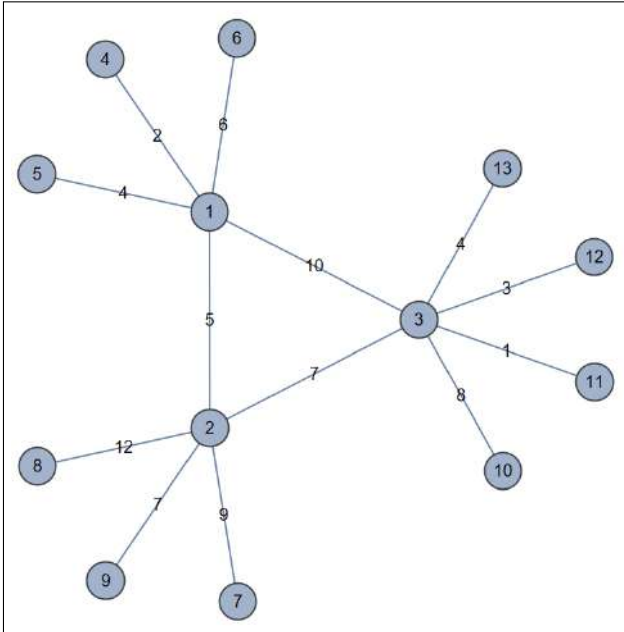


Figure 4. Network topology graph, connections with weights (output in Wolfram Mathematica).

The example output with specifying the LayeredEmbedding graph format, applying the vertex design options is shown in Fig. 5.

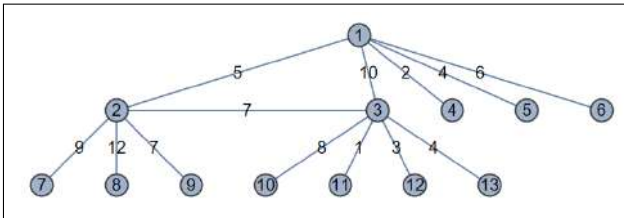


Figure 5. Network topology graph with LayeredEmbedding layout (output in Wolfram Mathematica).

The example output with specifying the LayeredDigraphEmbedding graph format, applying the vertex design options is shown in Fig. 6.

Let us consider one of the dynamic routing protocols, OSPF (Open Shortest Path First), as the example of the described network topology. This protocol is interesting

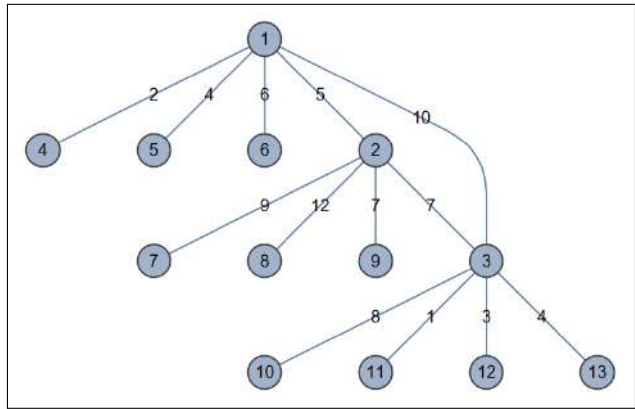


Figure 6. Network topology graph with LayeredDigraphEmbedding layout (output in Wolfram Mathematica).

because it uses the Dijkstra algorithm to find the shortest path [30].

The principle of the protocol is as follows:

- 1) Once routers are enabled, the protocol searches for directly connected neighbors and establishes communication with them.
- 2) Routers exchange information with each other about the networks connected and available to them – build a network map (network topology). This map is the same on all routers.
- 3) Based on this information, the SPF (Shortest Path First) algorithm is run, which calculates the best route to each network. This process is similar to building a tree, with the root being the router itself, and the branches being paths to available networks.

In the OSPF protocol, convergence occurs rather quickly due to the use of the Dijkstra algorithm [31].

Fig. 7 illustrates the example of solving the problem of finding the optimal route between two nodes in a network. The following Wolfram Mathematica functions were used: GraphDistance, NeighborhoodGraph, Sow, DirectedEdge, Placed, Union, and Flatten.

Consider the following situation: in this initial network topology the communications are upgraded. Between nodes 2 and 3 a fiber optic is laid, which provides more bandwidth and data transfer speed. As a result of the replacement, the “cost” of transmitting data over this link decreases from 8 conventional units to 3. How will the changes affect the solution to the problem at hand: the optimal route between nodes 4 and 12 of the network?

When routers are included in the network, they check the speed capabilities of all available data channels. In our situation, routers 2 and 3 will determine that the communication between them has improved qualitatively. In accordance with the OSPF protocol, these devices on request will transmit this information about the topology change to the neighboring router 1. Now, in the future, when deciding on a path to transmit data from node 4

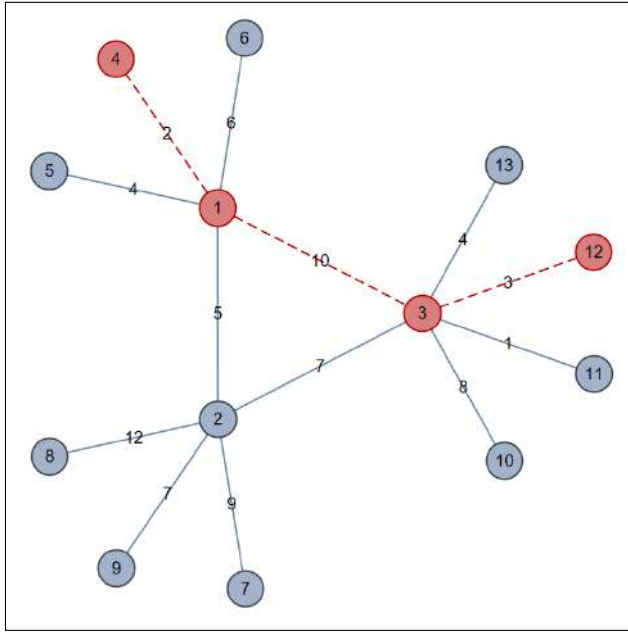


Figure 7. Solving the problem of finding the optimal route between two network nodes (output in Wolfram Mathematica).

to node 12, router 1 will choose a new shortcut. The solution to find the optimal route between two nodes in the changed network configuration is shown in Fig. 8.

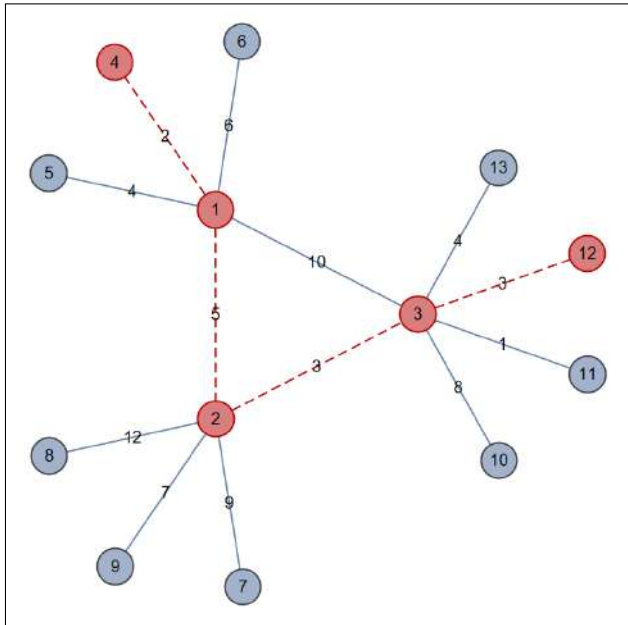


Figure 8. Solving the problem of finding the optimal route between two network nodes in updated topology (output in Wolfram Mathematica).

The results obtained and considered include labor-intensive for implementation in programming languages graphics problems, as well as mathematically and algorithmically complex problems of the subject area. Presented options of visualization, finding a solution require

only a careful study of examples of the help system Wolfram Mathematica, certain programming skills, i.e. are available to most software engineers. Transferring results to other software applications isn't difficult either, because WM provides export options in any standard format.

VII. CONCLUSION

Programming language, high level of documentation, features of step-by-step problem solving and graphical visualization of initial data and computational results favorably differentiate Mathematica from other CAS. These features combined with an intelligent predictive interface help subsystem, provide wide range of opportunities for integrating Wolfram Language functions with components of Ecosystem OSTIS.

The presented example of addition of intellectual educational resource built within the ostis-system for discipline "Computer Systems and Networks" illustrates connection of difficult for programming functions of visualization and solution of complex mathematical problem of finding the optimal path in the info-communication network with a complex topology.

REFERENCES

- [1] V. Golenkov, N. Guliakina, and D. Shunkevich, Otkrytaja tehnologija ontologicheskogo proektirovaniya, proizvodstva i jekspluatatsii semanticheski sovmestimyh gibridnyh intellektual'nykh komp'yuternyh sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems], V. Golenkov, Ed. Minsk: Bestprint [Bestprint], 2021, P. 690 (In Russ.)
- [2] GeoS — Geometry Problem Solver. Available at: <https://geometry.allenai.org> (accessed 2023, Feb).
- [3] Handwriting Recognition. Available at: <https://mathpix.com/handwriting-recognition> (accessed 2023, Feb).
- [4] Create Graph online and find shortest path or use other algorithm. Available at: <https://graphonline.ru/en> (accessed 2023, Feb).
- [5] M. Kovalev, A. Kroshchanka, V. Golovko, Convergence and integration of artificial neural networks with knowledge bases in next-generation intelligent computer systems. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, 2022, pp. 173-186.
- [6] V. D'yakonov, Entsiklopediya komp'yuternoj algebrы [Encyclopedia of Computer Algebra], Moscow, DMK Press, 2009, P. 1264 (In Russ.)
- [7] List of computer algebra systems. Available at: http://en.wikipedia.org/wiki/List_of_computer_algebra_systems (accessed 2023, Mar).
- [8] V. Taranchuk, Osnovnye funktsii sistem komp'yuternoj algebrы [Basic Functions of Computer Algebra Systems] : posobie dlya studentov fak. prikladnoi matematiki i informatiki, Minsk, BSU, 2013, P. 59. (In Russ.)
- [9] Maxima, Sistema komp'yuternoj algebrы [Maxima, Computer Algebra System]. Available at: <http://maxima.sourceforge.net/ru> (accessed 2023, Feb) (In Russ.)
- [10] N. Stakhin, Osnovy raboty s sistemoi analiticheskikh (simvol'nykh) vychislenii Maxima [Fundamentals of working with the system of analytical (symbolic) calculations Maxima]. (PO dlya resheniya zadach analiticheskikh (simvol'nykh) vychislenii), Moscow, Federal'noe agentstvo po obrazovaniyu, 2008. 86p. (In Russ.)
- [11] AXIOM. The Scientific Computation System. Available at: <http://axiom-developer.org> (accessed 2023, Feb).

- [12] OpenAxiom: The Open Scientific Computation Platform. Available at: <http://open-axiom.sourceforge.net> (accessed 2023, Feb).
- [13] FriCAS — an advanced computer algebra system. Available at: <http://fricas.sourceforge.net> (accessed 2023, Feb).
- [14] MathWorks. MATLAB. The Language of Technical Computing. Available at: <http://www.mathworks.com/products/matlab/> (accessed 2023, Mar).
- [15] V. D'yakonov, MATLAB. Polnyi samouchitel' [MATLAB. Complete tutorial], Moscow, DMK Press, 2012, P. 768 (In Russ.)
- [16] MATLAB Versiya — Osobennosti i preimushchestva versii MATLAB [MATLAB Version — Features and Benefits of MATLAB Version]. Available at: <https://ru.education-wiki.com/2303364-MATLAB-version> (accessed 2023, Mar). (In Russ.)
- [17] Maplesoft. Books. Available at: <http://www.maplesoft.com/books/index.aspx> (accessed 2023, Feb).
- [18] V. Aladjev, V. Vaganov Modul'noe programmirovaniye: Maple vs Mathematica, and vice versa [Modular programming: Maple vs Mathematica, and vice versa], CA: Palo Alto, Fultus Corp., 2011, P. 417 (In Russ.)
- [19] V. D'yakonov, Maple 10/11/12/13/14 v matematicheskikh raschetakh [Maple 10/11/12/13/14 in mathematical calculations], Moscow, DMK Press, 2011. 800 p. (In Russ.)
- [20] Maple Product History. Available at: <http://www.maplesoft.com/products/maple/history> (accessed 2023, Mar).
- [21] V. Alad'ev, Sistemy komp'yuternoï algebrы: Maple: Iskusstvo programmirovaniya [Computer Algebra Systems: Maple: The Art of Programming], Moscow, Laboratoriya Bazovykh Znaniy, 2006, P. 792 (In Russ.)
- [22] Mathematica. Istoriya versii. Available at: http://ru.wikibooks.org/wiki/Mathematica/История_версий (accessed 2023, Mar). (In Russ.)
- [23] Mathematica Quick Revision History. Available at: <http://www.wolfram.com/mathematica/quick-revision-history.html> (accessed 2023, Mar).
- [24] V. D'yakonov, Mathematica 5/6/7. Polnoe rukovodstvo [Mathematics 5/6/7. Complete Guide], Moscow, DMK Press, 2009, P. 624 (In Russ.)
- [25] Stephen Wolfram. Books. Available at: <http://www.stephenwolfram.com/publications/books> (accessed 2023, Mar).
- [26] Wolfram Mathematica: Modern Technical Computing. Available at: <https://www.wolfram.com/mathematica> (accessed 2023, Feb).
- [27] V. Taranchuk, Intel'ktual'nye vychisleniya, analiz, vizualizatsiya bol'shikh dannykh [Intelligent Computing, Analysis, Visualization Of Big Data]. *BIG DATA i analiz vysokogo urovnya [BIG DATA and Advanced Analytics]*, 2019, no. 1, pp. 337–346. (In Russ.)
- [28] V. Taranchuk, Integration of computer algebra tools into OSTIS applications. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, 2022, pp. 369-374.
- [29] APIFunction — Wolfram Language Documentation. Available at: <https://reference.wolfram.com/language/ref/APIFunction.html> (accessed 2023, Mar).
- [30] OSPF Version 2. Available at: <https://www.ietf.org/rfc/rfc2328.txt> (accessed 2023, Apr).
- [31] T. Roughgarden, Algorithms Illuminated: Graph algorithms and data structures. Part 2, New York, Soundlikeyourself Publishing LLC, 2018, P. 222

Примеры интеграции инструментов Wolfram Mathematica в приложения OSTIS

Таранчук В. Б., Савёнок В. А.

В рамках концепции конвергенции и унификации интеллектуальных компьютерных систем нового поколения, обсуждаются технические решения, приведены примеры разработки и модернизации, интеграции средств Экосистемы OSTIS с системой компьютерной алгебры (СКА) Wolfram Mathematica (WM).

На примере интеграции в специализированном комплексе интеллектуального образовательного ресурса по дисциплине “Компьютерные системы и сети” рассмотрены возможности использования инструментов WM в ostis-системе при решении задач, связанных, в частности, с топологией инфокоммуникационных сетей.

Показано применение инструментов WM для визуализации топологии сети, а также эмуляции поиска оптимального маршрута передачи данных.

Received 10.03.2023

Intelligent Health Monitoring Systems

Vladimir Rostovtsev
RSPC of Medical technologies
Ministry of Health of the Republic of Belarus
Minsk, Belarus
Email: vnrost@kmsd.su

Abstract—The paper considers the problems of the current state of complex monitoring of human health, as well as relevant diagnostic technologies. An approach to the intellectualization of the process of regular health monitoring based on FSD-diagnostics systems is proposed. The possibility of transition to a new technological level of public health control through the application of the proposed approaches is shown.

Keywords—intelligent health monitoring, OSTIS Technology, ontology, FSD-diagnostics

I. INTRODUCTION

Taking (in a zero approximation and formally) the understanding of health as the absence of a diagnosis of a disease, four gradations of health status should be distinguished:

- harmonious health,
- risk of disease,
- latent (hidden) stage of the disease,
- early stage of the disease (there are already symptoms, but the diagnosis has not yet been made).

Unfortunately, the diagnosis is often made at a late stage of the disease. This means that in order to help the natural intelligence of a doctor, it is necessary to create artificial intelligence systems for medical purposes.

In artificial and natural systems, intelligence is a tool designed to solve problems of recognition and generation of images and problems of recognition and generation of meanings.

Artificial intelligence systems use two approaches, including logical-semantic (based on ontologies) and logical-statistical (based on logical-analytical or neural network algorithms).

In medicine, due to the key importance of the diagnosis of risks and diseases, the most relevant tasks are image recognition, for which both approaches are used - logical-semantic and logical-statistical, including neural network, as well as hybrid versions of the architecture of diagnostic systems.

In addition to the development of intelligent diagnostic systems, the development of two more classes of intelligent medical systems is relevant. Firstly, it is the development of logical and semantic systems for consulting a doctor in the processes of making medical decisions concerning individual health-preventive or

treatment-rehabilitation programs, including prescribing medicines. Secondly, it is the development of intelligent user interfaces in order to integrate existing medical information systems into a single distributed interoperable system.

Monitoring systems are diagnostic systems and are highly reliable due to the availability of dynamic data.

Health monitoring is designed to identify the risks of developing diseases and latent (hidden) stages of their development, as well as for timely diagnosis of manifest diseases.

The main task of intelligent health monitoring systems is to provide diagnostics of current risks and latent stages of diseases.

The organization of population monitoring based on laboratory diagnostic methods, even for just a few diseases, is economically unbearable. Existing wearable gadgets do not pretend to diagnose diseases, not to mention the diagnosis of nosological risks.

To solve the designated tasks of health monitoring, the most effective diagnostic technology is the technology of functional spectral dynamic diagnostics (FSD-diagnostics) [1]. This is due to the fact that FSD-diagnostics is effective against all common infectious and non-communicable diseases. Today, only FSD-diagnostics claims to provide economically justified and diagnostically effective health monitoring.

It is technologically important that each recording of the body's FSD-signal allows you to assess nosological risks for hundreds of nosological positions simultaneously in the teliagnostic mode in a few minutes.

FSD-monitoring of health will ensure the transition to a new technological level of public health control.

II. FSD-DIAGNOSTICS

The main areas of the body include genetic, metabolic, functional, mental and wave. Each sphere has its own type of processes.

The wave sphere includes all electromagnetic wave processes in the body and its surroundings. The frequency range of these wave processes is from fractions of Hertz to far ultraviolet radiation.

The state of the wave sphere depends on the state of all other spheres of the body and on the field load. Field cargo consists of technogenic fields (electrosmog),

geopathogenic fields and unfavorable space and anthropogenic fields.

The mutual interdependence of the wave sphere with each of the other spheres of the organism makes the wave sphere the main factor, and the material-wave interactions are the main mechanism for integrating the organism into a single whole. Without this mechanism, the integrity of the organism is unattainable as such. The wave sphere "covers" all other spheres, or rather it forms a single organic whole with them.

The division of processes into real and wave processes is conditional. There really are unified and inseparable material-wave processes. Or, in other words, there is a single complex system of material-wave processes in biological organisms. The generalization of these provisions is the fundamental principle of the material-wave dualism of biological processes.

The real-wave interactions are symmetric. Real processes affect field processes, since all real processes generate wave radiation having their own strictly specific spectral-dynamic parameters. In turn, wave processes have an impact on the corresponding spectral-dynamic parameters of real processes. Material-wave interactions are realized mainly through resonance and compensation mechanisms.

In the existing term "biofield", the prefix "bio" has the meaning of a formal indication of the origin of the field, that is, its belonging to a biological object. In terms of content, the biofield is a set of physical electromagnetic fields, which has only one distinctive feature. This feature lies in the high complexity of the organization of the biofield, which corresponds to the complexity of the organization of processes in a biological object. As R. Virchow wrote: "I do not know anything essentially biological except biological organization."

Due to the regular display in the wave sphere of all other spheres, or rather the natural unity of the wave sphere and the other spheres of the body, the wave sphere contains information about all processes in the body without exception. This makes the wave sphere a unique source of diagnostic data and a universal intermediary for the correction of any pathological processes. And not only pathological, but also pathogenetic and risk-genetic.

In the development of pathology, there is a sequence of processes of riskogenesis (gradual formation of high risk), pathogenesis (disease formation) and nosogenesis (disease development).

The most important is the diagnostic identification of the processes of risk genesis, and in other words, the diagnosis of disease risks. Diagnosis of nosological risks is an absolutely necessary basis for effective disease prevention. Just as the diagnosis of diseases is a necessary basis for their effective treatment.

Medical technologies for working with the wave sphere of the body are called wave diagnostics and wave

correction technologies.

Every process in the body is a functional process. Wave processes are as functional as they are real. The material-wave unity of all functional processes in any biosystems determines the understanding of wave diagnostics as functional diagnostics.

The essence of FSD-diagnostics consists in passive (without any impact) recording of a wave signal from the skin surface for a duration of 35 seconds using a metal electrode in the frequency range from 20 Hz to 11 kHz (EMF audio range), spectral analysis of the recorded signal based on the wavelet transform of Dobsha 3 and subsequent recognition of the presence of spectral correspondences with similar spectra of electronic copies reference diagnostic markers.

Spectral correspondence, that is, the similarity of the marker with the corresponding pattern of the patient's spectrum, expressed in %, is the main indicator for the doctor, who, analyzing the indicators for tens and hundreds of markers, makes diagnostic conclusions. The main classes of markers are markers of physiological organ-tissue processes, inflammatory processes, degenerative processes, etiological agents (viral, bacterial, etc.), environmental factors and medicines.

FSD-diagnostics allows you to diagnose common infectious and non-infectious diseases and their risks across all body systems.

The use of FSD-diagnostics for the construction of intelligent health monitoring systems is favored by the following unique combination of technological factors:

- short time, passivity and ease of recording the FSD-signal,
- one-time access to all markers after recording the signal,
- ease of transmitting the audio signal file to the server,
- low cost of the diagnostic cycle in automatic mode, that is, in the presence of an intelligent health monitoring system,
- the ability to use a smartphone to independently record and transmit an audio file of the FSD-signal to a server for automatic diagnostics using an intelligent health monitoring system.

III. INTELLIGENT FSD-HEALTH MONITORING SYSTEMS

The logical-semantic approach based on ontologies involves the use of expert knowledge about markers and the nosological object being diagnosed. This approach does not require big data, but places high demands on experts and, in some cases, involves additional scientific research.

The logical-statistical approach is based on the extraction of knowledge from data. The extracted knowledge concerns, first of all, the diagnostic informativeness of

specific markers. In addition, the extracted knowledge relates to the diagnostic effectiveness and semantics (expert component) of specific sets of informative diagnostic markers. Highly informative sets of markers are used to statistically assess the risks of the development of the diagnosed processes or to assess the chances of the reverse development of these processes. In the monitoring process, logical procedures for making diagnostic decisions based on preset threshold values of risk values are used as the main logical rules for a variety of current risk assessments. Exceptions to the basic rules are used as additional logical rules for making diagnostic decisions. In health monitoring technology, the basic and additional rules may include parameters of the current dynamics of risk values. The implementation of the logical-statistical approach does not require big data and special expert support.

The neural network approach does not involve the extraction of knowledge explicitly and requires big data, which is often inaccessible in medical diagnostic tasks. In addition, neural network systems are semantically opaque and subject to the influence of digital noise.

Marker data obtained with the help of FSD-diagnostics allow the use of all the mentioned approaches.

The implementation of a complete health monitoring system based on FSD-diagnostics includes the following stages:

- 1) recording the body's wave signal and calculating the Dobsny-3 wavelets,
- 2) formation of the patient's wavelet image and logical-statistical recognition of diagnostic markers,
- 3) identification of informative markers and logical-statistical recognition of pathogenesis risks,
- 4) risk profile formation and logical-semantic recognition of system processes,
- 5) formation of a profile of system processes and logical-semantic generation of an individual scenario of preventive rehabilitation.

Technologically implemented the first three stages of a complete system of health FSD-monitoring. Of these, the third stage requires filling with a variety of nosological monitoring positions.

The technology of designing subsystems of the third stage of the implementation of intelligent health FSD-monitoring systems was developed by us on the example of a system for monitoring the risk of developing mastitis in cows.

The cow mastitis risk monitoring system has the necessary health FSD-monitoring infrastructure (sensors, server, terminals, communication channels) and the actual intelligent FSD mastitis risk monitoring system.

This system belongs to logical-statistical intelligent pattern recognition systems based on a highly informative set of diagnostic FSD-markers of mastitis. A set

of 14 wave markers was obtained as a result of multi-stage selection from a variety of informative markers and includes markers of breast inflammation, drug markers and markers of immune system conditions.

The selection of informative mastitis markers was carried out by us on the basis of an FSD-study and a parallel laboratory study of a sample of cows (90 animals). Laboratory studies included the determination of the number of somatic cells and a microbiological ruler. The sample included healthy cows, individuals with subclinical mastitis and individuals with clinical mastitis (according to the results of laboratory diagnostics).

In the system of FSD-monitoring of the risk of developing mastitis in cows, two main diagnostic criteria were used - the magnitude of the risk of mastitis, that is, the assessment of the probability of developing mastitis, and the direction of the dynamics of changes in the magnitude of risk (its increase) in recent days during daily automatic monitoring, as well as several additional logical rules. The presence of a high risk in combination with the parameters of its growth dynamics makes it possible to identify not only the presence of subclinical mastitis, but also the actual risk of its development. The diagnostic error of the intelligent mastitis risk monitoring system was 5

The implementation of the fourth and fifth stages of the creation of a complete health monitoring system is planned to be carried out on the basis of the Open Semantic Technology for Intelligent Systems (OSTIS) [2]. On this basis, the most transparent prospects for the ontological design, production and operation of semantically compatible hybrid intelligent computer health monitoring systems and other intelligent medical systems open up.

IV. THE TASK OF INTEGRATING MEDICAL SYSTEMS

The system of individual health FSD-monitoring should be integrated into the general system of intelligent and conventional, non-intelligent computer systems for medical purposes. The task of integrating many existing and projected medical systems is key to reaching a new level of organization of the industry and the healthcare system. The healthcare organization needs to be rethought in the context of the integration of many computer systems for medical purposes based on modern technological means of artificial intelligence. The OSTIS technological system claims maximum compliance with the level of complexity of medical tasks.

Today, the main obstacle to the construction of intelligent medical systems is the difference in the conceptual apparatus of different areas of medicine. Overcoming this is possible only by ensuring semantic compatibility and semantic convergence of subsystems in the design process of medical intelligent systems [3]–[6].

Ensuring semantic compatibility and semantic convergence creates the basis for achieving interoperability

of intelligent computer systems. Interoperability should be understood as the interaction of intelligent computer systems based on mutual understanding between these systems themselves. Achieving the quality of interoperability allows the construction of a multi-agent network of interoperable intelligent computer systems for medical purposes.

The multi-agent architecture of ostis-systems involves the development of communicative aspects of machine "thinking", which can be traced in multi-agent systems [7]–[9].

OSTIS Technology makes it possible to effectively solve the problems of integrating various medical systems into a single interoperable network. To do this, it is necessary and sufficient to create an intelligent user interface (IPI) for each individual medical system, many of which form a single interoperable network. The IPI network will combine (integrate) traditional medical information systems (MIS), conventional diagnostic systems (DS), intelligent diagnostic systems (IDS), consulting systems (CS), telemedicine systems (TMS), external medical knowledge bases (MBZ) and other computer systems for medical purposes.

As part of the OSTIS development strategy, the concept of a personal intellectual assistant (secretary, referent) has been developed.

In line with this concept, the intelligent health FSD-monitoring system will act as a personal intellectual consultant on individual health-improving and preventive regimes of a healthy lifestyle, on the expediency of additional diagnostic studies, on the desired timing of treatment to a doctor of a particular specialty. The ostis-systems development strategy includes the position of developing a set of tools for individual comprehensive permanent medical control and health monitoring within the framework of a personal intelligent assistant.

V. CONCLUSION

Doctors have understood the need for regular monitoring of the health of an individual and the population for a very long time. It is also well understood that the possibilities of such control are limited by the set of diagnostic technologies used, in which there are no positions with a sufficiently high efficiency/cost ratio, that is, with high efficiency and low cost. In relation to the tasks of health control, effectiveness is determined, first of all, from the point of view of the possibilities of early diagnosis of diseases.

Today, the system of medical examination can be conditionally attributed to the monitoring of an individual's health, but the frequency of a year or two is too high for effective monitoring. The cost of functioning of the medical examination system is also high and, unfortunately, the diagnostic effectiveness of the existing medical examination system is low, which was shown in

the pilot project of using FSD diagnostics in medical examination with subsequent verification of diagnoses [10].

The creation of an intelligent health FSD-monitoring system will make it possible to move from a medical examination system to a health monitoring system and reach a new technological level of public health control.

REFERENCES

- [1] The medical spectral-dynamic complex. Available at: <http://http://www.kmsd.su> (accessed 2009, Sep)
- [2] V. V. Golenkov, N. A. Gulyakina Printsipy postroeniya massovoi semanticheskoi tekhnologii komponentnogo proektirovaniya intellektual'nykh sistem [Principles of construction of mass semantic technology of component design of intelligent systems]. *Open semantic technologies of design of intelligent systems* (OSTIS-2011): materials of the International Scientific and Technical Conference. Minsk, February 10-12, 2011). Minsk: BSUIR, 2011, pp. 21–58.
- [3] J. Jung, I. Choi, M. Song An integration architecture for knowledge management systems and business process management systems. *Computers in Industry*, 2007, Vol. 58, pp. 21–34.
- [4] V. V. Golenkov, N. A. Gulyakina Semanticheskaya tekhnologiya komponentnogo proektirovaniya sistem, upravlyayemykh znaniyami [Semantic technology of component design of knowledge-driven systems]. *Open semantic technologies of intelligent systems design*, 2015, No. 5, pp. 57–78.
- [5] B. A. Kobrinskii, A. E. Yankovskaya Convergence of Applied Intelligent Systems with Cognitive Component. *Open Semantic Technologies for Intelligent System*. Cham, Communications in Computer and Information Science, 2020, vol. 1282, pp. 34–47.
- [6] B. A. Kobrinsky Argumentatsiya i intuitsiya v estestvennom i iskusstvennom intellekte [Argumentation and intuition in natural and artificial intelligence]. VI National Conference on Artificial Intelligence with international participation KII'98: Proceedings of the conference, Pushchino, 1998, Vol. 1, pp. 7–14.
- [7] V. B. Tarasov Ot mnogoagentnykh sistem k intellektual'nym organizatsiyam: filosofiya, psikhologiya, informatika [From multi-agent systems to intellectual organizations: philosophy, psychology, informatics]. M., Editorial URSS, 2002, P. 352.
- [8] V. I. Gorodetsky Samoorganizatsiya i mnogoagentnye sistemy. I. Modeli mnogoagentnoi samoorganizatsii [Self-organization and multi-agent systems. I. Models of multi-agent self-organization]. *News of the Russian Academy of Sciences. Theory and control systems*, 2012, No. 2, pp. 92–120.
- [9] V. I. Gorodetsky Samoorganizatsiya i mnogoagentnye sistemy. II. Prilozheniya i tekhnologiya razrabotki [Self-organization and multi-agent systems. II. Applications and development technology]. *News of the Russian Academy of Sciences. Theory and control systems*, 2012, No. 3, pp. 102–123.
- [10] V. N. Rostovtsev, T. I. Terekhovich, A. N. Leaders, I. B. Marchenkova Diagnosticheskii skrining v sisteme dispanserizatsii [Leaders Diagnostic screening in the system of medical examination]. *Issues of organization and informatization of healthcare*, 2018, No. 2 (95), pp. 39–46.

Интеллектуальные системы мониторинга здоровья

Ростовцев В. Н.

В работе рассмотрены проблемы современного состояния комплексного мониторинга здоровья человека, а также соответствующих диагностических технологий. Предложен подход к интеллектуализации процесса регулярного контроля здоровья на основе систем ФСД-диагностики.

Received 13.03.2023

Technology of Neurological Disease Recognition Using Gated Recurrent Unit Neural Network and Internet of Things

Uladzimir Vishniakou and Yiwei Xia and Chuyue Yu
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus

Email: vish@bsuir.by, minskxiayiwei@qq.com, ycy18779415340@gmail.com

Abstract—In this paper, authors proposed a neurological disease recognition technique using gated recurrent unit neural network and supporting Internet of Things (IoT), which was checked by taking Alzheimer’s disease (AD) and Parkinson’s disease (PD) as examples. In this method first pre-emphasized and denoised the voice data, then segmented the voice signals with a sliding fixed window using the Hamming window function. Then we were extracted the eGeMAPSv02 voice features from the window signal, fed the features into the gated recurrent unit neural network model for its training, testing and achieve the disease diagnosis. The results of the study showed that despite the limited generalization ability of the gated recurrent unit model, it can still efficiently achieve voice recognition detection of a portion of neurological diseases. The model is implemented on the basis of the IoT platform for building a subsystem of IT diagnostics of patients as part of the smart city project. The code is stored in <https://github.com/HkThinker/Technology-of-neural-disease-recognition>.

Keywords—gated recurrent unite neural network, Internet of things, voice recognition, neurological disease

I. INTRODUCTION

Neurological disease usually result in structural or functional changes in the nervous system, causing patients to suffer from perception, thinking, emotion and behavior, and present a significant challenge to the global healthcare system. They are a group of diseases that affect the nervous system and include a variety of disorders such as neurodegenerative diseases, autoimmune diseases, cerebrovascular diseases, and brain injuries. For example, PD is a neurodegenerative disease that affects motion management and is characterized by symptoms such as hand tremors, limb stiffness, slow movements, and postural instability. AD is similar and results in memory loss, cognitive decline, and abnormal language and behavior. They tend to occur in older age groups, currently have no complete medical cure, but early diagnosis and prompt treatment can alleviate symptoms and slow progression. Traditional diagnosis of neurological diseases is usually based on doctors’ clinical experience, medical history, physical examination and specific tests, which has limitations and requires a lot of labor and resources. In recent years, with the rapid development of artificial intelligence and IoT technologies, neurological

disease identification technologies using neural networks and supporting IoT are expected to become a new breakthrough point.

The main purpose of this paper is to investigate the Gated Recurrent Unit (GRU) neural networks and IoT technologies to recognition for neurological diseases. To be specific, our research aims to achieve the following objectives:

- 1) To develop a GRU neural network model, which was trained through a publicly available database to implement the diagnosis and prediction of PD and AD.
- 2) By using IoT technology, we collected patients’ voice data and combined these data with the GRU neural network model to improve the precision and accuracy of diagnosis and prediction of neurological diseases.
- 3) To deploy the GRU neural network model to the Thingspeak IoT platform.

II. RELATED WORK

A. Application of IoT in Neurological Disease Diagnosis

Neurological disease diagnosis systems that are based on neural network technology and IoT technology have been widely used.

B. Lu [1] built a practical brain MRI-based AD diagnostic classifier using deep learning/transfer learning on datasets of unprecedented size and diversity. The purpose of Mukherji [2] was to identify non-invasive, inexpensive markers and develop neural network models that learn the relationship between those markers and the future cognitive state. David Payares-Garcia [3] proposed a classification technique that incorporates uncertainty and spatial information for distinguishing between healthy subjects and patients from four distinct neurodegenerative diseases: AD, mild cognitive impairment, PD, and Multiple Sclerosis. Abbas Sheikhtaheri [4] aimed to identify and classify the IoT technologies used for AD dementia as well as the healthcare aspects addressed by these technologies and the outcomes of the IoT interventions.

Researchers had identified the feasibility of integrating deep learning, cloud, and IoT, Syed Saba Raof [5] explained a summary of various techniques utilized in smart healthcare, i.e., deep learning, cloud-based-IoT applications in smart healthcare, fog computing in smart healthcare, and challenges and issues faced by smart healthcare and it presents a wider scope as it is not intended for a particular application such as patient monitoring, disease detection, and diagnosing and the technologies used for developing these smart systems are outlined. Reyazur Rashid Irshad [6] proposed a novel healthcare monitoring system that tracks disease processes and forecasts diseases based on the available data obtained from patients in distant communities. Rafael A Bernardes [7] presented a perspective on integrating wearable technology and IoT to support telemonitoring and self-management of people living with PD in their daily living environment.

B. Classification of Voice Features

Since more than 90 % of PD patients have varying degrees of dysphonia in the early stages of the disease, the diagnosis of PD based on voice features has the merits of being non-invasive and convenient. Darley [8] first used voice to diagnose aphasia in 1969. Saker et al. [9] preprocessed the voice data and extracted features, then applied SVM and KNN classification algorithms to the feature matrix for classification, eventually obtaining an average accuracy and a maximum accuracy of 55 % and 85 %, respectively, which initially confirmed the feasibility of voice features to classify PD. To further improve the accuracy of model prediction and simplify the algorithmic model, scholars have applied different feature selection algorithms.

III. METHODOLOGY AND DATASETS

A. Pre-emphasis and Denoising of Voice Signals

It is difficult to obtain the high-frequency part of the unprocessed voice signal because the power of the voice signal will be significantly attenuated after the sound gate excitation as well as the influence of mouth and nose radiation, combined with the smaller energy corresponding to the high frequency while the larger energy corresponding to the low frequency in the spectrogram of the voice signal. In order to facilitate the spectrum analysis, this paper adopted a first-order FIR high-pass digital filter for the pre-emphasis processing of the voice signal. The purpose of pre-emphasis is to improve the high-frequency part, so that the spectrum of the voice signal becomes flat, thus the spectrum can be obtained with the same signal-to-noise ratio in the whole frequency band.

Voice denoising is an effective part of signal pre-processing, mainly to improve the quality of voice and obtain more pure voice signals. The Fig. 1 shows the process of voice signal denoising.

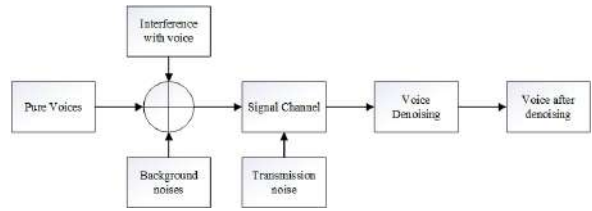


Figure 1. Flow chart of voice denoising.

According to the different parts of the noise introduction, voice noise can be divided into background noise and transmission noise. In this paper, spectral subtraction algorithm was used to denoise the voice. The spectral subtraction algorithm is designed based on the principle that pure voice is statistically independent of the noise signals.

B. Framing and Windowing of Voice Signals

The Fourier transform commonly used in voice signal processing calls for a smooth signal, but the main feature of the voice signal is the short-time smoothness, i.e., the stability of the voice signal in 10–30 ms period. Therefore, if we want to characterize the voice signal, it is necessary to analyze the short-time characteristics of the voice signal, the original signal is framed, and the frame frequency signal with short-time smoothness is derived. In the process of frame splitting, the signal tends to produce spectral deficiencies, so a windowing process must be performed between frames to keep the signal at the truncation without distortion. The windowing function used in this paper is the Hamming window function, with window size of 1024, frequency of voice signal is 44.1kHz, and the overlap rate of window is 50 %, hence the voice time of one window is about 23 ms.

C. Feature Extraction of Voice Signals

We used an extended version of GeMAPS (Basic Affective Parameter Set), eGeMAPSv02 [10], a speech feature set. It uses acoustic features and spectral-based features to describe the speech signal, with a total of 88 features. It contains 25 low-level descriptor features, namely pitch, jitter, gating frequency, gating bandwidth, gloss, loudness, harmonic-to-noise ratio (HNR), Alpha ratio, Hammarberg index, spectral slope 0–500 Hz, spectral slope 500–1500 Hz, 3 gating relative energies, 3 relative energies, 3 harmonic differences, 4 Mel–Frequency Cepstral Coefficients, 1 spectral flux. 53 other parameters are derived from these basic parameters.

D. 6-layer Gated Recurrent Unit Model

In the paper, a multi-layer GRU model is constructed for voice data recognition. Two mechanisms, an update gate and a reset gate, are included in the GRU module. The internal equation of a single GRU model is :

$$r_t = \sigma(W_r \times [h_{t-1}, x_t]) \quad (1)$$

$$z_t = \sigma(W_z \times [h_{t-1}, x_t]) \quad (2)$$

$$\tilde{h}_t = \tanh(W_{\tilde{h}} \times [r_t \times h_{t-1}, x_t]) \quad (3)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \quad (4)$$

$$y_t = \sigma(W_\sigma \times h_t) \quad (5)$$

Where σ represents sigmoid function, z_t is the update gate of the unit, sigmoid function converges the value of the update gate to 0 or 1, i.e., whether the value of the previous step is remembered or discarded. r_t is the reset gate, the smaller r_t , the more information about the previous state needs to be ignored, W is the weight value, h_t and \tilde{h}_t are the output and temporary hidden states in the module.

The GRU model has a lower computational cost with faster training, so the model is extensively used in various fields of deep learning. The structure of a single GRU module is shown in Fig. 2.

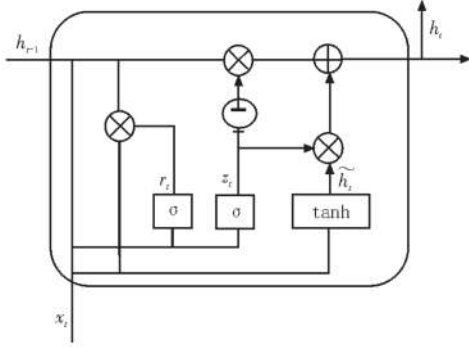


Figure 2. Structure of a single GRU model.

The GRU neural network in this work had a total of 6 layers and 1700 learnable properties. Table III illustrated the structure of the GRU neural network.

Table I
THE STRUCTURE OF GRU NEURAL NETWORKS

Name	Activations	Learnable Properties
Sequence Input	$88(C) \times 1(B) \times 1(T)$	
GRU	$6(C) \times 1(B)$	InputWeights 18×88 RecurrentWeights 18×6 Bias 18×1
ReLU	$6(C) \times 1(B)$	
Fully Connected	$2(C) \times 1(B)$	Weights 2×6 Bias 2×1
Softmax	$2(C) \times 1(B)$	
Classification Output	$2(C) \times 1(B)$	

E. Public Voice Datasets Used

This paper used public datasets [11] collected from 188 PD patients (107 men, 81 women) aged 33–87 at Istanbul University. The control group includes 64 healthy individuals (23 men, 41 women) aged 41–82.

Participants were asked to sustainably pronounce the vowel /a/ while a microphone set at 44.1 KHz recorded their voice three times.

DementiaBank [12] is a resource that collects voice, video, and text data from older adults and patients with AD. It contains two groups of participants; the elderly group includes 60 healthy older adults from New York City who ranged in age from 60 to 91 years, while the AD group includes 64 patients from Pittsburgh who ranged in age from 60 to 95 years. Each participant was asked to answer a series of questions. Data were collected using a specialized recording device, with recorded voice data at a sampling rate of 44.1 kHz.

Visualization of voice can help to extract feature information. The voice waveform and spectrum of AD are shown in Fig. 3.

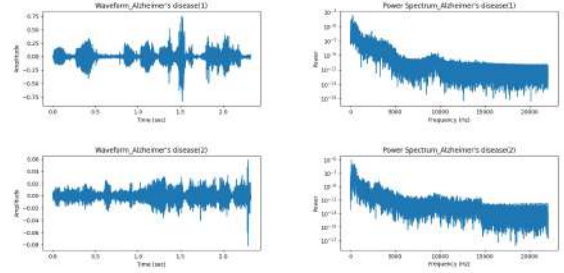


Figure 3. Alzheimer data voice waveforms and spectrograms.

F. Training and Testing Process

After noise removal and signal segmentation of all the voice data in the dataset, we extracted 88 voice signal features for each voice window signal, and then created a neural network for training and learning. In this paper, a 6-layer GRU model was adopted. Its model structure is shown in Fig. 4.

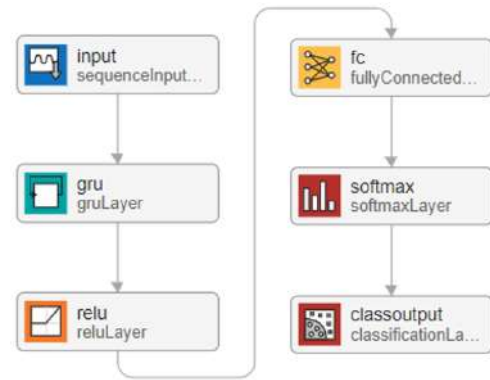


Figure 4. Structure of 6-layer GRU model.

To avoid overfitting, we added a Relu layer after the GRU layer and output the probability of class labels by

defining a Softmax layer to vectorize the labels one-hot before calculating the correctness. To accelerate the model convergence, the training took a batch gradient descent approach for weight update, and each batch contained 64 features. A flow chart of the overall model structure of the experiment is shown in Fig. 5.

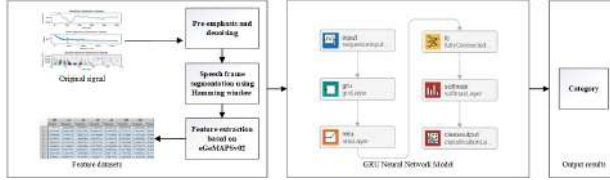


Figure 5. Flow chart of the overall model structure of the experiment

G. Deploying the Model to the Thingspeak Platform

Thingspeak is an open source IoT application platform that allows users to conveniently collect, process and analyze data from IoT devices. The platform provides a way for developers and manufacturers to collect, store, analyze and visualize data from the center of IoT devices and use that data for real-time decision-making and operations.

To upload the sensor data from the mobile phone to the Thingspeak IoT platform and read the results of the data analysis using an application developed by ourselves, we followed the following steps:

- 1) Registering a Thingspeak account and creating new channel.
- 2) Getting our channel write/read API Key, which we can find in our Thingspeak account.
- 3) Adding network authority and sensor permission in our application.
- 4) Adding the Thingspeak API library, we can get the source code of the library from the Thingspeak website.
- 5) Implementing the code to upload data in our application, the code should use HTTP protocol to upload our sensor data to our Thingspeak channel, providing the channel write API Key to authenticate our identity.
- 6) After uploaded data, we can analyze the data using Thingspeak's analytics tool. Once we have uploaded the data, we could use Thingspeak's analytics tool to analyze the data. To get the result of data analysis by using HTTP GET request.
- 7) Implementing the code to read the analysis results in our application. We need to get data analysis results using HTTP GET request and read API key to parse the results into JSON format so that we can process and display the data in our application.

In summary, to upload the sensor data from the phone to the Thingspeak IoT platform and read the data analysis

results, we need to register an account and create a new channel, get the channel write/read API Key, add network permissions and sensor permissions, add the Thingspeak API library, implement the code to upload the data, use the platform's analysis tool to analyze the data, implement the code to read the analysis results, and parse the results into JSON format to process as well as display the data in the application.

Deployment of the GRU model to the Thingspeak IoT platform for data analysis Data analysis on Thingspeak using the GRU model involves the following steps:

- 1) Creating a new channel on Thingspeak to store the data to be analyzed. We can use Thingspeak's REST API or MQTT API to add the sensor data to the channel.
- 2) Training a GRU model on our local computer and exporting the model to a format that can be used on.
- 3) Uploading the exported KNN model to the Thingspeak platform. We can use Thingspeak's REST API or MQTT API to upload the model to the channel.
- 4) Once the model is uploaded successfully, we can use Thingspeak's MATLAB analysis toolbox or matlab scripts to load the model and classify the uploaded data. In MATLAB, we can read the uploaded data using the thingSpeakRead function, load the GRU model using the load function, and classify the data using the predict function.
- 5) Displaying the classification results on the user interface of Thingspeak or sending the results to our cell phone as well as to an email for easy viewing of the identification results.

IV. EXPERIMENTS AND RESULTS

A. Experimental Setup

In this paper, the feature datasets were divided into training datasets and test datasets in the ratio of 9:1. The training datasets were trained and validated using the 5-fold cross-validation method, which was repeated five times. The test datasets were then used to test the final results. And We evaluated the experiment using the confusion matrix [13].

The Table II showed the GRU neural network model hyperparameter setting table in this experiment.

B. Experiment Results and Evaluation

The Fig. 6 showed the process of training the GRU model in 1000 epochs based on the Parkinson's public voice datasets.

As seen in the Fig. 6, the GRU neural network model based on the Parkinson's public voice dataset can converge substantially in a short time. The model uses stochastic gradient descent and variable learning rate in solving the minimization loss function, so there was

Table II
GRU NEURAL NETWORK MODEL HYPERPARAMETER SETTING

Number	Parameter Name	Parameter Value
1	Mini Batch Size	64
2	Max Epochs	1000
3	Initial Learn Rate	0.01
4	Learn Rate Drop Factor	0.1
5	Learn Rate Drop Period	700
6	Shuffle	every-epoch
7	optimization	adam

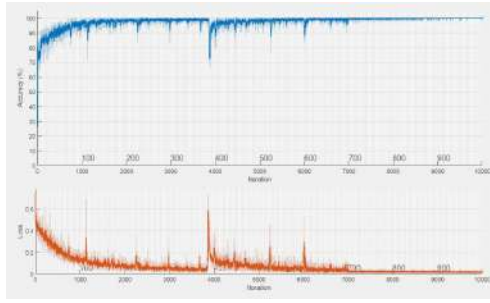


Figure 6. The process of training the GRU model based on Parkinson's datasets in 1000 epochs.

some jitter in the convergence process of the model, but the general trend of the model accuracy was improved, the loss function of the model corresponded to a decreasing trend. The final training accuracy of the model reached 100 %.

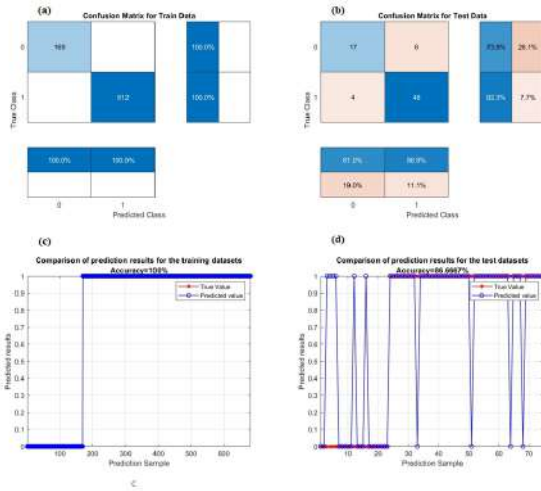


Figure 7. a - confusion matrix of training datasets; b - confusion matrix of testing datasets; c - prediction results for the training datasets; d - prediction results for the testing datasets.

As can be seen from Fig. 7, the accuracy of the model on the test set was 86.66 %, while the accuracy on the training set was much better than the accuracy on the test set, the model may have been overfitted. The overfitting phenomenon may arise because of the small amount of data in the public voice dataset of Parkinson's, coupled

with the uneven distribution of samples in this public dataset, so the model's performance was degraded.

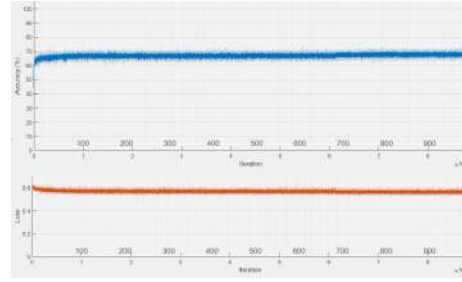


Figure 8. The process of training the GRU model based on Alzheimer's datasets in 1000 epochs.

As can be seen in Fig. 8, the model converged after the Alzheimer's voice training dataset was fed into the model and entered 2000 training cycles. The Fig. 9 showed a comparison of the prediction results and the confusion matrix of the training and testing datasets for AD.

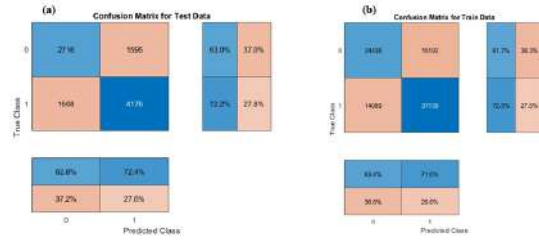


Figure 9. a - confusion matrix of testing datasets; b - confusion matrix of training datasets.

Table III showed the experimental results of PD recognition and AD using GRU based on the test datasets.

Table III
THE EXPERIMENTAL RESULTS OF PD RECOGNITION AND AD USING GRU BASED ON THE TEST DATASETS

Public Datasets	Average Precision	Average Sensitivity	Average F1 score	Test Accuracy
Parkinson's	84.95 %	83.10 %	84.01 %	86.66 %
Alzheimer's	67.60 %	67.50 %	67.55 %	68.27 %

In summary, the accuracy of the GRU-based PD model could reach 86.67 % on the test dataset and 100 % on the training dataset. On the testing datasets, the average precision was 84.95 %, the average sensitivity was 83.10 %, and the average F1 score was 84.01 %. This experimental result showed that the recognition of PD using GRU algorithm based on freezing of gait data was effective.

However, the test results of the model on Alzheimer's data were not satisfactory, which may be due to the fact that Alzheimer's data were more complex and harder to find feature points compared to Parkinson's data, after which we will try new models or improve the model in a

way to increase the accuracy. The model is implemented on the basis of the IoT platform for building a subsystem of IT diagnostics of patients as part of the smart city project using elements of OSTIS technology [14].

V. CONCLUSION

The aim of this paper was to explore the performance of GRU neural networks in voice recognition tasks within neurological diseases. We used a 6-layer GRU model that was trained and tested on the Parkinson's public voice dataset and the Alzheimer's public voice dataset. With the experimental results, we found that.

- 1) On the Parkinson's public voice dataset, our model can achieve 86.66 % accuracy, which has better performance than traditional machine learning methods. However, the accuracy on the Alzheimer's public voice dataset was only 68.27 %, indicating that the 6-layer GRU model does not have good generalization ability.
- 2) During the training of the model, we noticed that the training error of the model was gradually decreasing with the increase of training times, but the testing error started to increase. This indicated that the model appeared overfitting phenomenon.
- 3) We also explored our scheme to implement the GRU model on the IoT. The scheme has potential for practical applications and provides a reference for research in related fields.

Summing up, our experimental results showed that the GRU model can be deployed on IoT platforms to solve part of the problem of IT diagnostics of neurological disorders by recognizing changes in patients' speech.

REFERENCES

- [1] B. Lu, B. Li, H. Chang, Z. Li, L. Chen, N. Zhu, Z. Zhou, H. Li, X. Wang, Y. Cui *et al.*, "A practical alzheimer disease classifier via brain imaging-based deep learning on 85,721 samples [preprint]," *Neuroscience*, 2020.
- [2] D. Mukherji, M. Mukherji, and N. Mukherji, "Early detection of alzheimer's disease using neuropsychological tests: a predict-diagnose approach using neural networks," *Brain Informatics*, vol. 9, no. 1, pp. 1–26, 2022.
- [3] D. Payares-Garcia, J. Mateu, and W. Schick, "Spatially informed bayesian neural network for neurodegenerative diseases classification," *Statistics in medicine*, vol. 42, no. 2, pp. 105–121, 2023.
- [4] A. Sheikhtaheri and F. Sabermahani, "Applications and outcomes of internet of things for patients with alzheimer's disease/dementia: A scoping review," *BioMed Research International*, vol. 2022, 2022.
- [5] S. Saba Raoof, M. Durai *et al.*, "A comprehensive review on smart health care: Applications, paradigms, and challenges with case studies," *Contrast Media & Molecular Imaging*, vol. 2022, 2022.
- [6] R. R. Irshad, A. A. Alattab, O. A. S. Alsaiari, S. S. Sohail, A. Aziz, D. i. Madsen, and K. M. Alalayah, "An optimization-linked intelligent security algorithm for smart healthcare organizations," in *Healthcare*, vol. 11, no. 4. MDPI, 2023, p. 580.
- [7] R. A. Bernardes, F. Ventura, H. Neves, M. I. Fernandes, and P. Sousa, "Wearable walking assistant for freezing of gait with environmental iot monitoring: A contribution to the discussion," *Frontiers in Public Health*, vol. 10, 2022.
- [8] F. L. Darley, A. E. Aronson, and J. R. Brown, "Differential diagnostic patterns of dysarthria," *Journal of speech and hearing research*, vol. 12, no. 2, pp. 246–269, 1969.
- [9] B. E. Sakar, M. E. Isenkul, C. O. Sakar, A. Sertbas, F. Gurgen, S. Delil, H. Apaydin, and O. Kursun, "Collection and analysis of a parkinson speech dataset with multiple types of sound recordings," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 4, pp. 828–834, 2013.
- [10] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan *et al.*, "The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing," *IEEE transactions on affective computing*, vol. 7, no. 2, pp. 190–202, 2015.
- [11] C. O. Sakar, G. Serbes, A. Gunduz, H. C. Tunc, H. Nizam, B. E. Sakar, M. Tutuncu, T. Aydin, M. E. Isenkul, and H. Apaydin, "A comparative analysis of speech signal processing algorithms for parkinson's disease classification and the use of the tunable q-factor wavelet transform," *Applied Soft Computing*, vol. 74, pp. 255–263, 2019.
- [12] A. M. Lanzi, A. K. Saylor, D. Fromm, H. Liu, B. MacWhinney, and M. L. Cohen, "Dementiabank: Theoretical rationale, protocol, and illustrative analyses," *American Journal of Speech-Language Pathology*, pp. 1–13, 2023.
- [13] A. Luque, A. Carrasco, A. Martín, and A. de Las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognition*, vol. 91, pp. 216–231, 2019.
- [14] D. Shunkevich, "Agent-oriented problem solvers of intelligent systems component," *autoref. of PhD thesis in tech. sciences: 05.13.17 / Belarusian State University of Informatics and Radioelectronics, Minsk*, pp. 1–24, 2018.

Технология распознавания нейрологических заболеваний с использованием нейронной сети закрытого рекуррентного блока и интернета вещей

Вишняков В. А., Ивей С., Чуюэ Ю.

В этой статье авторы предложили метод распознавания неврологических заболеваний с использованием закрытой рекуррентной нейронной сети и поддержки Интернета вещей (IoT), который был проверен на примере болезни Альцгеймера (БА) и болезни Паркинсона (БП). В этом методе сначала предварительно выделяются и ослабляются голосовые данные, затем голосовые сигналы сегментируются с помощью скользящего фиксированного окна с использованием функции окна Хэмминга. Затем извлекаются голосовые характеристики eGeMAPSv02 из сигнала окна, вводятся эти характеристики в модель нейронной сети с закрытым рекуррентным модулем для ее обучения, тестирования и достижения диагноза заболевания. Результаты исследования показали, что, несмотря на ограниченную способность модели gated recurrent unit к обобщению, она может эффективно обеспечивать распознавание голоса при выявлении части неврологических заболеваний. Модель реализуется на базе платформы IoT для построения подсистемы ИТ-диагностики пациентов в рамках проекта умного города. Код хранится <https://github.com/HkThinker/Technology-of-neural-disease-recognition>.

Ключевые слова — нейронная сеть с закрытым рекуррентным модулем, сеть Интернета вещей, распознавание голоса, неврологические заболевания.

Термины индекса — закрытая рекуррентная нейронная сеть, Интернет вещей, распознавание голоса, неврологические заболевания.

Received 12.03.2023

Hand Gesture Recognition Based on Skeletal Image Properties

J. Ma and V. Yu. Tsviatkou and A. A. Boriskevich

Belarusian State University of Informatics and Radioelectronics

Minsk, Belarus

Email: majun1313@hotmail.com, vtsvet@bsuir.by, anbor@bsuir.by

Abstract—Hand gesture Recognition is an important task and can be used in a lot of applications. In intelligent systems, hand gesture recognition can be used to access information through a video interface. In recent years, skeleton-based hand gesture recognition become a popular research topic. The existing methods have the low discriminative power due to sensitivity of features to image noise. We have proposed new methods to decrease the influence of the noise to extract hand image features. The objective of the research is to improve the hand gesture classification accuracy. A hand gesture recognition method based on skeleton image properties is developed. For 5 classes recognition, this approach allows us to increase the classification accuracy on test set from 0.4% till 20.4% as compared with existing well-known methods. For 10 classes recognition, this approach allows us to increase the classification accuracy from 5% till 18% as compared with existing well-known methods.

Keywords—Color images, Skeleton images, hand gesture feature, Machine Learning, Classification Accuracy

I. INTRODUCTION

Hand gesture recognition is widely used in many applications such as sign language recognition [1], clinical and health [2], and robot control [3]. Open semantic technologies provide the ability to access the knowledge base of an intelligent system using a video interface. The inclusion of a hand gesture recognition system in the video interface makes it easier to enter commands and data into an intelligent system.

There are two existing practical approaches to recognizing hand gestures [4]. The first approach is based on data gloves (wearable or direct contact) [5], [6], and the second is based on computer vision, which does not require special sensors except cameras [7]. Moreover, vision-based methods can provide contactless communication between humans and computers. Therefore they are considered ordinary, suitable approaches.

As one of the types of camera vision-based approaches [4], skeleton-based approaches are attracting much attention in recent years since the skeleton feature describes geometric attributes and constraints and easily translates features and data correlations [8]. The skeleton-based approaches can further be classified as RGB-based [9] and RGB-D-based [10]- [12] approaches according to the different ways of obtaining the skeletal images. The

RGB-D-based approaches adopt the depth sensor of the Kinect camera to obtain the skeletal image. One of the merits of these methods is that the lighting, shade, and color did not affect the obtained skeletal image. However, the depth camera's cost, size, and availability will limit their use. On the contrary, RGB-based approaches only require standard cameras. However, it must first convert the RGB images into grayscale images and then follow binarization and skeletonization to extract the skeletons. The skeletons extracted by this kind of method may include many useless skeletal branches or skeletal rings that are caused by the noise. The noise problem will be evident when the contrast of the input image is low. Since the existence of the noise, the accuracy of hand gesture recognition using RGB skeleton-based approaches are not satisfying. However, it would be a promising method if the effect of the noise could be reduced.

In the past several decades, many denoise methods have been proposed to alleviate the effects of noise on the skeletonization algorithm and produce stable skeletons as much as possible. These methods can be concluded into three different types, which are skeletonization-based denoising approaches [13]- [16], pruning-based denoising approaches [17]- [19], and scale-space-based denoising approaches [20], [21].

In this paper, a hand gesture recognition system based on skeleton image properties is developed, in which skeleton images are extracted by using different combinations of the skeletonization and denoising method. The objective of the research is to improve hand gesture classification accuracy.

II. RELATED METHODS

In this section, some skeletonization and denoising methods used in our hand gesture recognition system are introduced. There are five skeletonization methods implemented, two of which are classical skeletonization methods and others proposed by us. In addition, a post-pruning method and a space-based denoising approach are also deployed in this system.

A. Image Skeletonization Method

OPTA algorithm [22] is a classical parallel skeletonization method proposed by Roland T. Chin et al. This

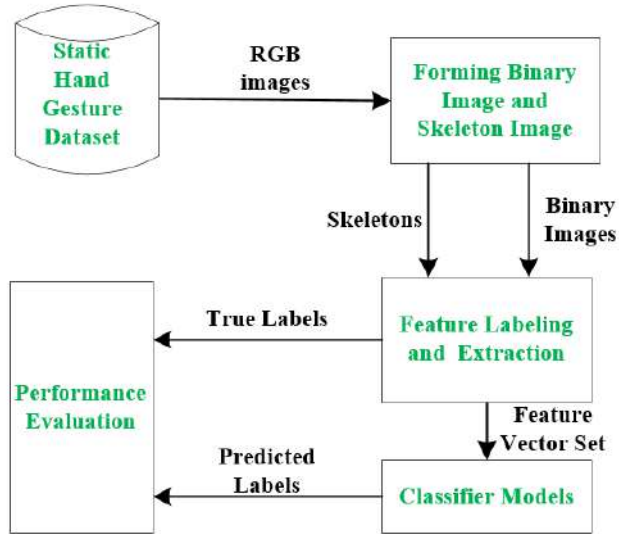


Figure 1. General block-scheme of the hand gesture recognition.

algorithm uses eight 3×3 thinning templates to remove the pixels. In addition, two other restoring templates are applied to address the breakage and disappearance of horizontal and vertical limbs with double-pixel widths. The drawback of this algorithm is that it is susceptible to noise.

ZS algorithm [23] is another classical parallel skeletonization method, which is one of the most popular methods since it can offset the influence of the noise to some extent by breaking one iteration in OPCA into two sub-iterations. Therefore, the computation speed of the ZS method is slower than OPTA. In addition, the ZS algorithm has some potential problems, such as sometimes it may suffer the problem of excessive erosion, and it fails to maintain one-pixel width, which may increase the difficulty when applied to recognition tasks.

Based on these two classical methods, we have proposed three new skeletonization methods: OPCA, ZSM, and OPTA.

OPCA [24] is a denoising version of the OPTA algorithm by modifying some deletion conditions. It is more robust than the OPTA method and, at the same time, shares a similar computation speed with the OPTA method. In addition, it can achieve a single-pixel width. However, it is more sensitive to the noise than the ZS algorithm.

ZSM [25] is an improved version of the ZS algorithm, in which the drawbacks of excessive erosion are overcome and it can achieve single pixel width by adopting extra five thinning templates. The denoise ability of this algorithm is similar to the ZS algorithm. However, this method is still a sub-iterative method as the ZS algorithm.

MOPCA [26] is the improved version of the OPCA, in which there is a total of 13 templates are used to enhance the robustness of the algorithm to the noise. This method combines the merits of ZS algorithms and OPTA algorithms. It is insensitive to noise as the ZS and as fast as the OPTA method.

B. Post-pruning and Scale-Space based Denoising Method

Using denoise-skeletonization can only partly offset the influence of the noise. Therefore, it is necessary to use other denoising techniques to further improve the noise-against ability. As a result, we also proposed a new post-pruning method and a new scale-space denoising method.

The post-pruning method proposed by us is named DCEM [27], modified from the famous pruning algorithm of DCE [28]. One of the limitations of the DCE is that it requires manual tuning of the parameter of the pruning power's strength. In DCEM, conducting this tedious work is unnecessary, making it more convenient in many applications.

Our proposed scale-space denoising method is ATFM [29], derived from the ATF [30] method. The core idea of the ATF method is first to extract skeletons from different smoothed images that are filtered by using different scale-space filters to the original image. Then, they used their proposed sensitive measure to evaluate these skeletons, from which the skeleton with the lowest score is considered stable. However, their method sometimes may suffered the problem of the deformation of the skeleton. To overcome this problem, we proposed our ATFM method. In our method, the significant modification on

the sensitive measure, in which more information is included.

III. PROPOSED HAND GESTURE RECOGNITION METHOD

The general block scheme of hand gesture recognition includes the following components, which are static hand gesture dataset, binary and skeleton image formation, feature labeling and extraction, classifier models, and performance evaluation. This general block-scheme is presented in Fig. 1, all major components are marked with green color.

Static hand gesture images are stored in the images dataset. They are used to train the classifier and evaluate the accuracy of the classification task. These RGB images of hand gestures are first passed into a block that can form the binary and skeleton images from them. All the proposed skeletonization and denoise methods are embedded in this block.

Next, feature extraction is conducted based on these obtained binary and skeleton images. For each pair of binary image and skeletal image, there are nine geometry features should be extracted, and they together compose a feature vector. Next, manual labeling for each pair of binary and skeleton images is also required to get the truth labels that corresponding to each feature vector. These feature vector can passed to the trained classifier for prediction. By comparing the predicted label and the truth label to compute the accuracy. In the classification module, there are six different well-known classifiers for optional, which includes decision tree(DT) [31], k-nearest neighbors (KNN) [32], naïve Bayes (NB) [33], support vector machine (SVM) [34], ensemble learning (EL) [35], multilayer perceptron (MLP) [36].

A. Creation of the Hand Gesture Dataset

All static hand gesture images that in dataset are captured with the iPhone 11. The resolution of images are $3024 \times 3024 \times 3$. Since directly processing these images is time-consuming, resize operation is used to converting these images into $95 \times 95 \times 3$ images. The dataset consists of ten different classes, example pictures are shown in Fig. 2.

In each one class, there are more than 100 different images. As a result, the total amount of our dataset is over 1000 images. These images are randomly divided into train-validation group and testing group. The number of images in testing group is equal to 20% of the initial image set, and the number of images in train-validation group is 80% of the initial image set.

B. Forming Binary Image and Skeleton Image using Hybrid Combining Denoising Techniques and Skeletonization Methods

The skeleton and pattern images are extracted from the original images by using different combinations of



Figure 2. Example of the Ten Class of Hand Gestures.

skeletonization method and denoise methods. There are six hybrid methods are used, which including ZS+ATFM, OPTA+ATFM, OPCA+ATFM, ZSM+ATFM, MOPCA+ATFM, and MOPCA+ATFM+DCEM. The time consumption of these methods is listed in Tab. I.

From Tab. I, it is noted that ZS+ATFM, OPTA+ATFM, ZSM+ATFM, MOPCA+ATFM, and MOPCA+ATFM+DCEM respectively spend more 38%, 22%, 33%, 0.4%, and 5% time when compared with the method of OPTA+ATFM. Besides, we can learned the use of DCEM may take extra 0.02 seconds.

In Fig. 3, we listed example skeletons extracted from

Table I
TIME CONSUMPTION OF SIX METHODS

Skeleton Image Extract algorithm	Average Time of skeleton process (s)
ZS+ATFM	0.704
OPTA+ATFM	0.624
OPCA+ATFM	0.510
ZSM+ATFM	0.682
MOPCA+ATFM	0.512
MOPCA+ATFM+DCEM	0.536

the images that are shown in Fig 2 by using the skeletonization method MOPCA with both ATFM and DCEM.

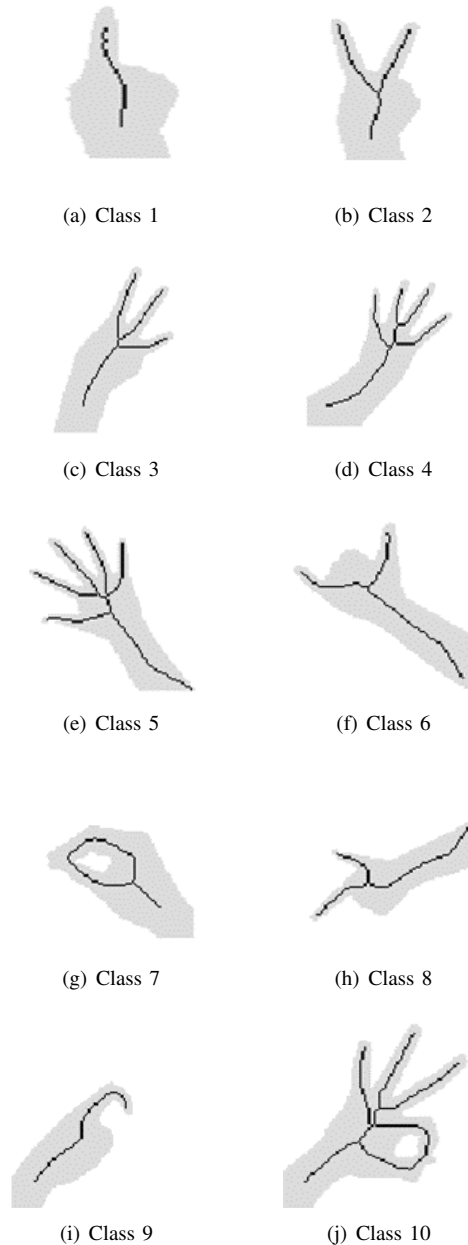


Figure 3. Skeleton examples of ten hand gesture classes.

C. Feature extraction based on Skeleton and Binary Images

After a skeletal image and its pattern image are obtained from an input image, it is necessary to transform the skeletal images along with its pattern image to a 9-dimension feature vector used in the later classification. This 9-dimension vector includes the following significant geometrical features: the number of endpoints (NEP); the number of cross points (NCP); the existence of the inner hole (EIH); the average virtual-real distance rate between each pair of endpoints (AVRD); the number of virtual cross points (NVCP); Rate of the deviation of the thick of the endpoints (RDTE); Average distance between the thickest point in a pattern image and each endpoint in the skeletal image (ADTPE); distance between pattern thickest point and skeletal thickest point (DPSP); average angle of the endpoint (AAEP). Each dimension of this feature vector is manually selected with respect to the topology of these different classes.

The NEP is obtained by summarizing the number of these foreground pixels, which have only one neighbor foreground pixel in its 8-neighborhood window in the skeletal image.

The NCP is obtained by summarizing the number of these foreground pixels, which have more than two neighbor foreground pixels in its 8-neighborhood window in the skeletal image.

The EIH is an important geometry feature with only two values, 0 or 1. The inner hole denotes that the hole should be enclosed by the skeleton. Ideally, only Class 7 and Class 10 have the inner hole. One method to judge the existence of the inner hole for these hand images is to compute the number of closed areas in the skeleton image.

The AVRDR describes the similarity of the real connecting line between endpoints to the virtual closet straight line between them. For each pair of endpoints, the real connecting line and its distance can be obtained using breadth-first search (BFS) algorithms, and the distance of the virtual line is calculated using the Euclidean distance formula. Then the average value is easily obtained.

The NVCP is obtained by summarizing the total number of points at the intersection of the virtual line and the real line.

Before presenting the definition of RDTE, ADTPE, and DPSP, the concept of thickness is first introduced. The thickness of a pixel is defined by the distance between this pixel and its closest pixel located on the boundary in the pattern image. Boundary pixels comprise the foreground pixel, whose four neighbors have at least one background pixel.

For a given skeleton with n endpoints, all endpoints can form a set S_{EP} , in which the i -th endpoint is denoted as S_{EP_i} . The thickness of S_{EP_i} can be denoted as T_{EP_i} . The set formed by all T_{EP_i} is denoted as $T_{S_{EP}}$. Then,

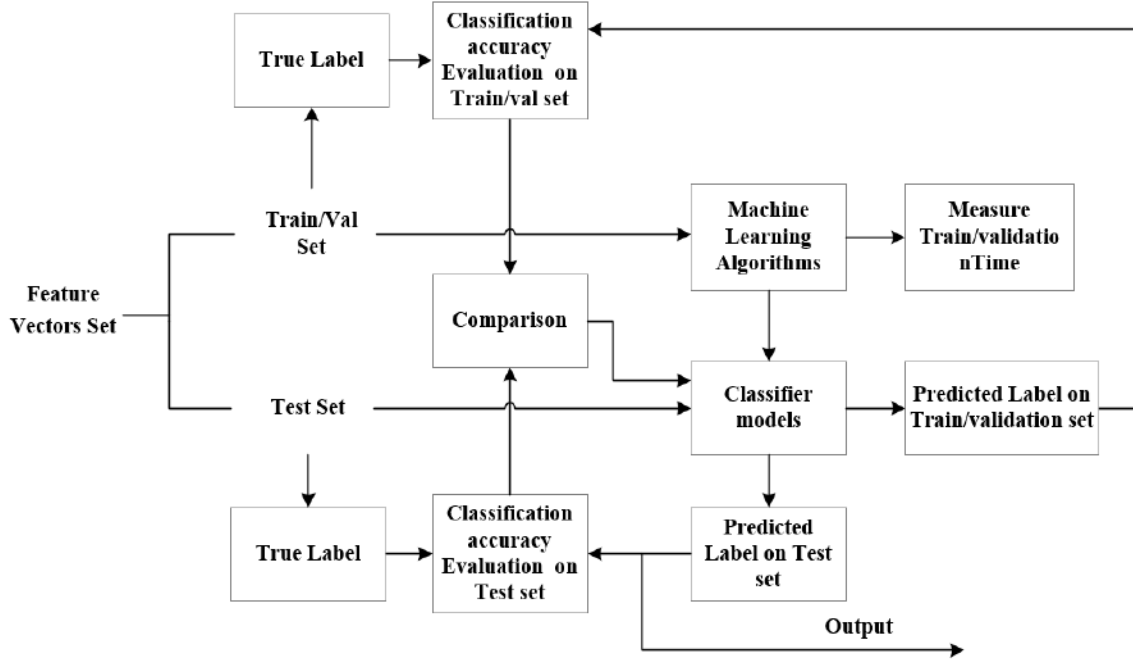


Figure 4. Classification Models and Performance Evaluation.

the RDTE for this skeleton can be computed by using the following formula:

$$RDTE = \begin{cases} 0 & n \leq 1 \\ \sum_{i=1}^n \frac{\sqrt{(TEP_i - \frac{1}{n} \sum_{i=1}^n TEP_i)^2}}{\max(TSEF) - \min(TSEF)} & n > 1 \end{cases} \quad (1)$$

We suppose the coordinates of the thickest pixel in the pattern image are P_x and P_y , and its thickness is T_p . We suppose that in a skeletal image, there are n endpoints. The coordinates of the i -th endpoint are denoted as EP_{i_x} and EP_{i_y} . Then, the ADTPE can be calculated by using the following formula:

$$ADTPE = \begin{cases} 0 & n = 0 \\ \frac{\sum_{i=1}^n \sqrt{(P_x - EP_{i_x})^2 + (P_y - EP_{i_y})^2}}{nT_p} & n > 0 \end{cases} \quad (2)$$

Supposing the coordinate of the thickest pixel in the pattern image is P_x and P_y , and the coordinate of the thickest pixel in the skeletal image is S_x and S_y , the DPSP can be calculated according to the following formula:

$$DPSP = \sqrt{(P_x - S_x)^2 + (P_y - S_y)^2} \quad (3)$$

Before obtaining the value of the AAEP, the main axis is defined by the thickest point in the pattern image and the farthest endpoint in the skeletal image from that point. Based on that, it is easy to calculate the relative angle of the remaining endpoint to these axes, and the

AAEP is the mean of these angles. If the number of endpoints is less than 2, the AAEP is set as 0.

D. Classifier Models and Performance Evaluation

The obtained feature vectors of the images from the training set of the dataset and their labels are passed to classifiers mentioned, then conduct the learning process. The hyperparameter of these classifiers is listed in Tab. II.

Then, the classifier's learning result are evaluated by considering the accuracy of these classifiers on the test set.

Here, our aim is to explore the relationship between the accuracy of the classifiers and the different skeleton extracted by different methods, the relationship between the accuracy of the classifiers, and the difference in the feature selection. In addition, we also study the difference between distinct classifiers and their performance under a different number of classes. The general block diagram is shown in Fig. 4.

There are many criteria to evaluate the classifier's performance, such as accuracy, F1, precision, recall, roc, and so on. Here, we only take accuracy as the evaluation criteria for simplification. The formula of accuracy is described in the following:

$$Accuracy = \frac{1}{m} \sum_{i=1}^m I(x_i, y_i) \quad (4)$$

$$I(x_i, y_i) = \begin{cases} 1 & f(x_i) = y_i \\ 0 & f(x_i) \neq y_i \end{cases} \quad (5)$$

Table II
HYPERPARAMETER OF DIFFERENT CLASSIFIERS

Classifier	Hyperparameter	Setting
DT	Maximum number of splits	100
	Split criterion	Gini
KNN	Number of neighbors	1
	Distance metric	Euclidean
	Distance weight	Equal
NB	Kernal type	Gaussian
SVM	Kernal function	Quadratic
	Box constraint level	1
	Kernel scale mode	Auto
EL	Ensemble Method	Bagged Trees
	Learning type	Decision Tree
	Maximum number of splits	99
	Number of learners	30
	Learning rate	0.1
MLP	Subspace dimension	1
	Number of fully connected layers	1
	Layer size	25
	Activation	Relu
	Iteration limit	1000

Where m is the number of the pair of feature vectors x and its corresponding true label y . $f(x)$ is predicted label for feature vector x .

IV. EXPERIMENTAL RESULTS

Two separate experiments are conducted for the purpose of evaluating the performance of the proposed recognition method. In the first experiment, the recognition experiment is conducted on five hand gesture classes, which include Class 1 to Class 5. Whereas in the second experiment, the recognition experiment is conducted on all ten hand gesture classes.

A. Performance evaluation of static hand gesture Classification for five classes

In order to explore the influence of the features on the classification task for five classes, an experiment of the feature selection has been conducted based on the classifiers of KNN and SVM, in which we removed one feature from the feature vector and conducted the classification task by using remaining features. Experimental results are shown in Tab. III.

From Table III, it is clear that the deletion of the NEP or the deletion of NCP may decrease the accuracy of the classification result in both KNN and SVM. Deleting the NVCP, ADTPE, and AAEP may improve the accuracy of the KNN classification, whereas deleting them may not change the performance of the SVM classification. The reason for that is the KNN classifier used in our experiment only considers the closest neighbors.

Next, the influence of the different skeleton extractions on the accuracy of all six classifiers is also studied. The results are shown in Tab. IV and Tab. V.

From Tab. IV and Tab. V, it is obvious that skeletonization methods can affect the accuracy of the classification. Among the methods with ATFM denoise

Table III
CLASSIFICATION ACCURACY COMPARISON FOR 8 AND 9 FEATURES AND 5 CLASSES

Feature Deleted	Classification Accuracy			
	Train/Validation Set		Test Set	
	KNN	SVM	KNN	SVM
NEP	92.30%	97.00%	88.80%	97.80%
NCP	93.80%	97.90%	90.30%	98.50%
EIH	94.80%	97.60%	94.00%	99.30%
AVRD	94.40%	97.20%	94.00%	99.30%
RDTE	95.30%	97.90%	92.50%	99.30%
NVCP	96.40%	97.80%	96.30%	99.30%
ADTPE	95.90%	97.40%	97.00%	99.30%
DPSP	94.40%	97.40%	97.00%	99.30%
AAEP	94.60%	97.40%	94.80%	99.30%
Full Features	94.40%	97.80%	94.00%	99.30%

operation, the proposed three skeletonization methods: ZSM, OPCA, and MOPCA, have higher accuracy of classification over ZS and OPTA in all six classifiers, in which the MOPCA has the highest average accuracy, which is 96.15% and 97.17% on the validation set and testing set respectively.

In addition, the denoising methods influence the performance of classification. For example, we can see that the average accuracy of MOPCA with ATFM+DCEM is 96.67% and 97.55% on the validation and testing sets, respectively, which is 2% higher than that of MOPCA with ATFM.

From the perspective of the classifiers, the decision tree and ensemble learning are the top two best classifiers in the task of five classes classification on all skeletons extracted by different methods, which have up to 98.5% and up to 98.3% accuracy on the validation set, respectively. For the testing set, both have up to 100% accuracy of classification. In contrast, the naïve Bayes has the worst performance in terms of accuracy, which has only

Table IV
CLASSIFICATION ACCURACY EVALUATION ON TRAIN/VALIDATION SET THAT HAS 5 DIFFERENT CLASSES

Classifier Models	ZS+ ATFM	OPTA+ ATFM	ZSM+ ATFM	OPCA+ ATFM	MOPCA+ ATFM	MOPCA+ATFM +DCEM
DT	88.4%	82.4%	93.0%	92.9%	97.4%	98.5%
NB	73.6%	73.8%	87.3%	87.5%	96.6%	95.7%
SVM	86.2%	78.5%	93.1%	87.3%	96.8%	97.8%
KNN	81.5%	71.8%	87.9%	83.6%	93.2%	94.4%
EL	90.7%	83.7%	95.5%	94.6%	97.3%	98.3%
MLP	85.4%	74.0%	88.4%	83.0%	95.6%	96.3%
Mean	84.3%	77.3%	90.8%	86.8%	96.1%	96.6%

Table V
CLASSIFICATION ACCURACY EVALUATION ON TEST SET THAT HAS 5 DIFFERENT CLASSES

Classifier Models	ZS+ ATFM	OPTA+ ATFM	ZSM+ ATFM	OPCA+ ATFM	MOPCA+ ATFM	MOPCA+ATFM +DCEM
DT	85.8%	82.1%	93.3%	94.0%	96.3%	100.0%
NB	79.9%	75.4%	86.6%	82.8%	95.8%	94.0%
SVM	90.3%	78.4%	91.8%	86.6%	97.0%	99.3%
KNN	85.1%	75.4%	86.6%	82.8%	97.8%	94.0%
EL	90.3%	82.8%	93.3%	94.0%	98.3%	100.0%
MLP	85.1%	68.7%	82.8%	88.1%	97.8%	98.0%
Mean	86.0%	77.1%	89.0%	88.0%	97.1%	97.5%

95.7% in the validation set and 94.00% in the testing set. Regarding training time, when skeletonization is set as MOPCA+ATFM+DCEM, the average time consumed by the decision tree is about 0.6s, which is faster than ensemble learning, which consumes about 4.2s. In Fig. 5, training Time consumed by different classifiers in 5 classes is presented.

For five classes classification task, the best combination method is using MOPCA skeletonization to extract the skeleton, using ATFM and DCEM to offset the noise's influence, and selecting decision tree to predict the class of the static hand gesture. The overall accuracy can reach 98.5%, and the train time is 0.6435s.

B. Performance evaluation of static hand gesture Classification for 10 classes

Similar to the previous section, the experiment of the feature selection has been conducted based on the classifiers of KNN and SVM once more. The only difference is that the current experiment considered more classes, which increased from 5 to 10. Experimental results of the feature selection are shown in Tab. VI.

By comparing Tab. VI and Tab. III, it is notable that the overall accuracy of classification is significantly reduced with the increasing number of classes since there are more complicated hand gestures are considered. In addition, the importance of each feature is also altered. For example, in Tab. III, we knew that the deletion of the NEP and NCP might significantly worsen the accuracy; however, in Tab. VI, the degree of the influence caused by them is much slightly when compared with the feature of AAEP. On the other hand, removing the NVCP and

Table VI
CLASSIFICATION ACCURACY COMPARISON FOR 8 AND 9 FEATURES AND 10 CLASSES

Feature Deleted	Classification Accuracy			
	Train/Validation Set		Test Set	
	KNN	SVM	KNN	SVM
NEP	81.30%	79.70%	84.00%	80.60%
NCP	81.80%	81.10%	84.80%	81.00%
EIH	82.60%	81.50%	84.40%	81.60%
AVRD	82.50%	80.50%	84.40%	79.70%
RDTE	84.50%	80.90%	85.70%	82.70%
NVCP	84.40%	80.90%	88.20%	82.70%
ADTPE	82.50%	78.70%	83.10%	79.30%
DPSP	83.80%	81.20%	85.20%	81.90%
AAEP	73.20%	74.40%	73.40%	78.10%
Full Features	82.70%	81.60%	84.40%	81.00%

RDTE may increase the classification accuracy on both KNN and SVM.

Similar to the previous section, the influence of the different skeletonization methods and the different classifiers on the accuracy are explored in static hand gesture classification. The accuracy of 10 classes on the validation and testing sets are presented in Tab. VII and Tab. VIII, respectively. Training time consumed by different classifiers on ten classes is shown in Fig. 6.

From Tab. VII and Tab. VIII, we can see that the average accuracy of classification based on skeletons extracted by distinct methods are all decreased to some extent when comparing with the results in Tab. IV and Tab. V. However, the MOPCA+DCEM+ATFM method still outperforms other methods. For example, for classifying ten types of static hand gesture tasks, the average accuracy of classification of the MOPCA+DCEM+ATFM

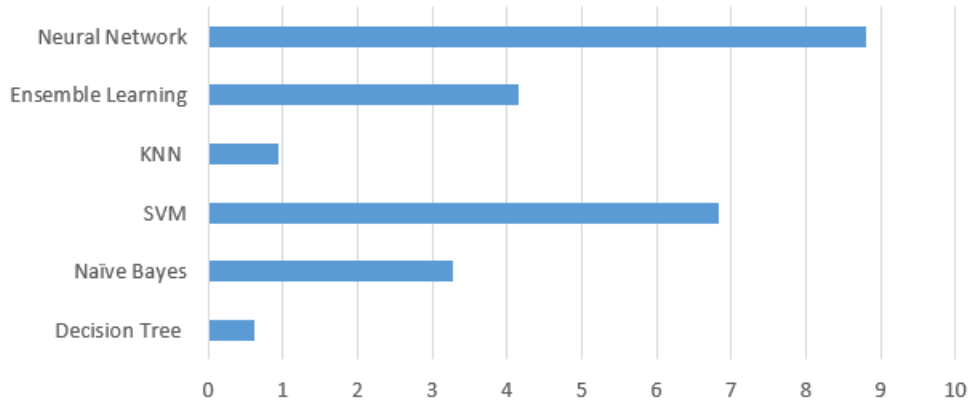


Figure 5. Training Time Consumed by 6 Classifiers on 5 classes dataset.

Table VII
CLASSIFICATION ACCURACY EVALUATION ON TRAIN/VALIDATION SET THAT HAS 10 DIFFERENT CLASSES

Classifier Models	ZS+ ATFM	OPTA+ ATFM	ZSM+ ATFM	OPCA+ ATFM	MOPCA+ ATFM	MOPCA+ATFM +DCEM
DT	76.5%	70.6%	84.4%	83.3%	86.7%	91.1%
NB	62.1%	61.8%	82.0%	76.1%	77.4%	82.9%
SVM	74.1%	59.5%	79.2%	77.4%	80.4%	81.6%
KNN	74.2%	63.9%	76.7%	75.5%	83.8%	82.7%
EL	80.9%	76.7%	88.7%	88.3%	90.0%	92.9%
MLP	73.4%	59.1%	74.9%	73.0%	82.6%	82.3%
Mean	73.5%	65.2%	80.9%	78.9%	83.4%	85.5%

Table VIII
CLASSIFICATION ACCURACY EVALUATION ON TEST SET THAT HAS 10 DIFFERENT CLASSES

Classifier Models	ZS+ ATFM	OPTA+ ATFM	ZSM+ ATFM	OPCA+ ATFM	MOPCA+ ATFM	MOPCA+ATFM +DCEM
DT	75.9%	73.8%	82.7%	87.3%	86.1%	91.1%
NB	59.5%	62.0%	86.5%	81.4%	80.2%	77.2%
SVM	73.8%	62.4%	77.6%	75.5%	82.3%	81.0%
KNN	69.6%	63.7%	78.9%	75.9%	87.8%	84.4%
EL	81.4%	79.3%	89.0%	91.6%	87.8%	92.8%
MLP	70.5%	60.8%	79.7%	76.8%	83.1%	83.5%
Mean	71.7%	67.0%	82.4%	81.4%	84.5%	85.0%

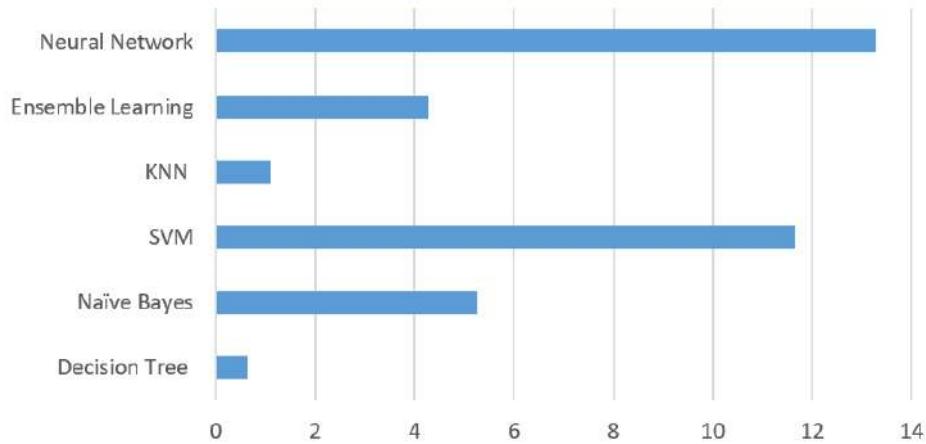


Figure 6. Training Time Consumed by 6 Classifiers on 10 classes dataset.

method can reach 85.58% on the validation set and 85.00% on the testing set. On the other hand, the average accuracy of classification on skeletons extracted by ZSM and OPCA is higher than the average accuracy of classification on skeletons extracted by ZS and OPTA.

In addition, from the perspective of the classifiers, the accuracy of the classification of the decision tree and ensemble learning surpassed all the other classifiers. Based on the skeleton extracted by the MOPCA+DCEM+ATFM method, the decision tree and ensemble learning classifier can obtain 91.10% and 92.80% accuracy on the testing set, respectively. In contrast, the classification accuracy of other classifiers can only achieve about 85%. Although the ensemble learning classifier has a slight bit advantage over the decision tree regarding classification accuracy, the time spent on training the ensemble learning is much more than the decision tree.

In a word, for ten classes classification task, the best combination method is using MOPCA skeletonization to extract the skeleton, using ATFM and DCEM to offset the noise's influence, and using ensemble learning to predict the class of the static hand gesture. The overall accuracy can reach 91.1%, and the train time is 0.6134s.

V. CONCLUSION

The hand gesture recognition based on the new image skeletonization methods, extracted gesture feature vector and using machine learning technique allow us to increase the classification accuracy. For 5 classes and 10 classes hand gesture classification task, the improvement of accuracy on test set is within the range of 0.4% to 20.4% , and that of 5% to 18% . The MOPCA+ADFM+DCEM method is effective in terms of average classification accuracy on test set. It achieves 97.5% on 5 classes recognition task and 85.00% on 10 classes recognition task. In addition, for 5 classes recognition task and 10 classed recognition task , the training time consumed by six classifiers is within the range of 0.7s to 8.9s and that of the 0.3s to 11s, respectively. It is set that ensemble learning model is the best classifier and it allows us to achieve 100% (5 classes) and 92.8% (10 classes) on test set. Increasing the accuracy of hand gesture classification based on the proposed skeletonization methods improves the technical characteristics of intelligent systems using video interfaces for entering commands and data, and makes a significant contribution to the development of semantic technologies for designing such systems.

REFERENCES

- [1] S. Sharma and S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," *Expert Systems with Applications*, vol. 182, p. 115657, 2021.
- [2] J. Zeng, Y. Sun, and F. Wang, "A natural hand gesture system for intelligent human-computer interaction and medical assistance," in *2012 Third Global Congress on Intelligent Systems*. IEEE, 2012, pp. 382–385.
- [3] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlitz, D. Wollherr, L. Van Gool, and M. Buss, "Real-time 3d hand gesture interaction with a robot for understanding directions from humans," in *2011 Ro-Man*. IEEE, 2011, pp. 357–362.
- [4] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: a review of techniques," *Journal of Imaging*, vol. 6, no. 8, p. 73, 2020.
- [5] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *Ieee transactions on systems, man, and cybernetics, part c (applications and reviews)*, vol. 38, no. 4, pp. 461–482, 2008.
- [6] J. J. LaViola Jr, "A survey of hand posture and gesture recognition techniques and technology," 1999.
- [7] G. Murthy and R. Jadon, "A review of vision based hand gestures recognition," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 405–410, 2009.
- [8] H. Kaur and J. Rani, "A review: Study of various techniques of hand gesture recognition," in *2016 IEEE 1st international conference on power electronics, intelligent control and energy systems (ICPEICES)*. IEEE, 2016, pp. 1–5.
- [9] D. Konstantinidis, K. Dimitropoulos, and P. Daras, "Sign language recognition based on hand and body skeletal data," in *2018-3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*. IEEE, 2018, pp. 1–4.
- [10] C. Xi, J. Chen, C. Zhao, Q. Pei, and L. Liu, "Real-time hand tracking using kinect," in *Proceedings of the 2nd International Conference on Digital Signal Processing*, 2018, pp. 37–42.
- [11] G. Devineau, F. Moutarde, W. Xi, and J. Yang, "Deep learning for hand gesture recognition on skeletal data," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 106–113.
- [12] F. Jiang, S. Wu, G. Yang, D. Zhao, and S. Kung, "Independent hand gesture recognition with kinect," *Signal, Image and Video Processing*, vol. 8, pp. 163–172, 2014.
- [13] F. Y. Shih and W.-T. Wong, "Fully parallel thinning with tolerance to boundary noise," *Pattern Recognition*, vol. 27, no. 12, pp. 1677–1695, 1994.
- [14] J. Feldman and M. Singh, "Bayesian estimation of the shape skeleton," *Proceedings of the National Academy of Sciences*, vol. 103, no. 47, pp. 18 014–18 019, 2006.
- [15] F. Gao, G. Wei, S. Xin, S. Gao, and Y. Zhou, "2d skeleton extraction based on heat equation," *Computers and Graphics*, vol. 74, pp. 99–108, 2018.
- [16] C. Yang, B. Indurkha, J. See, and M. Grzegorzczek, "Towards automatic skeleton extraction with skeleton grafting," *IEEE transactions on visualization and computer graphics*, vol. 27, no. 12, pp. 4520–4532, 2020.
- [17] W. Shen, X. Bai, R. Hu, H. Wang, and L. J. Latecki, "Skeleton growing and pruning with bending potential ratio," *Pattern Recognition*, vol. 44, no. 2, pp. 196–209, 2011.
- [18] A. S. Montero and J. Lang, "Skeleton pruning by contour approximation and the integer medial axis transform," *Computers and Graphics*, vol. 36, no. 5, pp. 477–487, 2012.
- [19] L. Serino and G. S. di Baja, "A new strategy for skeleton pruning," *Pattern Recognition Letters*, vol. 76, pp. 41–48, 2016.
- [20] M. E. Hoffman and E. K. Wong, "Scale-space approach to image thinning using the most prominent ridge-line in the image pyramid data structure," in *Document Recognition V*, vol. 3305. SPIE, 1998, pp. 242–252.
- [21] J. Cai, "Robust filtering-based thinning algorithm for pattern recognition," *The Computer Journal*, vol. 55, no. 7, pp. 887–896, 2012.
- [22] R. T. Chin, H.-K. Wan, D. Stover, and R. Iverson, "A one-pass thinning algorithm and its parallel implementation," *Computer Vision, Graphics, and Image Processing*, vol. 40, no. 1, pp. 30–40, 1987.
- [23] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.
- [24] J. Ma, T. V. Yurevich, and V. K. Kanapelka, "Image skeletonization based on combination of one- and two-sub-iterations models(in russ.)," *Informatics*, vol. 17, no. 2, pp. 25–35, 2020.

- [25] J. Ma, X.-H. Ren, T. V. Yurevich, and V. K. Kanapelka, "A novel sub-iterative parallel skeletonization method," *Journal of Computers (Taiwan)*, vol. 32, no. 6, pp. 83–97, 2021.
- [26] J. Ma, X. Ren, V. Y. Tsviatkou, and V. K. Kanapelka, "A novel fully parallel skeletonization algorithm," *Pattern Analysis and Applications*, pp. 1–20, 2022.
- [27] J. Ma, X. Ren, V. K. Kanapelka, and V. Y. Tsviatkou, "An automatic pruning method for skeleton images," in *Proceedings of the 15th International Conference, 2021*. United Institute of Informatics Problems of the National Academy of Sciences of Belarus, 2021, pp. 232–235.
- [28] X. Bai, L. J. Latecki, and W.-Y. Liu, "Skeleton pruning by contour partitioning with discrete curve evolution," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 449–462, 2007.
- [29] J. Ma, X. Ren, H. Li, W. Li, V. Y. Tsviatkou, and A. A. Boriskevich, "Noise-against skeleton extraction framework and application on hand gesture recognition," *IEEE Access*, vol. 11, pp. 9547–9559, 2023.
- [30] H. Chatbri and K. Kameyama, "Using scale space filtering to make thinning algorithms robust against noise in sketch images," *Pattern Recognition Letters*, vol. 42, pp. 1–10, 2014.
- [31] O. Z. Maimon and L. Rokach, *Data mining with decision trees: theory and applications*. World scientific, 2014, vol. 81.
- [32] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [33] D. J. Hand and K. Yu, "Idiot's bayes—not so stupid after all?" *International statistical review*, vol. 69, no. 3, pp. 385–398, 2001.
- [34] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.
- [35] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.
- [36] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

Распознавание жестов рук на основе свойств скелетизированных изображений

Ма Ц., Цветков В. Ю., Борискевич А. А.

Распознавание жестов рук является важной задачей и может использоваться во многих практических приложениях. В интеллектуальных системах распознавание жестов рук может использоваться для ввода информации посредством видеоинтерфейса. В настоящее время распознавание жестов рук на основе скелета стало популярной темой исследований. Существующие методы имеют низкую дискриминационную способность из-за чувствительности признаков к шуму изображения. Мы предложили новые методы уменьшения влияния шума на выделение признаков изображения руки. Разработан новый метод распознавания жестов рук, основанный на свойствах скелетизированных изображений. Цель исследования состоит в повышении точности классификации жестов рук. Данный подход позволяет повысить точность классификации с 5% до 21% по сравнению с существующими известными методами.

Received 12.03.2023

Automation of Educational Activities within the OSTIS Ecosystem

Alena Kazlova and Aliaksandr Halavaty
Belarusian State University
Minsk, Belarus
Email: kozlova@bsu.by,
alex.halavaty@gmail.com

Natalia Grakova
Belarusian State University
of Informatics and Radioelectronics
Minsk, Belarus
Email: grakova@bsuir.by

Abstract—An analysis of the need for a comprehensive restructuring of the education system, taking into account the requirements of the digital economy, is presented. The ways of solving some problems of the implementation of the educational process at the level of general school education are determined. A semantic approach to building a complex of intelligent learning subsystems including teaching, assisting learning and analytical to accompany the learning process ones within the framework of the OSTIS ecosystem is proposed.

Keywords—intelligent learning systems, semantics, OSTIS ecosystem, knowledge processing.

I. INTRODUCTION

In the context of the transition to the information society and the comprehensive digitalization of all areas of human activity at its various levels, highly qualified personnel are of the greatest value. The volume and level of requirements for the presentation and use of information in all spheres of life is increasing, which entails the inevitable active involvement of professionals in the process of continuous education. A modern person in the information society must be able to adapt to rapidly changing information flows. The formation of such skills is the main task of every educational institution, including universities, which in modern conditions are subject to increasingly stringent requirements. This applies to both the level of teaching and the level of organization of educational activities. Today, the organization of educational activities in schools and in secondary specialized vocational, higher educational institutions largely determines the level of development of the state. Therefore, we can fully explain the great interest in the use of information technology in order to increase the efficiency of this activity. However, despite the rather active research carried out in this direction, it is too early to say that the use of information technology has significantly increased the effectiveness of educational activities. There are many reasons for this. Among them there are both objective technical, methodological reasons, as well as reasons of a purely organizational, administrative nature. These reasons include:

- lack of a systematic approach to the selection of the main objects and processes for automating the activities of educational institutions, including higher ones;
- a small number of viable technologies for the integrated development, implementation, operation, maintenance and evolution of educational automation tools;
- natural social resistance, conservatism that impedes the comprehensive automation of the educational process.

II. CURRENT APPROACHES TO THE EDUCATIONAL ACTIVITIES AUTOMATION

Automation of educational activities requires an integrated approach, taking into account the peculiarities of educational work at all stages of education, from elementary school to graduation from the magistracy, and possibly further, when obtaining higher qualifications in graduate school. This approach certainly requires the use of existing ones, as well as further development and widespread use of methods and tools of artificial intelligence, and related disciplines. Work in this direction has been carried out by various groups of researchers for more than 30 years. Research is being carried out on the theory and methodology of distance education, new approaches to the development of distance and open education are proposed, based on the ideas of organizational design and reengineering of organizations, methods of knowledge engineering and the theory of agents, models of multi-agent systems and virtual organizations [1]–[7].

A number of authors identified and studied the main classes of systems and technologies needed to create virtual departments of universities and universities in general. Particular attention of researchers is paid to the problems of symbiosis of network and intelligent technologies, for example, models of intelligent learning systems based on multi-agent technologies [8], [9].

One of the current trends in the development of applied intelligent systems (IS) is the implementation of ISs that can not only solve problems from the relevant

subject area, but also train the user in knowledge and skills from this subject area. At the same time, when developing modern computer learning systems (CLS), it became necessary to use methods and tools of artificial intelligence, which led to the emergence of a new class of CLS - intelligent learning systems. At present, due to the growing requirements for systems of this class, the problem of developing The intelligent learning systems have to become relevant, which can be characterized by the following features they should provide:

- 1) processing large volumes of complexly structured information of various types;
- 2) flexibility and easy modifiability of the system;
- 3) integration of various models and mechanisms for solving problems;
- 4) support for various models of learning and user interaction management;
- 5) integration of various software systems within one system and management of their operation and interaction;
- 6) wide use of multimedia tools;
- 7) work in real time.

Currently, there are CLSs in which these problems are solved with the elaboration of only some individual issues. This is due to the fact that it is not easy to solve all the problems in a complex by the currently existing information and intellectual means. The peculiarity of the implementation of the learning process in ILS is that, in addition to representing and processing knowledge about the subject area, the system must contain information about its users, be able to process it and thus adapt to the individual characteristics of each specific user [10]. In addition, one of the most important issues in the design of a learning system is the management of a dialogue with a user (usually untrained in computer technology). User interaction, in contrast to the interaction of subsystems in a computer system, is a more complex process, since it contains an element of unpredictability.

Consider the features inherent in the initial stage of education - general secondary education. School education is an initial, but very important stage that shapes the entire further development of an individual's education. If schooling is too low, then no further steps will fix it. It is impossible to get a good engineer out of an illiterate, unprepared person. If there is no knowledge base, then it is impossible to build a knowledge base of a higher level. Therefore, we will begin our consideration of the issue of education with school education, especially since many problems and issues that arise in the field of education at higher levels coincide with the problems of school education, or are their consequences. The purpose of education is not only and not so much to expand the horizons and knowledge of the student, the main task is to help him decide on the choice of a field of activity according to his abilities and the needs of society.

An integrated approach to the formation of a strategy for the intellectualization and digitalization of this stage of education requires an analysis of all aspects of education - academic disciplines, pedagogical technologies and methods, psychological characteristics of children of different ages, from primary school students to graduates, organizational features of the educational process, technical equipment of educational institutions, the capabilities of telecommunication networks etc. Such an approach should provide a deep modernization of school education based on the use of artificial intelligence technologies, data and systems analysis, multi-agent technologies, synergistic approaches, modeling, ontologies and semantic technologies. The latter play a special role in the formation of student knowledge at all stages of education. It is imperative to take into account the individuality of the student when choosing pedagogical methods and technologies, which will make it possible to find the right approach to the effective teaching of students with varying degrees of preparedness for the perception of certain disciplines [11].

The main task of the learning intellectual system is to explain, teach and continuously unobtrusively control the process of human learning. Here, a special role is played by semantic intelligent systems that use semantic relationships. Semantic intelligent systems allow, on the one hand, to speed up the process of obtaining knowledge based on obtaining instant access to a huge information field, and on the other hand, intelligent systems should help choose the best learning path, the path along which complete knowledge will be obtained in the shortest possible time.

The socialized characteristics of the quality and quantity of knowledge of graduates of various educational institutions include:

- competencies – a combination of knowledge, skills and experience necessary for the high-quality performance of tasks;
- breadth of knowledge – a set of knowledge from various fields that can complement each other and form a single picture of the world around;
- depth of knowledge – a characteristic showing the extent and the level of complexity of a person's knowledge on a particular issue or phenomenon. For a correct construction of the educational process at a particular level, it is necessary to specifically outline the set of knowledge that a graduate must possess in a particular academic subject;
- stage of education - an independent completed stage of training and education of the education system;
- academic subject - a system of knowledge, skills and abilities selected from a specific branch of human activity.

III. ORGANIZATION OF THE EDUCATIONAL PROCESS

An important task in organizing an effective educational process is to fulfill the requirement that the curricula of each year of study in subjects complement and expand, and not duplicate the knowledge gained in previous years of study. Curricula in subjects and the learning process itself must comply with certain principles (rules, requirements), which will be discussed below.

When compiling curricula, it is necessary to take into account the relationship between various academic disciplines. Knowledge gained in one academic discipline is used in the study of other disciplines. At the same time, the use of knowledge gained in other subjects allows one to consolidate this knowledge. For example, the use of knowledge about trigonometric functions, projections, vectors in solving physical problems allows one to consolidate, deepen and substantiate this knowledge obtained from mathematics.

Consideration of new material, its development in the framework of solving problems, should contribute to the repetition of previously covered material both in this and other related subjects. At the initial stage, tasks are given directly on this material (for example, on the application of a certain formula). For real knowledge of the topic, one must be able to solve problems, in the solution of which it is necessary to use, along with the material of the topic being studied, the knowledge obtained earlier in the study of other topics within this or other subjects.

Consideration of phenomena and laws studied in various sections of educational subjects (regardless of the time of study) at each stage must be complete and cannot be incomplete due to the fact that students do not have any preliminary data. It is also necessary to take into account the consistency and connectivity of the acquired knowledge so that knowledge does not turn into a set of fragmentary information and definitions that require banal memorization. Logical comprehension of the material, building connections of this material with the available knowledge and the surrounding reality are the main components of individual experience. Knowledge appears in the form of concepts and relations between them, as well as judgments and conclusions of the student derived from them. It is such knowledge in the form of skills and abilities that is best stored in the memory of the student. It is necessary to adjust the programs for studying disciplines for the timely use of knowledge in the study of other disciplines.

It is also necessary to consider the issue of filling the content of educational material on each topic in each individual discipline. It should be borne in mind that there are certain restrictions both in complexity and in the volume of new material that a student can perceive in the time allotted for this, in order to avoid overloads that may adversely affect his health. Topics that are displayed

in the physical and information space around us due to their vital relevance are mastered most effectively. Topics that do not find reflection in the environment and are not required to obtain the necessary life skills should be derived from the compulsory school curriculum material and given in the most compact form at the familiarization level. Satisfaction of educational programs and the very process of teaching these principles contributes to the fact that various disciplines will form comprehensive knowledge about the world around them.

When considering issues related to education, one cannot ignore the aspect related to the individuality of each student. Each student has abilities and predispositions for certain subjects. Taking into account these factors ensures the maximum possible disclosure of the creative potential of each person. The development of these abilities and the preparation of schoolchildren for the choice of professional activity in later life should be served by additional and optional education. At the level of such education, it is possible to implement a more personalized approach to each student, in which it becomes possible to fully reveal the creative potential of each. At the same time, it is necessary to give more extensive and in-depth knowledge in the chosen disciplines. Also, the issues of organizing secondary education must be closely linked with the organization of education at subsequent levels. The foundation of knowledge, their base formed at school is the starting point from which the next stage in the student's life begins.

Unfortunately, it should be noted that in recent years there has been a decline in the level of school education of graduates of secondary educational institutions. The introduction of new curricula does not correct, and sometimes exacerbates this situation. There are various subjective and objective reasons leading to this. Among the main ones are a sharp increase in the amount of information in various sections, caused by the development of science and an increase in the availability of information, the inertia and conservatism of school programs and educational technologies. One of the reasons is that programs and textbooks for different courses are compiled by different people who specialize in certain areas of knowledge. These people do not see the general picture of emerging knowledge, they try to fill their subject with as deep new data and definitions as possible, to increase the number of hours allotted for studying the subject at school. Sometimes such an increase in material leads to the fact that the school curriculum in some sections of the subject practically does not differ from the programs of higher educational institutions. According to the authors of these programs, this should lead to interest in their subject, "improvement" of the quality of knowledge on the subject. In fact, this only leads to the fact that students have to remember a lot more information (sometimes unrelated), which leads

to overload and confusion of students. No group of specialists will be able to fully take into account and calculate all the issues related to education, since these problems are multidirectional and multilevel. The traditional educational system cannot provide graduates with a timely and proper level of knowledge. To maintain a high level of demand in the labor market, students must rapidly update the necessary knowledge, the volume of which doubles on average every year and a half, which requires constant retraining. The problem can be solved by creating intelligent computer education systems based on very large knowledge bases. Such systems should make it possible to correct mistakes made in approaches to the development of knowledge, to take into account changes in the requirements for graduates' competencies that appear with the development of scientific and technical knowledge and the material and technical base of society more quickly and efficiently. Digital technologies will create an education system that will be more efficient and balanced.

What should an intelligent system be able to do and what knowledge base should it possess? In the intellectual system, a knowledge base should be formed, covering all the knowledge that a graduate should have at the end of school. The knowledge base should be multi-level. Otherwise, there will be no deep convergence between the preparation of students and the knowledge that they should have at the end of school. This part of the general knowledge base should cover all subjects of the school curriculum, but only to the extent sufficient to understand and assimilate information on each subject. Also, the knowledge base should take into account the problem of the amount of information, taking into account the complexity of the material that can be mastered in a particular period of study. Based on this knowledge, the intellectual system must form a training program, distribute during which period of study specific sections of various subjects are studied and at what level knowledge is formed during this period. At the same time, the intellectual system should take into account what knowledge the student has when he starts studying a new topic. By this time, the student should know all the necessary prerequisites, have knowledge in this and other subjects necessary to study the topic. As a result, a balanced distribution of all studied material from various subjects over time should be formed. Since the volume of the studied material, taking into account the complexity and the training time, have a finite value, the intellectual system must limit the amount of materials given for training in each period of time, and cut off materials that only increase the amount of memorized data, but do not carry any additional information necessary to understand and master the basic laws and phenomena. As a result, a model of the subject area should be created for each discipline, taking into

account interdisciplinary links. In this regard, it is very important to provide technological means of "transition" of boundaries between educational materials of different academic disciplines. The domain model plays a major role, since it is used to solve the problems of structuring and systematizing educational material, implementing navigation and search algorithms for educational material and implementing adaptive learning management, etc. In ideal case, the student should be able to work with educational material in solving a number of problems not on the scale of a single academic discipline, but on the scale of all disciplines related to the issue under study. It is necessary to mention the logical organization of educational material within the framework of the specialty, which allows you to identify the connections of academic disciplines, certain topics of these academic disciplines, their constituent fragments (theorems, definitions of concepts, etc.) with other academic disciplines, topics, fragments of educational material, subsequent and previous. It becomes possible to determine a more rational sequence for studying educational material. The knowledge management system located in the knowledge base should also ensure the collection and systematic organization, analysis of data and knowledge from various sources, replenishment of the knowledge base from within the system itself.

An intelligent subsystem with an open, self-supplementing database should also be created, which will grow over time as topics in various subjects are covered. An educational approach based on the gradual filling of the student with knowledge and skills through their gradual presentation within a set of disciplines remains relevant. At each stage of training, the student should have access to the level of knowledge in the database that corresponds to the material he has studied. learning based on these assessments, i.e. an intellectual system should itself consist of a number of subsystems containing knowledge bases semantically correlated with each other.

The individual activity of students is an important component of education, which forms the skills of the student, contributes to the most effective assimilation of knowledge, determines his inclinations and contributes to their development. Individual activity is present in various ways of studying information: training in a mandatory program, additional optional individual training. Additional types of education may use information that is not included in the mandatory school curriculum. Such broader information should be available to students, but it should be in the general knowledge base and used for additional, optional or independent education. The development of this information should take place outside the main school education, both in terms of time and information and program components, and contribute to the development of individual abilities of students.

In any type of learning with the participation of intelligent learning systems, the student needs an assistant - "personal assistant" = "personal manager", who will help establish communication with the intelligent system and make its use more efficient. Personal assistant" of each user participant is included in the system in order to ensure both the security of the system and the comfort of the user's work already at the stage of the first interaction of the user with the intelligent interface. An adaptive approach to designing a user interface in educational systems is one of the promising directions for the development of this class of systems. This approach provides for the creation of a flexible structure of the dialogue between the system and the user in accordance with a number of individual characteristics of the user: readiness to work with the system, characteristics of interaction with the system, interface preferences, individual psychological characteristics.

Since the development of the educational system is supposed to be organized on the basis of a common OSTIS system, it is natural to use common approaches when building separate subsystems. The developed general models of the interface of the OSTIS metasytem must be adapted for the educational OSTIS system. This will significantly reduce the material and time costs for building the system interface.

Along with the task of creating a personal user interface, a personal manager has many other tasks. It forms a spectrum of information about the learner. Determines his inclinations and abilities in a particular field of knowledge. When considering individual topics studied, a personal manager helps to form a variant of navigation through semantic links, when a student or a system forms a certain path, reveals a feature, moves to the next feature, and so on through the educational material; proposes tasks on laws from the studied material, at the next stage - tasks in which, along with this, material from previously covered topics, including from other subjects, is used, then tasks of an even more complex level are proposed. Also, a personal manager must form a system for evaluating results at each stage of training. This is a constant unobtrusive diagnostics of the state of the student and the adjustment of his learning model based on the analysis of the control results. It is formed, on the one hand, on the basis of information about how much time the student spent studying the topic, the tasks of what complexity he was able to solve, etc. On the other hand, a personal manager must form a "reasonable" question-answer system based on the specified material (i.e., a system capable of finding answers to a sufficiently large number of questions regarding the meaning and semantics of the relevant material).

IV. WAYS TO THE EDUCATIONAL ACTIVITIES AUTOMATION

The adjustment of the learning process should take place on the basis of these assessments, forming a set of necessary information for the next level, i.e. an intellectual educational system should itself consist of a number of subsystems containing knowledge bases semantically correlated with each other, specialized databases and knowledge, electronic textbooks, including those prepared using hypermedia and multimedia, as well as network sources based on the Internet. Thus, the requirements for the efficiency and practice orientation of training systems are constantly increasing, which leads to the inevitable realization of the relevance of the problem of developing such computer training systems, which should provide:

- processing of large volumes of complex structured information of various types;
- flexibility and easy modifiability of the system;
- integration of various models and mechanisms for solving problems;
- support for various models of learning and user interaction management;
- integration of various software systems within one system and management of their operation and interaction;
- widespread use of multimedia;
- operation in real time.

It is also possible to formulate the basic requirements for intelligent learning systems at different levels of education, as follows:

- Versatility. Availability of means of representation and processing of educational and educational-methodical knowledge, focused on any subject area;
- Adaptability. Availability of means of forming a model of the student and means of adapting the educational process to the student;
- Flexibility. The availability of tools that allow the implementation of various learning models, as well as supporting various forms of learning;
- Extensibility. The ability to modify existing system properties and add new properties without violating the conceptual integrity of the system;
- Distribution. Availability of means of organizing remote access to the system. Ability to work with the system from different locations (locally and remotely) at any time.

To implement such properties of educational intelligent semantic systems and meet the requirements for them, it is necessary to develop and build systems such as:

- The semantic model of the learner, which provides the personification of learning (adaptation) to the learner (give examples of specific ontologies);

- Semantic model of various pedagogical methods;
- Semantic model of general management of the process of individual learning.

An example of building such intelligent learning systems can be an intelligent learning system based on semantic electronic textbooks [11]. An electronic textbook, as a rule, is used for independent study of an academic discipline. As far as the knowledge presented in the electronic textbook is well structured, their completeness and consistency, clarity and accessibility, the possibility of quick associative search are ensured, the effectiveness of the learning process increases. Semantic electronic textbook - a set of software tools built using methods and tools of artificial intelligence, in particular, OSTIS technology, in which the knowledge base of educational and educational material is presented in the form of a hypertext semantic network and the possibility of associative access to any fragment of this educational material is provided. Such a textbook is based on a knowledge base, since that the electronic semantic textbook turns into a fairly "reasonable" question-answer system for the specified material. This means that the SEU becomes a system that "understands" the meaning of the educational and teaching materials contained in it.

An intellectual learning system should be able to track the consistency and integrity of the picture of the world presented in it and presented to the student, teach through its own ability to solve problems, contain a system of assessments and decision-making on a learning strategy based on these assessments, i.e. should itself consist of a number of subsystems containing knowledge bases semantically correlated with each other. The knowledge base of the intelligent educational system as a whole should also be semantically correlated across various disciplines (directions) of education. This can be represented by a system of OSTIS communities that provide educational activities in the field of AI within the overall OSTIS ecosystem as an association of structures, objects and their interaction with each other and with the external environment. The high connectivity of knowledge bases of the OSTIS ecosystem subsystems, a single technology for their organization and effective interaction, a variety of technologies used within the ecosystem, independence from the interface language, combined with the ability to include new subsystems built on the same principles, makes this approach a key success factor in analysis and synthesis of a comprehensive solution for the organization of educational activities at all levels of education in our society.

REFERENCES

- [1] V. F. Gornev To manufacturing engineering of the XXI century. *Vestnik MGTU. Machinostroenie*, 1999, No. 4, pp. 53–62
- [2] V. V. Emelyanov, V. V. Kureychik, V. M. Kureychik Teoria i praktika evolyutsionnogo modelirovaniya [Theory and practice of evolutionary modeling], Moskva, Fizmatlit, 2003, P. 432 (In Russ.)
- [3] Yu.M. Vishnyakov, S.I. Rodzin Virtual'ny universitet: mif ili real'nost'? [Virtual university: myth or reality?]. *Izvestiya TRTU*, 2000, No. 6, pp. 275–282 (In Russ.)
- [4] V.M. Kureychik, V.I. Pisarenko Dinamika sovremennogo obrazovatel'nogo prostranstva v kontekste sinergeticheskikh idey [Dynamics of modern educational space in the context of synergetic ideas]. *Trudy mezhdunarodnykh nauchno-tehnicheskikh konferentsiy Intellektual'nye sistemy (AIS'08) I Intellektual'nye SAPR (CAD'2008)*, Fizmatlit, 2008, Vol. 3, pp. 182–199 (In Russ.)
- [5] P. Senge Pyataya distsiplina. Iskusstvo I praktika obychnyushey organizatsii [Fifth discipline. Art And Practice Teaching Organization]. ZAO Olymp-bussiness, 1999 (In Russ.)
- [6] V. B. Tarasov, V. V. Yarnyh Intellektual'nye I samoobuchayushiesya organizatsii v sfere vysshego obrazovaniya [Intelligent AND Self-Learning Organizations in Higher Education]. *Otkrytoe obrazovanie*, 2006, No. 4, pp. 34–38 (In Russ.)
- [7] V. B. Tarasov, V. V. Golenkov Sinergeticheskie uchebnye organizatsii v sfere vysshego obrazovaniya [Synergistic educational organizations in the field of higher education]. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, Minsk, BGUIR, 2017, pp. 221–228 (In Russ.)
- [8] E. A. Lugovskaya, V. B. Tarasov Mnogoagentnye sistemy podderzhki otkrytogo obrazovaniya v tehnichestkom universitete [Multi-agent support systems for open education at a technical university]. *Programmnye produkty I sistemy [Software products and systems]*, 2001, No. 2, pp. 29–34. <https://cyberleninka.ru/article/n/mnogoagentnye-sistemy-podderzhki-otkrytogo-obrazovaniya-v-tehnicheskomi-universitete> (date of access: 01.04.2023). (In Russ.)
- [9] N. Guarino, P. Giaretta Ontologies and Knowledge Bases. Towards a Terminological Clarification. Towards Very Large Knowledge Bases. N.J.I. Mars (eds), Amsterdam: IOS Press, 1995.
- [10] Yu. A. Kravchenko Integratsiya svoistv kognitivnykh stiley I intellektual'nykh agentov rfr osnova sozdaniya adaptivnykh informatsionnykh obuchayushchikh sistem [Integration of the properties of cognitive styles AND intelligent agents rfr is the basis for the creation of adaptive information learning systems]. *Otkrytoe obrazovanie [Open Education]*, 2010, No. 4, S/20–28. <https://cyberleninka.ru/article/n/integratsiya-svoistv-kognitivnykh-stiley-i-intellektualnykh-agentov-kak-osnova-sozdaniya-adaptivnykh-i-nformatsionnykh-obuchayushchikh> (date of access: 01.04.2023). (In Russ.)
- [11] N. Gulyakina, A. Kazlova Semantically compatible ostis educational automative systems. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, Minsk, BGUIR, 2022, pp. 375–380.

Автоматизация образовательной деятельности в рамках экосистемы OSTIS

Козлова Е. И., Головатый А. И., Гракова Н. В.

Представлен анализ необходимости комплексной реструктуризации системы образования с учетом требований цифровой экономики. Определены пути решения некоторых проблем реализации образовательного процесса на уровне общего школьного образования. Предложен семантический подход к построению комплекса интеллектуальных обучающих подсистем, включающих обучающую, вспомогательную обучающую и аналитическую, сопровождающую учебный процесс, в рамках Экосистемы OSTIS.

Received 13.03.2023

Building a Comprehensive Digital Archive for the Polotsk Cadet Corps: A Case Study Intelligent Information and Reference System Design

Arkadzi Oskin

Euphrosyne Polotskaya State University of Polotsk

Polotsk, Belarus

Email: a.oskin@psu.by

Abstract—On the example of designing and building an Intelligent Information Retrieval System (IIRS) "Polotsk Cadet Corps", the main stages of creating an IIRS are described. The principles of ontological modeling and the tools used for these purposes are briefly considered. Based on the analysis of the subject area, an ontological model is built, which can be transformed into a working IIRS by using the OSTIS technology stack. The advantages of OSTIS technologies for solving problems of this kind are described.

Keywords—OSTIS Technology, ontological modeling, reference system, intelligent system

I. INTRODUCTION

The Polotsk Cadet Corps was opened on July 8 (June 25, old style) 1835 by order of Emperor Nicholas I. The building was located in the building of the former Jesuit Academy. Currently, this historic building houses the Faculty of Information Technology and the Faculty of Humanities of Euphrosyne Polotskaya State University of Polotsk, which explains our interest in the history of the Polotsk Cadet Corps.

In order to create an Intelligent Information and Reference System, we built an ontological model of the subject area. Ontological modeling is a method of modeling information systems that focuses on the relationships between concepts and entities. It is a branch of artificial intelligence that is concerned with representing knowledge in a way that is intuitive and easy to understand. Ontological modeling is used in a variety of applications, including information retrieval, natural language processing, and semantic web technologies. In the context of intellectual information systems, ontological modeling is used to represent knowledge in a way that can be easily understood by humans and machines alike.

There are several steps involved in ontological modeling in Intellectual Information Systems:

- 1) Define the domain: Identify the domain of the system, including the concepts and entities that

are relevant to it. This involves understanding the context of the system and its intended use.

- 2) Create a conceptual model: Develop a conceptual model of the domain, which includes the concepts and relationships between them. This can be done using a variety of modeling techniques, such as UML, ER diagrams, or other graphical notations.
- 3) Define the ontology: Create an ontology that formalizes the conceptual model, using a language such as OWL or RDF. This involves specifying the relationships between concepts in a way that can be understood by both humans and machines.
- 4) Populate the ontology: Populate the ontology with data, using tools such as Protégé or other ontology editors. This involves creating instances of concepts and linking them together in a way that reflects the relationships defined in the ontology.
- 5) Use the ontology: Once the ontology has been populated, it can be used to support a variety of applications, such as information retrieval, natural language processing, or semantic web technologies.

Ontological modeling can be a complex process, but it is a powerful tool for representing knowledge in information systems. By using ontological modeling, it is possible to create systems that are more intuitive and easier to understand, both for humans and machines.

There are several advantages of using ontological modeling in intellectual information systems, including:

- 1) Automated reasoning: By having the essential relationships between concepts built into them, ontologies enable automated reasoning about data. This makes it easy to implement semantic graph databases that use ontologies as their semantic schemata.
- 2) Increased use, reuse, and maintainability: Ontologies facilitate domain knowledge sharing, with a common vocabulary across independent software

applications. This leads to increased use, reuse, and maintainability of the information systems.

- 3) Easy to relate to object-oriented programming paradigm or database design: Those who are familiar with the object-oriented programming paradigm or database design can easily relate the ontological representation of the domain entities to classes or database schemas. The classes are generic representations of the entities that encapsulate properties and behaviors.
- 4) Semantic model of data: One major advantage of using a domain ontology is its ability to define a semantic model of the data combined with the associated domain knowledge. Ontologies can also be used to define links between different types of semantic knowledge. Thus, ontologies can be used in formulating some data searching strategies.

Overall, ontological modeling in intellectual information systems provides a powerful tool for representing knowledge in a way that is intuitive and easy to understand, both for humans and machines.

Ontology editors are applications designed to assist in the creation or manipulation of ontologies. They use one or more ontology languages to create, visualize, and manipulate ontologies. These editors have features such as visual navigation possibilities within the knowledge model, inference engines and information extraction, support for modules, the import and export of foreign knowledge representation languages for ontology matching, and support of meta-ontologies such as OWL-S and Dublin Core. Additionally, there are various tools used for ontological modeling of intelligent information systems. Let's consider the most popular of ontology editors.

II. ONTOLOGY EDITORS

To create ontologies, specialized software products are widely used - ontology editors. Let's look at the most popular of them. We will compare the characteristics of the following software products.

There are several popular ontology editors that can help authors create their ontologies. Some of them provide additional functions and plugins that can be useful when working with ontologies. Some of the most popular ontology editors are listed below:

- 1) NeOn Toolkit is an ontology editor with many plug-ins available, especially suitable for large scale projects (eg multi-module ontologies, multilingual, ontology integration, etc.).
- 2) Neologism is an online dictionary editor and publishing platform.
- 3) Vitro is an integrated editor for ontologies and semantic web applications.
- 4) Knoodl is a community-oriented ontology and knowledge base editor.

- 5) Fluent Editor is a comprehensive tool for editing and manipulating complex ontologies that uses controlled natural language. It provides an instant natural language representation of OWL/SWRL, which improves performance and makes editable ontologies easier to read and understand.

Some other popular ontology editors that can be mentioned include:

- 1) Eddy
- 2) OntoME
- 3) OntoStudio (formerly known as OntoEdit)
- 4) Protégé

Let's take a closer look at some of the editors from the list above.

NeOn Toolkit.

NeOn Toolkit is an ontology editor that allows users to create and edit ontologies. It offers a variety of tools and features to support the development of ontologies, including visual modeling, ontology debugging, and ontology testing. Some of the advantages of using NeOn Toolkit are:

- User-friendly interface: NeOn Toolkit has an intuitive interface that makes it easy to create and edit ontologies.
- Visual modeling: The tool offers a visual modeling environment that allows users to create and edit ontologies using graphical representations.
- Collaboration: NeOn Toolkit supports collaboration between users, making it easy to work on ontologies as a team.
- Ontology debugging: The tool provides debugging capabilities that help users identify and fix errors in their ontologies.
- Ontology testing: NeOn Toolkit includes a testing framework that allows users to test their ontologies to ensure they are working correctly.
- Ontology testing: NeOn Toolkit includes a testing framework that allows users to test their ontologies to ensure they are working correctly.

Despite its advantages, NeOn Toolkit also has some drawbacks. For example:

- Steep learning curve: While NeOn Toolkit is user-friendly, it can still be challenging to learn for users who are new to ontology development.
- Limited documentation: Some users have reported that the documentation for NeOn Toolkit is limited, making it difficult to troubleshoot issues or learn about advanced features.
- Limited support: NeOn Toolkit is an open-source tool, which means that support is limited to user forums and community resources.

NeOn Toolkit is primarily used in the field of semantic web development, specifically in the development of ontologies. It is commonly used in research and academic

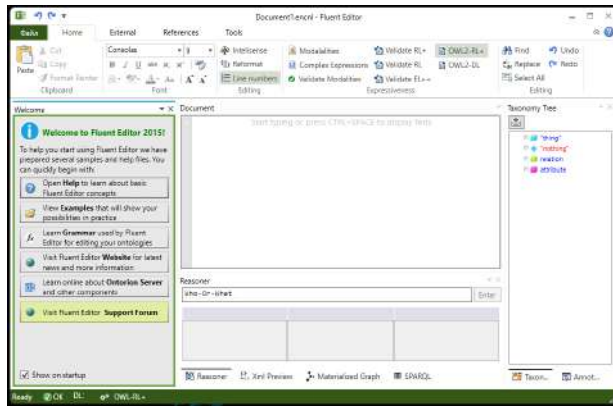


Figure 1. Fluent Editor.First start.

settings for ontology development, but it can also be used in industry settings for knowledge management and data integration.

Fluent Editor.

Fluent Editor is an ontology editor developed by the Polish company Cognitum. It is used for editing complex ontologies created using a controlled natural language (CNL). The editor allows users to create ontologies by entering phrases in natural language.

Some of the advantages of Fluent Editor are:

- User-friendly interface: Fluent Editor has a simple and intuitive graphical user interface that allows even non-experts to create and edit ontologies.
- Support for natural language: Fluent Editor supports natural language input, which makes it easier to create ontologies for non-experts.
- Advanced features: Fluent Editor includes advanced features such as automated reasoning, which can help identify inconsistencies and errors in ontologies.

However, there are also some limitations to Fluent Editor:

- Limited documentation: Fluent Editor has limited documentation available, which can make it difficult for users to learn how to use the editor.
- Limited support: Cognitum provides limited support for Fluent Editor, which can make it difficult for users to get help if they encounter problems.
- Limited customization: Fluent Editor has limited support for customization, which can make it difficult for users to tailor the editor to their specific needs.

Fluent Editor is primarily used for creating and editing complex ontologies [0][2][4]. It can be used in various domains, including healthcare, finance, and engineering, where ontologies are used to represent and organize knowledge.

Eddy.

Eddy is an ontology editor that allows users to create and edit ontologies. Ontologies are used to define concepts and relationships in a specific domain of knowledge. Eddy is a web-based application that can be used by anyone with an internet connection. It has several advantages, including:

- User-friendly interface: Eddy has a simple and intuitive interface that makes it easy for users to create and edit ontologies.
- Collaboration: Eddy allows multiple users to work on the same ontology simultaneously, making it ideal for collaborative projects.
- Integration: Eddy can be integrated with other tools and applications, making it a versatile tool for ontology development.

However, Eddy also has some limitations:

- Limited functionality: Eddy's functionality is limited compared to other ontology editors.
- Learning curve: Although Eddy is user-friendly, there is still a learning curve for users who are new to ontology editing.
- Lack of support: Eddy does not have a large community of users, which means that there is limited support available for users who encounter problems.

Eddy is primarily used in the field of knowledge engineering, which involves the creation and management of knowledge-based systems. It can be used in a variety of domains, including:

- Healthcare: Eddy can be used to create ontologies for medical terminology and patient data.
- E-commerce: Eddy can be used to create ontologies for product catalogs and online marketplaces.
- Education: Eddy can be used to create ontologies for educational resources and curriculum development.

Overall, Eddy is a useful tool for ontology editing, but it may not be the best option for all users depending on their specific needs and requirements.

OntoME.

OntoME is an ontology editor that enables users to create, edit and publish ontologies. Here are some advantages and disadvantages of ontoME:

Advantages:

- OntoME provides a user-friendly interface for creating and editing ontologies, allowing users to create ontologies without programming knowledge.
- It supports multiple ontology formats, such as OWL, RDF, and RDFS.
- It offers a range of features, such as the ability to import and export ontologies, search for terms, and visualize ontologies.
- OntoME allows for collaboration and sharing of ontologies through a web-based interface.
- It provides support for versioning and change management of ontologies.

Disadvantages:

- OntoME may not be suitable for heavy-weight projects, such as multi-modular ontologies or ontology integration, as it does not offer advanced features for such projects.
- It may not be suitable for users with advanced programming knowledge, as it does not allow for direct editing of the ontology code.
- The web-based interface may have limitations in terms of performance and speed compared to desktop-based ontology editors.

The scope of OntoME is to provide a user-friendly and accessible ontology editor for users with varying levels of ontology development expertise. It is suitable for small to medium-sized ontology projects with moderate complexity. For more complex projects, users may need to consider other ontology editors that offer advanced features and customization options.

OntoStudio.

OntoStudio is an ontology editor that provides a professional environment for ontology development. It supports W3C standards such as OWL, RDF, and RDFS, and F-Logic for the logic-based processing of rules. OntoStudio also comes with many connectors to databases, documents, file systems, applications, and web services.

The modular design of OntoStudio enables users to enrich it with self-developed modules and customize it according to their personal needs. It has modeling tools for ontologies and rules, as well as components for the integration of heterogeneous data sources. OntoStudio is available with a free evaluation license.

In a study that compared five ontology editors, including OntoStudio, the main criterion for comparison was the convenience for users. The study described the basic features and structure of the editors, as well as their way of use. OntoStudio was found to be a convenient tool for users.

OntoStudio's advantages are its support for various ontology languages and its modular design that allows customization. Its disadvantages are not mentioned in the sources. OntoStudio's scope is in the development and operation of semantic applications that involve ontology learning, reasoning, and text mining. It is also useful for the storage and management of semantic data and metadata.

Specific features of OntoStudio's modeling tools for ontologies and rules are not explicitly mentioned in the sources. However, OntoStudio combines modeling tools for ontologies and rules and is built on top of a powerful internal ontology model that allows the user to edit a hierarchy of concepts or classes. OntoStudio also supports W3C standards such as OWL, RDF, and RDFS, and F-Logic for the logic-based processing of rules.

The user-friendliness of OntoStudio for beginners in ontology modeling is not directly mentioned in the

sources. However, a study that compared five ontology editors, including OntoStudio, found that the main criterion for comparison was the convenience for users. The study described the basic features and structure of the editors, as well as their way of use. OntoStudio was found to be a convenient tool for users. It is worth noting that OntoEdit is considered simpler than OntoStudio but lacks some important features as apprenticeship on ontologies grows.

After reviewing the editors discussed above, we came to the conclusion that the Protégé editor would be the most suitable for our purposes. Consideration of the methodology for creating an ontology in this editor is devoted to the next section of our work.

III. ONTOLOGY EDITOR PROTÉGÉ

Protégé is an open-source ontology editor that allows users to create, edit, and manipulate ontologies. An ontology is a formal representation of a domain's knowledge that specifies a set of concepts and their relationships. Creating an ontology involves several steps, including defining the domain, identifying the key terms, and creating the class hierarchy.

To create an ontology in Protégé, one can follow these steps:

- 1) Define the domain: The first step is to determine the scope of the ontology, including the types of questions it should answer and the purpose it serves.
- 2) Identify key terms: After defining the domain, the next step is to identify the key terms that will be used in the ontology. This can be done by analyzing literature or consulting with experts. These terms should be organized into a table, including their properties or characteristics.
- 3) Create the class hierarchy: The next step is to create the class hierarchy, which involves identifying the most general concepts and gradually refining them into more specific ones. There are several approaches to creating a class hierarchy, including top-down, bottom-up, and combined approaches.
- 4) Define properties and characteristics: Once the class hierarchy is established, the next step is to define the properties and characteristics of each class. These may include attributes such as weight, habitat, and population size.
- 5) Add instances: Finally, instances of each class can be added to the ontology. These instances are specific examples of the classes in the ontology.
- 6) In addition to creating an ontology, Protégé also allows for merging ontologies and performing operations on classes, such as defining equivalent or inverse classes and transitive properties.

The advantages of using Protégé to create an ontology are:

- 1) Open-source and free: Protégé is a free and open-source ontology editor, which makes it accessible to anyone who wants to use it.
- 2) User-friendly interface: Protégé has a user-friendly interface that allows users to easily create, edit, and manipulate ontologies.
- 3) Strong community support: Protégé has a strong community of academic, government, and corporate users who use it to build knowledge-based solutions in various areas such as bio-medicine, e-commerce, and organizational modeling. This support ensures that Protégé is continually updated and maintained.
- 4) Customizable: Protégé allows users to customize their ontologies by adding new classes and properties and modifying existing ones.
- 5) Integration with other tools: Protégé can be integrated with other tools such as reasoners and visualization plugins, which makes it a powerful tool for ontology development.
- 6) Reasoning capabilities: Protégé has built-in reasoning capabilities that allow users to check the consistency of their ontologies and detect errors.
- 7) Overall, Protégé is a powerful and flexible tool for creating and managing ontologies, with a user-friendly interface, strong community support, and a range of customizable features.

IV. ONTOLOGICAL MODEL

The domain ontological model was built using the Protégé editor. We have identified the following three classes: "Documents", "Events", "People". The "TextDocuments", "Pictures", "Video" and "Audio". Class "People" — subclasses "Graduates", "Directors", "Mentors". The "Events" class is subclasses corresponding to the reign of one or another director of the corps.

The structure of the upper level of the ontology "History of the Polotsk Cadet Corps" is shown in Figure 1.

V. CREATION OF THE INTELLIGENT INFORMATION AND REFERENCE SYSTEM

We plan to complete the creation of the Intellectual Information and Reference System "History of the Polotsk Cadet Corps" using the OSTIS technology stack. We list the advantages of this approach, which guarantee the success of the completion of the project as a whole.

- 1) Any OSTIS — system can be easily supplemented with new knowledge or new methods for solving problems. This allows you to quickly and easily repurpose the developed intelligent system, reorienting it to a new range of tasks to be solved.
- 2) OSTIS — the system is focused on the reuse of the developed components. Thanks to a single and universal SC code, a library of typical components can be created, the use of which in the design



Figure 2. The structure of the upper level of the ontology "History of the Polotsk Cadet Corps"

process can reduce development time by 40-60 percent.

- 3) OSTIS — the system is reflexive, i.e. can analyze itself, due to the fact that it is fully described using the SC code. Reflexivity is one of the most important qualities of intelligent systems.
- 4) Due to the fact that the design of OSTIS — the system is reduced to the construction of its SC-model, it is completely platform-independent, and can be implemented both in software and in hardware.
- 5) Hardware implementation involves the creation of a new generation of computing devices — semantic computers.

VI. CONCLUSION

In conclusion, we note the following. Ontological modeling is an important method for developing intelligent information and reference systems. By focusing on relationships between concepts and entities, this approach ensures that knowledge is represented in an intuitive and machine-readable way.

With the help of ontological modeling, one can define a subject area, develop a conceptual model, create and populate an ontology with data, and finally put the ontology into action. Although it can be a complex process, the benefits of ontology modeling are clear: it provides a foundation for building powerful applications that address a variety of application problems.

An effective tool for building ontologies is the Protégé editor. It allows you to create classes, slots and instances, and also provides an easy way to modify ontologies without creating inconsistent data and knowledge. Protégé can be used for practical applications, such as creating ontologies for intelligent information retrieval systems,

as well as for educational purposes, such as creating ontology models for e-learning.

Note also that Protégé can be easily integrated with other software used to work with ontologies. Based on the ontology created by Protégé tools, using the OSTIS technology stack, you can quickly create an Intelligent Information Retrieval System or a digital archive of an organization.

Returning to the IIRS "History of the Polotsk Cadet Corps", we note that the electronic resource created as a result of the above steps can be used in at least three organizations. In the Euphrosyne Polotskaya State University of Polotsk — as an exhibit of the Museum of the History of Science and Education of Polotsk, in the Polotsk Cadet Corps — as an exhibit of the corps museum being created, as well as in the exposition of the Polotsk National Historical and Cultural Museum-Reserve. In addition, the resource will certainly find application in the educational process of Polotsk State University.

ACKNOWLEDGMENT

The author expresses his deep gratitude to Professor V. V. Golenkov, thanks to whose advice and support I decided to write this article.

REFERENCES

- [1] V. V. Golenkov and N. A. Guljakina *Cemanticheskaja tehnologija komponentnogo proektirovaniya sistem, upravljaemyh znanijami. Otkrytie semanticheskie tehnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, Minsk, BGUIR, 2015, pp. 57–78 (In Russ.)
- [2] A. Maedche, B. Motik, L. Stojanovic, R. Studer, R. Volz *Ontologies for enterprise knowledge management, IEEE Intelligent Systems*, vol. 18, No. 2, IEEE, 2003, pp.26–33.
- [3] R. S. Oktari, K. Munadi, R. Idroes, H. Sofyan *Knowledge management practices in disaster management: Systematic review. International Journal of Disaster Risk Reduction*, vol. 51, Elsevier, 2020, pp. 10–18.
- [4] V. A. Carriero, A. Gangemi, M. Aldo, M. L. Mancinelli, L. Marinucci, A. G. Nuzzolese, V. Presutti, C. Veninata. *ArCo: The Italian cultural heritage knowledge graph. The Semantic Web- ISWC 2019: 18th International Semantic Web Conference*, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18, Springer, 2019, pp. 36–52.
- [5] M. L. Turco, M. Calvano, E. Giovannini *Data modeling for museum collections. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, No. 2, 2019.
- [6] A. Vlachos, M. Perifanou, A. Economides *A review of ontologies for augmented reality cultural heritage applications. Journal of Cultural Heritage Management and Sustainable Development*, Emerald Publishing Limited, 2019.
- [7] M. N. Buharov *Informacionnaja sistema dlja ontologicheskogo modelirovaniya predmetnyh oblastej [Information system for ontological modeling of subject areas]. Journal of Cultural Heritage Management and Sustainable Development*, Informatika, vol. 19, No. 2, 2022 (In Russ.)
- [8] S. P. Vorona, E. O. Savkova *Ispolzovanie ontologicheskogo modelirovaniya pri razrabotke intellektual'noj sistemy dostupa k uchebno-metodicheskoj informacii [The use of ontological modeling in the development of an intelligent system for access to educational and methodological information]. Informatika, upravljajushhie sistemy, matematicheskoe i komp'juternoe modelirovanie (IUSMKM-2020)*, 2020, pp. 208–213(In Russ.)
- [9] E. Kalashnikov, V. Pavlenko *Primenenie ontologicheskogo modelirovaniya v raznyh otrasljah [Application of ontological modeling in different industries], XXV Nauchno-tehnicheskaja konferencija s mezhdunarodnym uchastiem sbornik dokladov*, P. 97 (In Russ.)
- [10] J. A. Balyberdin *Ontologicheskoe modelirovanie predmetnyh znanij v vuzah, konstruktorskih bjuro i nauchno-issledovatel'skikh institutah [Ontological modeling of subject knowledge in universities, design bureaus and research institutes]*, 2022 (In Russ.)
- [11] B. Reitemeyer, H.-G. Fill *Enterprise, Business-Process and Information Systems Modeling: 20th International Conference, BPMDS 2019, 24th International Conference, EMMSAD 2019, Held at CAiSE 2019, Rome, Italy, June 3–4, 2019, Proceedings 20*, Springer, 2019, pp. 212–226
- [12] Z. A. A. Salam, R. A. Kadir, A. Azman *Ontology Merging Using PROTÉGÉ—a Case Study. Journal of Information System and Technology Management*, IJSTM, 2021
- [13] C. Shimizu, K. Hammar, P. Hitzler *Modular graphical ontology engineering evaluated. The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17*, Springer, 2019, pp. 20–35
- [14] E. V. Glazyrin, T. E. Sohor *Polockij kadetskij korpus v otkrytyh pis'mah, vypushhennyh «Obshhestvom vzaimopomoshhi polochan» v nachale XX veka [Polotsk cadet corps in open letters issued by the Polokhan Mutual Aid Society at the beginning of the 20th century]*, 2019 (In Russ.)
- [15] S. G. Ljutko *Stanovlenie i razvitie voennogo obrazovanija na zemljah Belarusi v XVIII–nachale XX veka [Formation and development of military education on the lands of Belarus in the XVIII — the beginning of the XX century]*, Minskij gosudarstvennyj lingvisticheskij universitet, 2021 (In Russ.)
- [16] S. Poljakov and others *Polockij kadetskij korpus. Istorija v licah [Polotsk cadet corps. History in faces]*, Matjeryjaly navukovapraktychnaj kanferjency: pa vynikah navukova-dasledchaj raboty 2008 g, NPIKZ, 2009, pp.48–59 (In Russ.)
- [17] V. P. Vikent'ev *Polockij kadetskij korpus. Istoricheskij ocherk 75-letija ego sushhestvovaniya [Polotsk cadet corps. Historical sketch of the 75th anniversary of its existence]*, Polock: Tip. XV Kljachko, 1910 (In Russ.)
- [18] V. V. Golenkov, N. A. Gulyakina, I. T. Davydenko, D. V. Shunkevich *Semantic technologies of intelligent systems design and semantic associative computers. Doklady BGUIR*, 2019, No. 3, pp. 42–50
- [19] V. V. Golenkov, N. A. Gulyakina, V. A. Golovko, V. V. Krasno-proshin *Methodological problems of the current state of works in the field of Artificial intelligence. Otkrytie semanticheskie tehnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, 2021, vol. 5, pp. 17–32

Создание комплексного цифрового архива полоцкого кадетского корпуса: пример проектирования интеллектуальной информационно-справочной системы

Оськин А. Ф.

На примере проектирования и построения Интеллектуальной информационно-поисковой системы (ИИПС) «Полоцкий кадетский корпус» описаны основные этапы создания ИИПС. Кратко рассмотрены принципы онтологического моделирования и инструменты, используемые для этих целей. На основе анализа предметной области строится онтологическая модель, которая может быть преобразована в работающую ИИПС с использованием стека технологий OSTIS. Описаны преимущества технологий OSTIS для решения задач подобного рода.

Received 27.03.2023

An Automated Approach to Checking User Knowledge Levels in Intelligent Tutoring Systems

Wenzu Li

Belarussian State University Informatics and Radioelectronics

Minsk, Belarus

lwzzggml@gmail.com

Abstract—This article is dedicated to the issue of automating the implementation of rapid testing of user knowledge in new generation intelligent tutoring systems. A semantic-based approach to automating the entire process from test question generation and test paper generation to the automatic verification of user answers and the automatic scoring of test papers is described in detail in this article.

Keywords—testing user knowledge level, test question generation, user answer verification, intelligent tutoring systems, test paper generation, automatic scoring of test papers

I. INTRODUCTION

Educators have long shared a common desire to use computers to automate teaching and learning services. In recent years, with the development of artificial intelligence technology, this wish is likely to become a reality. The most representative product combining artificial intelligence and education is the intelligent tutoring system (ITS), which can not only improve the learning efficiency of users, but also ensure the fairness and impartiality of the education process [9].

Automatic generation of test questions and automatic verification of user answers are the most basic and important functions of ITS. Using these two functions in combination will enable the entire process from the automatic generation of test questions to the automatic scoring of the user test papers. This will not only greatly reduce the repetitive work of educators, but will also reduce the cost of user learning, thus providing more users with the opportunity to learn various knowledge [1], [2], [7].

Although in recent years, with the development of technologies such as the semantic web, deep learning and natural language processing (NLP), several approaches have been proposed for the automatic generation of test questions and the automatic verification of user answers, these approaches have the following main disadvantages:

- existing approaches to generating test questions allow only the simplest objective questions to be generated;
- some existing approaches (for example, keyword matching and probability statistics) to verifying user

answers to subjective questions do not consider the semantic similarity between answers;

- methods that use semantic to verify user answers to subjective questions can only calculate similarity between answers with simple semantic structures [7], [8], [10].

Objective questions usually have a unique standard answer. In this article, objective questions include: multiple-choice questions, fill in the blank questions and judgment questions. Objective questions differ from subjective questions, which have more than one potential correct answer. Subjective questions in this article include: definition explanation questions, proof questions and problem-solving task.

Therefore, based on existing methods and OSTIS Technology, an approach to developing a universal subsystem for automatic generation of test questions and automatic verification of user answers in tutoring systems developed using OSTIS Technology (Open Semantic Technology for Intelligent Systems) is proposed in this article [1], [2], [5]. The universality of the subsystem means that the developed subsystem can be easily transplanted to other ostis-systems (system built using OSTIS Technology). The developed subsystem allows the use of the knowledge bases of the ostis-systems to automatically generate various types of test questions and automatically verify the completeness and correctness of user answers based on the semantic description structures of the knowledge. The discrete mathematics ostis-system will be used as demonstration systems for the developed subsystem.

II. EXISTING APPROACHES AND PROBLEMS

A. Automatic generation of test questions

Approach to automatic generation of test questions mainly studies how to use electronic documents, text corpus and knowledge bases to automatically generate test questions. Among them, the knowledge base stores highly structured knowledge that has been filtered, and with the development of semantic networks, using the knowledge base to automatically generate test questions has become the most important research direction in

the field of automatic generation of test questions. The basics of how to use the knowledge base to automatically generate objective questions are described in detail in the literature [10], [12].

The main problems with the existing approaches to test question generation are as follows:

- the approach of using electronic documents to automatically generate test questions requires a large number of sentence templates;
- the creation of text corpus requires a lot of human resources to collect and process various knowledge;
- existing approaches only allow to generate simple objective questions [11], [13].

B. Automatic verification of user answers

Automatic verification of user answers is divided into verification of answers to objective questions and verification of answers to subjective questions. The basic principle of verification of answers to objective questions is to determine whether the string of standard answers matches the string of user answers. The basic principle of verification of answers to subjective questions is to calculate the similarity between standard answers and user answers, and then to implement automatic verification of user answers based on the calculated similarity and the evaluation strategy of the corresponding test questions [14], [15]. The verification of answers to subjective questions was classified according to the approach to calculating the similarity between answers as follows:

- Based on keyword phrases
This type of approach first allows to split the sentences into keyword phrases and then calculate the similarity between them according to the matching relationship of keyword phrases between sentences [16].
- Based on vector space model (VSM)
The basic principle of VSM is to use machine learning algorithms to first convert sentences into vector representations, and then calculate the similarity between them [17].
- Based on deep learning
This type of approach allows the use of neural network models to calculate the similarity between sentences. Representative neural network models include: Tree-LSTM, Transformer and BERT [18].
- Based on semantic graph
The basic principle of calculating the similarity between answers (i.e., sentence or short text) using this type of approach is to first convert the answers into a semantic graph representation using natural language processing tools, and then calculate the similarity between them. A semantic graph is a network that represents semantic relationships between concepts. In the ostis-systems, the semantic graph is constructed using SC-code (as a basis for knowledge

representation within the OSTIS Technology, a unified coding language for information of any kind based on semantic networks is used, named SC-code) [1]. The main advantage of this type of approach is computing the similarity between answers based on semantics. One of the most representative approaches is SPICE (Semantic Propositional Image Caption Evaluation) [19].

The main disadvantages of the existing methods are as follows:

- keyword phrase and VSM based approaches do not take into account semantic similarity between answers;
- semantic graph-based approaches supporting only the description of simple semantic structure;
- these approaches cannot determine the logical equivalence between answers;
- these approaches are dependent on the corresponding natural language.

In ITS information is described in the form of semantic graphs and stored in the knowledge base. Therefore based on existing approaches and OSTIS Technology an approach to automatically generate test questions using knowledge bases and verify user answers based on the similarity between semantic graphs is proposed in this article.

III. PROPOSED APPROACH

The subsystem can be divided into two parts according to the functions to be implemented: automatic generation of test questions and automatic verification of user answers. In the following, the functions of these two parts will be introduced separately.

A. Automatic generation of test questions

The basic principle of automatic generation of test questions in the ostis-systems is to first extract the corresponding semantic fragments from the knowledge base using a series of test question generation strategies, then add some test question description information to the extracted semantic fragments, and finally store the semantic fragments describing the complete test questions in the universal subsystem. The subsystem allows a series of test questions to be extracted from the subsystem and formed into test papers according to the user's requirements when test papers need to be generated. Test papers consisting of semantic graphs of test questions are converted to natural language descriptions using a nature language interface. An approach to developing natural language interface using OSTIS Technology is described in the literature [6]. In the following, the basic principles of automatic generation of test questions in the ostis-systems will be introduced using test question generation strategy based on class as examples.

The inclusion relation is one of the most frequently used relations in the knowledge base of the ostis-systems, which is satisfied between many classes (including subclasses), so that the inclusion relation between classes can be used to generate objective questions. The set theory expression form of inclusion relation between classes is as follows: $S_i \subseteq C(i \geq 1)$, (S -subclass, i -subclass number, C -parent class) [5], [7]. The following shows a semantic fragment in the knowledge base that satisfies the inclusion relation in SCn-code (one of SC-code external display languages) [1]:

binary tree

- ⇐ inclusion*:
directed tree
- ⇒ inclusion*:
 - binary sorting tree
 - brother tree
 - decision tree

An example of a judgement question generated using this semantic fragment is shown below in natural language: <<Binary tree contains binary sorting tree, brother tree and decision tree.>>

True False

Other types of objective questions can be generated using this strategy.

Other strategies used to generate objective questions include:

- Test question generation strategy based on elements;
- Test question generation strategy based on identifiers;
- Test question generation strategy based on axioms;
- Test question generation strategy based on relation attributes;
- Test question generation strategy based on image examples.

The process of generating subjective questions is shown below:

- searching the knowledge base for semantic fragments describing subjective questions using logic rules;
- storing the found semantic fragments in the knowledge base of the subsystem;
- using manual approaches or automatic approaches (such as natural language interfaces) to describe the definition, proof process or solution process of the corresponding test question according to the knowledge representation rules in SCg-code (SCg-code is a graphical version for the external visual representation of SC-code) or SCL-code (a special sub-language of the SC language intended for formalizing logical formulas) [2].

The proposed approach to generating test questions has the following main advantages:

- within the framework of OSTIS Technology, knowledge is described in a uniform form and structure,

so that the component developed using the proposed approach to generating test questions can be used in different ostis-systems;

- the semantic models of the test questions are described using SC-code, so that they do not rely on any natural language;
- using the proposed approach, high quality objective and subjective questions can be generated automatically.

B. Automatic verification of user answers

In the ostis-systems, test questions are stored in the knowledge base in the form of semantic graphs, so the most critical step of user answer verification is to calculate the similarity between the semantic graphs of answers, and when the similarity is obtained and combined with the evaluation strategy of the corresponding test questions, the correctness and completeness of user answers can be verified.

Since the knowledge types and knowledge structures used to describe different types of test questions are not the same, answer verification is further divided into: 1. verification of answers to objective questions; 2. verification of answers to subjective questions.

C. Verification of answers to objective question

Semantic graphs of answers to objective questions are described using factual knowledge according to the same knowledge structures, so the similarity between the semantic graphs of answers to different types of objective questions can be calculated using the same approach. Factual knowledge refers to knowledge that does not contain variable types, and this type of knowledge expresses facts. When the user answers to objective questions in natural language are converted into semantic graphs, they are already integrated with the knowledge already in the knowledge base. So the similarity between answers is calculated based on the semantic description structures [19]. The process of calculating the similarity between the semantic graphs of the answers to the objective questions is shown below:

- decomposing the semantic graphs of the answers into sub-structures according to the structure of the knowledge description;
- using formulas (1), (2), and (3) to calculate the precision, recall and similarity between semantic graphs.

$$P_{sc}(u, s) = \frac{|T_{sc}(u) \otimes T_{sc}(s)|}{|T_{sc}(u)|} \quad (1)$$

$$R_{sc}(u, s) = \frac{|T_{sc}(u) \otimes T_{sc}(s)|}{|T_{sc}(s)|} \quad (2)$$

$$F_{sc}(u, s) = \frac{2 \cdot P_{sc}(u, s) \cdot R_{sc}(u, s)}{P_{sc}(u, s) + R_{sc}(u, s)} \quad (3)$$

The parameters are defined as shown below:

- $T_{sc}(u)$ — all substructures after the decomposition of the user answers u ;
- $T_{sc}(s)$ — all substructures after the decomposition of the standard answers s ;
- \otimes — binary matching operator, which represents the number of matching substructures in the set of two substructures.

Once the similarity between the answers is obtained, the correctness and completeness of the user answers can be verified by combining it with the corresponding evaluation strategy. The evaluation strategy of the objective questions is shown below:

- if there is only one correct option for the current test question, only if the standard answer and the user answer match exactly, the user answer is considered correct;
- if the current question has multiple correct options:
 - as long as the user answer contains an incorrect option, the user answer is considered incorrect;
 - if all the options in the user answer are correct, but the number of correct options is less than the number of correct options in the standard answer, the user answer is considered correct but incomplete. At this time, the user answer score is $R_{sc} * Max_{score}$;
 - if all the options in the standard answer match exactly with all the options in the user answer, the user answer is exactly correct.

Fig. 1 shows an example of verification of user answer to judgment question in SCg-code.

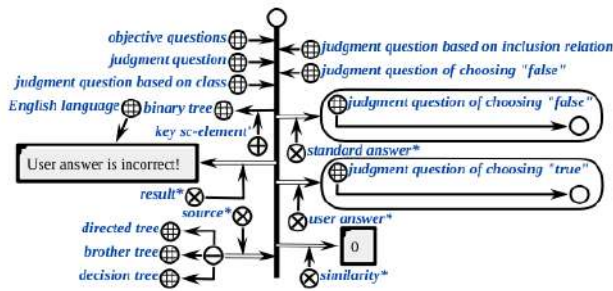


Figure 1. An example of verification of user answer to judgment question.

D. Verification of answers to subjective questions

The approach to calculating the similarity between the semantic graphs of answers to subjective questions, according to the knowledge description structure of the different types of subjective questions, has been divided into: 1. the approach to calculating the similarity between answers to definition explanation questions; 2. the approach to calculating the similarity between answers to proof questions and problem-solving task.

Calculating the similarity between answers to definition explanation questions

The answers to the definition explanation questions are described based on logical formulas (SCL-code). Logic formulas are powerful tools for formal knowledge representation in the framework of OSTIS Technology, which are expanded based on the first-order predicate logic formulas [5]. In the process of calculating the similarity between the semantic graphs of answers to this type of test question, the following tasks need to be solved:

- establishing the mapping relationship of potential equivalent variable sc-node pairs between the semantic graphs of the answers;
- calculating the similarity between semantic graphs;
- if the similarity between semantic graphs is not equal to 1, they also need to be converted to the prenex normal form (PNF) representation separately, and then the similarity between them is calculated again [23].

The semantic graphs of answers to the definition explanation questions are constructed based on logical formulas, the variables sc-nodes (bound variables) are included in the semantic graphs. In order to calculate the similarity between semantic graphs, mapping relationship of potential equivalent variable sc-node pairs between them needs to be established.

In the ostis-systems, the sc-construction composed of sc-tuple, relation sc-node, role relation sc-node and sc-connector is used to describe logical connectives (such as negation (\neg) and implication (\rightarrow), etc.) and quantifiers (universal quantifier (\forall) and existential quantifier (\exists)), atomic logic formula (various sc-constructions) or multiple atomic logic formulas that satisfy conjunctive relation are contained in the sc-structure and connected with the corresponding sc-tuple, and these sc-elements together constitute the semantic graph of answers to the definition explanation questions. Its structure is a tree.

If the standard answer and the user answer are exactly equal, it means that the atomic logic formulas with the same semantics between the answers have the same position in the semantic graph. Thus a mapping relationship between variables sc-nodes can be established by determining the position in the semantic graph of each sc-construction containing the variable sc-nodes and the semantic connotation it expresses [20], [21], [22].

The process of establishing the mapping relationship of the potential equivalent variable sc-node pairs between answers is shown below:

- each sc-tuple and sc-structure in the semantic graph is numbered separately according to the depth-first search strategy (DFS), (for indirectly determining the position of variables sc-nodes in the semantic graph);

- according to the matching relationship of each sc-element between each sc-construction pair with the same number in the semantic graph of the standard answer and the semantic graph of the user answer, the mapping relationships of potential equivalent variable sc-nodes pairs between the semantic graphs are established.

Fig. 2 shows an example of establishing the mapping relationship between semantic graphs in SCg-code.

In Fig. 2, the definition of the partial ordering relation is described. A binary relation R is called a partial ordering, or partial order if and only if it is: reflexive, antisymmetric and transitive.

When the mapping relationship between the potential equivalent variable sc-node pairs between the semantic graphs is established, the similarity between answers can be calculated, and the detailed calculation process is shown below:

- decomposing the semantic graphs of the answers into substructures according to the structure of the knowledge description;
- establishing the mapping relationship of potential equivalent variable sc-node pairs between the semantic graphs;
- using formulas (1), (2) and (3) to calculate the precision, recall and similarity between semantic graphs.

If the similarity between semantic graphs is not equal 1, it is also necessary to determine whether their logical formulas are logically equivalent. Because any predicate logic formula has a PNF equivalent to it. Therefore, based on the approach to convert predicate logic formulas into PNF and characteristics of logic formulas in ostis-systems, an approach to convert logic formulas into unique (deterministic) PNF according to strict restriction rules is proposed in this article [23], [24]. The strict restrictions mainly include the following:

- renaming rule is preferred when converting logical formulas to PNF;
- existential quantifier is moved to the front of the logical formula in preference;
- the logical formula can usually be expressed in the following form: $(Q_1x_1Q_2x_2\dots Q_nx_n(A \leftrightarrow B))$, where $Q_i(i = 1, \dots, n)$ is a quantifier. A is used to describe the definition of a concept at a holistic level, and it does not contain any quantifiers. B is used to explain the semantic connotation of a definition at the detail level, and it is usually a logical formula containing quantifiers [8], [24]. Therefore, in order to simplify the knowledge processing, it is only necessary to convert the logical formula B to PNF;

The process of converting the semantic graph constructed based on logic formula into PNF descriptions is shown below:

- if there are multiple sc-structures connected by the same conjunctive connective, the sc-constructions contained in them are merged into the same sc-structure;
- eliminating all the implication connectives;
- moving all negative connectives to the front of the corresponding sc-structure;
- using renaming rules so that all bound variables in the semantic graphs are not the same;
- moving all quantifiers to the front of the logical formula;
- merging again the sc-structures in the semantic graphs that can be merged.

If the calculated similarity between the semantic graphs of PNF representation is not 1, the similarity between the semantic graphs calculated for the first time is used as the final answer similarity.

Fig. 3 shows an example of converting a semantic graph into PNF representation in SCg-code.

In Fig. 3, the definition of the reflexive relation is described. In mathematics, a binary relation R on a set M is reflexive if it relates every element of M to itself.

Calculating the similarity between answers to proof questions and problem-solving task

Both proof questions and problem-solving task follow a common task-solving process:

- 1) the set (Ω) of conditions consisting of some known conditions;
- 2) deriving an intermediate conclusion using some of the known conditions in Ω and adding it to Ω . Each element in Ω can be regarded as a solving step;
- 3) repeat step 2) until the final result is obtained [25], [26].

This task-solving process is abstracted as a directed graph, whose structure is in most cases an inverted tree, and is called a reasoning tree (i. e. the reasoning tree of the standard answer). The automatic verification process of user answers to this type of test questions is the same as the traditional manual answer verification process, i.e., verifying whether the current solving step of the user answer is a valid conclusion of the partial solving step preceding that step. This means whether the solving step in the user answer corresponding to the parent node in the reasoning tree always is located after the solving steps in the user answer corresponding to the child nodes [27].

The semantic graphs of user answers to proof questions and problem-solving task in the ostis-systems are linear structures consisting of some semantic sub-graphs for describing the solving steps and some semantic fragments for describing the logical order and transformation processes between the semantic sub-graphs. The semantic graph of standard answers to this type of test questions is an reasoning tree consisting of a number of search templates (which can be abstracted as the nodes in the tree). Each search template is constructed using SCL-

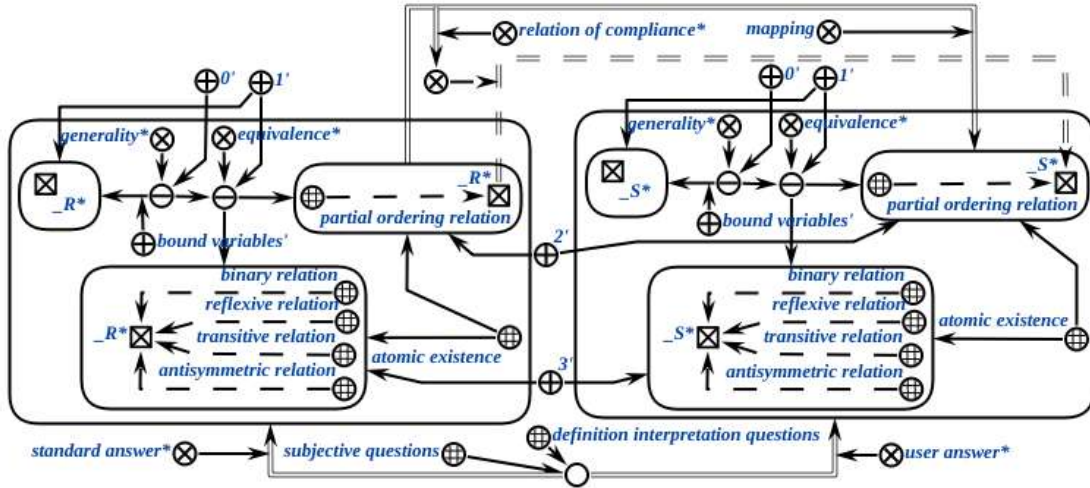


Figure 2. An example of establishing the mapping relationship between semantic graphs.

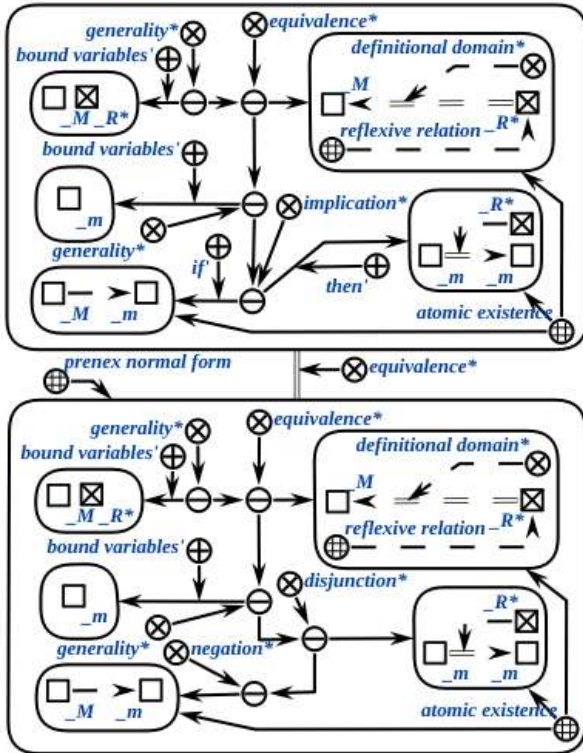


Figure 3. An example of converting a semantic graph into PNF representation.

code in strict accordance with the standard solution steps corresponding to the test question. The search template is used to search in the knowledge base for all semantic fragments corresponding to it [7], [28].

Since the user answers in natural language are converted into semantic graphs they are already integrated

with the knowledge already available in the knowledge base. Therefore, when calculating the similarity between the semantic graphs, it is not necessary to consider the differences of the concepts at the natural language level. For example, Segment AB and Segment BA are represented by the same sc-node, they are just two identifiers of the sc-node [6], [7]. An approach to calculate the similarity between the semantic graphs of answers to proof questions and problem-solving task according to the reasoning tree of standard answer (semantic graph of standard answer) is proposed in this article, and the specific calculation process is shown below:

- 1) numbering each semantic sub-graph in the semantic graph of user answer (the numbering order started from 1);
- 2) each node in the reasoning tree (search template) is traversed in turn according to the DFS strategy. At the same time, the corresponding semantic sub-graph that is included in the semantic graph of the user answer are searched in the knowledge base using the search template currently being traversed. If such a semantic sub-graph exists, then determine whether the searched semantic sub-graph number is smaller than the semantic sub-graph number corresponding to the search template of the current search template parent node (except for the root node of the reasoning tree), and if so, the searched semantic sub-graph is considered correct;
- 3) repeat step 2) until all search templates in the reasoning tree have been traversed and the number of correct semantic sub-graphs is counted at the same time;
- 4) using formulas (1), (2) and (3) to calculate the precision, recall and similarity between answers.

Since this article focuses on the entire process from test

question generation to the automatic scoring of test papers and the effective evaluation of subsystems, the basic principles of answer verification to subjective questions are thus briefly presented. For a detailed understanding of the process of constructing semantic models for subjective questions and user answer verification to subjective questions please refer to the literature [3].

Once the similarity between the answers to the subjective questions is obtained, the correctness and completeness of the user answers can be verified combined with the evaluation strategy for the subjective questions. The evaluation strategy for subjective questions includes:

- if the similarity between the answers is equal to 1, the user answer is completely correct;
- if the similarity between the answers is less than 1 and the precision is equal to 1, the user answer is correct but incomplete and the user score is $R_{sc} * Max_{score}$;
- if the similarity between the answers is greater than 0 and less than 1, and the precision is less than 1, then the user answer is partially correct and the user score is $F_{sc} * Max_{score}$;
- if the similarity between the answers is equal to 0, the user answer is wrong.

The proposed approach to automatic verification of user answers has the following advantages:

- verifying the correctness and completeness of user answers based on semantics;
- the logical equivalence between answers can be determined;
- the similarity between any two semantic graphs in the knowledge base can be calculated;
- the developed component using the proposed approach can be easily transplanted to other ostis-systems.

IV. KNOWLEDGE BASE OF THE SUBSYSTEM

The knowledge base of subsystem is used to store automatically generated test questions, and it also allows to automatically extract a series of test questions and form test papers according to user requirements. Therefore, in order to improve the efficiency of accessing the knowledge base of the subsystem and the efficiency of extracting the test questions, an approach to construct the knowledge base of the subsystem according to the type of test questions and the generation strategy of the test questions is proposed in this article.

The basis of the knowledge base of any ostis-system (more precisely, the sc-model of the knowledge base) is a hierarchical system of subject domains and their corresponding ontologies [1], [2], [5]. Let's consider the hierarchy of the knowledge base of subsystem in SCn-code:

Section. Subject domain of test questions

```

⇐ section decomposition*:
{• Section. Subject domain of subjective
  questions
⇐ section decomposition*:
{• Section. Subject domain of
  definition explanation question
  • Section. Subject domain of proof
  question
  • Section. Subject domain of
  problem-solving task
}
• Section. Subject domain of objective
  questions
⇐ section decomposition*:
{• Section. Subject domain of
  multiple-choice question
  • Section. Subject domain of fill in
  the blank question
  • Section. Subject domain of
  judgment question
}
}

```

Objective types of test questions are decomposed into more specific types according to their characteristics and corresponding test question generation strategies. Next, taking the judgment question as an example let us consider its semantic specification in SCn-code:

judgment question

```

∈ maximum class of explored objects':
  Subject domain of judgment question
⇐ subdividing*:
{• judgment question based on relation
  attributes
  • judgment question based on axioms
  • judgment question based on image
  examples
  • judgment question based on identifiers
  • judgment question based on elements
⇐ subdividing*:
{• judgment question based on role
  relation
  • judgment question based on
  binary relation
}
• multiple-choice question based on classes
⇐ subdividing*:
{• judgment question based on
  subdividing relation
  • judgment question based on
  inclusion relation
  • judgment question based on strict
  inclusion relation
}

```

```

}
← subdividing*:
{• judgment question of choosing true
• judgment questions of choosing false
}

```

V. PROBLEM SOLVER

One of the most important components of every intelligent system is the problem solver, which provides the ability to solve a variety of problems. The problem solver of any ostis-system (more precisely, the sc-model of the ostis-system problem solver) is a hierarchical system of knowledge processing agents in semantic memory (sc-agents) that interact only by specifying the actions they perform in the specified memory [1], [4].

Therefore, a problem solver for automatic generation of test questions and automatic verification of user answers has been developed based on the proposed approach, and its hierarchy is shown below in SCn-code:

Problem solver for the automatic generation of test questions and automatic verification of user answers

```

← decomposition of an abstract sc-agent*:
{• Sc-agent for automatic generation of test
  questions
← decomposition of an abstract sc-agent*:
{• Sc-agent for quick generation of
  test questions and test papers
• Sc-agent for generating single
  type of test questions
• Sc-agent for generating a single
  test paper
}
• Sc-agent for automatic verification of
  user answers
← decomposition of an abstract sc-agent*:
{• Sc-agent for automatic scoring of
  test papers
• Sc-agent for calculating similarity
  between answers to objective
  questions
• Sc-agent for calculating the
  similarity between answers to
  definition explanation questions
• Sc-agent for converting a logical
  formula into PNF
• Sc-agent for calculating the
  similarity between the answers to
  proof questions and
  problem-solving task
}
}

```

The function of the sc-agent for quick generation of test questions and test papers is to automate the entire process

from test question generation to test paper generation by initiating the corresponding sc-agents (sc-agent for generating single type of test questions and sc-agent for generating a single test paper).

The function of the sc-agent for automatic scoring of test papers is to implement automatic verification of user answers to test questions and automatic scoring of test papers by initiating sc-agents for calculating the similarity between user answers and sc-agents for converting a logical formula into PNF.

VI. EVALUATING THE EFFECTIVENESS OF THE SUBSYSTEM

The effectiveness of the developed subsystem will be evaluated from the following aspects:

- availability of the generated test questions;
- difficulty level of the generated test papers;
- closeness between automatic scoring and manual scoring of user answers to subjective questions.

In order to evaluate the availability of the automatically generated test questions, 200 automatically generated test questions were randomly sampled from the tutoring system for discrete mathematics and the proportion of test questions that could be used directly was counted (Table I).

Table I
TABLE. RESULTS OF THE EVALUATION OF THE AVAILABILITY OF THE GENERATED TEST QUESTIONS

Availability indicators	Test questions that can be used directly	Test questions that can be used after modification	Unavailable test questions
Number of test questions (total 200)	188	12	0
Proportion	94%	6%	0

It can be seen from Table I that of the 200 automatically generated test questions sampled at random, 94% were able to be used directly and 6% were able to be used after modification.

The difficulty of the test paper is closely related to the user's score. Therefore, 40 second-year students were randomly selected to evaluate the difficulty of the test paper for discrete mathematics, which was automatically generated using the subsystem. 10 multiple-choice questions, 10 fill in the blank questions, 10 judgment questions, 2 definition explanation questions and 2 proof questions are included in this test paper. The maximum score for each objective question is 2 points, the maximum score for each subjective question is 10 points, and the maximum score for the whole test paper is 100 points (Table II).

From the Table II, it can be seen that the students' scores generally follow a normal distribution. Therefore, it can be concluded that the difficulty of the current type

Table II
TABLE. STATISTICAL RESULTS OF STUDENT SCORES

Score	<40	[40-49]	[50-59]	[60-69]	[70-79]	[80-89]	≥90
Total number of students (40)	0	1	4	10	14	8	3
Proportion	0	2.5%	10%	25%	35%	20%	7.5%
Average score	72.85						

of test paper is moderate and that the actual knowledge level of the user can be checked objectively and fairly.

In order to evaluate the closeness between the automatic scoring and manual scoring of user answers to the subjective questions, we decided to first enter the 40 students' answers to the subjective questions into the subsystem, then use the subsystem to automatically verify the students' answers, and finally count the error between the automatic scoring and manual scoring of user answers to the subjective questions (Table III).

Table III
TABLE. RESULTS OF SCORING ERROR STATISTICS FOR USER ANSWERS TO SUBJECTIVE QUESTIONS

Error range (Φ)	Definition explanation question 1	Definition explanation question 2	Proof question 1	Proof question 2	Total	Proportion
$\Phi \leq 1$	35	31	26	28	120	75%
(1-1.5]	2	4	8	8	22	13.75%
(1.5-2]	2	3	5	3	13	8.125%
$\Phi > 2$	1	2	1	1	5	3.125%

The formula for calculating the error Φ is shown below (4):

$$\Phi = |x - y| \quad (4)$$

The parameters are defined as shown below:

- x — is the manual scoring of user answers to the test questions;
- y — is the automatic scoring of user answers to the test questions;

From the Table III, it can be seen that the automatic scoring and manual scoring of user answers to subjective questions in the tutoring system for discrete mathematics generally remained consistent, and that when the maximum score for a subjective question was 10, the sample size with an error $\Phi \leq 1.5$ between scores was over 88%.

The above experimental results show that the developed subsystem can satisfy the conditions for practical applications.

VII. CONCLUSION

An automated approach to checking the knowledge level of users in tutoring systems developed using OSTIS

Technology is proposed in this article. Based on the proposed approach, a universal subsystem for automatic generation of test questions and automatic verification of user answers is developed. Using the developed subsystem, the entire process can be automated from test question generation, test paper generation to automatic verification of user answers and automatic scoring of test papers.

Finally the effectiveness of the developed subsystem was evaluated in terms of the availability of the generated test questions, the difficulty of the generated test papers and the closeness between the automatic scoring and the manual scoring of the test questions in the discrete mathematics ostis-system. From the evaluation results, it can be seen that the developed subsystem can meet the conditions for practical application.

ACKNOWLEDGMENT

The work in this article was done with the support of research teams of the Department of Intelligent Information Technologies of Belarusian State University of Informatics and Radioelectronics. Authors would like to thank every researcher in the Department of Intelligent Information Technologies.

REFERENCES

- [1] V. V. Golenkov and N. A. Guljakina, "Proekt otkrytoj semanticheskoy tehnologii komponentnogo proektirovaniya intellektual'nyh sistem. chast' 1: Principy sozdaniya project of open semantic technology for component design of intelligent systems. part 1: Creation principles]," *Ontologija proektirovaniya [Ontology of design]*, no. 1, pp. 42–64, 2014.
- [2] V. Golenkov, N. Gulyakina, "Next-generation intelligent computer systems and technology of complex support of their life cycle," in *Otkrytye semanticheskie tehnologii proektirovaniya intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2022, pp. 27–40.
- [3] D. Shunkevich, "Metodika komponentnogo proektirovaniya sistem, upravlyaemyh znaniyami," in *Otkrytye semanticheskie tehnologii proektirovaniya intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2015, pp. 93–110.
- [4] D. Shunkevich, "Hybrid problem solvers of intelligent computer systems of a new generation," in *Otkrytye semanticheskie tehnologii proektirovaniya intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2022, pp. 119–144.
- [5] (2023, Mar) *Ims.ostis metasystem*. [Online]. Available: <https://ims.ostis.net>
- [6] Longwei. Qian, "Ontological Approach to the development of natural language interface for intelligent computer systems," in *Otkrytye semanticheskie tehnologii proektirovaniya intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2022, pp. 217–238.
- [7] Wenzu. Li, "A semantics-based approach to automatic generation of test questions and automatic verification of user answers in the intelligent tutoring systems," in *Otkrytye semanticheskie tehnologii proektirovaniya intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2022, pp. 369–374.
- [8] Li. W., Qian L. *Semantic Approach to User Answer Verification in Intelligent Tutoring Systems*. *Communications in Computer and Information Science*, 2022, Vol. 1625, pp. 242–266.

- [9] Jiang. H., Zhao. Y., Li. X. Intelligent education system: A comprehensive review. *IEEE Access*, 2020, Vol. 8, pp. 201016-201026.
- [10] Li. H. Research on item automatic generation based on DL and domain ontology. *Journal of Changchun University of Technology (Natural Science Edition)*, 2012, Vol. 33(04), pp. 461-464.
- [11] Kaur. M., Batra. S. Ontology-based question generation: a review. *Journal of Intelligent and Fuzzy Systems*, 2020, Vol. 39(01), pp. 49-57.
- [12] Andreas. P., Konstantinos. K., Konstantinos. K. Automatic generation of multiple-choice questions from domain ontologies. In: *IADIS International Conference e-Learning*, 2008, pp. 427-434.
- [13] Arjun. S. B., Manas. K., Sujan. K. S. Automatic Generation of Multiple Choice Questions Using Wikipedia. *International Conference on Pattern Recognition and Machine Intelligence*, 2013, Vol. 8251, pp. 733-738.
- [14] Chen. C. H. A comprehensive survey of text similarity measures. *Journal of Information Science*, 2019, Vol. 45(01), pp. 1-19.
- [15] Singh. A., Singh. V. K. Text similarity techniques: a systematic literature review. *International Journal of Computer Applications*, 2019, Vol. 182(41), pp. 23-30.
- [16] Baly. R., Jansen. P., De. Rijke. M. A review of evaluation methodologies in automatic short answer grading. *Educational Research Review*, 2014, Vol. 13, pp. 20-33.
- [17] Kim. Y., Kwon. O. N., Kim. Y. Automatic short answer grading using latent semantic analysis with fuzzy matching. *Journal of Educational Technology and Society*, 2016, Vol. 19(03), pp. 257-269.
- [18] Mingyu. Ji., Xinhai. Zhang. A Short Text Similarity Calculation Method Combining Semantic and Headword Attention Mechanism. *Scientific Programming*, 2022.
- [19] Anderson. P., Fernando. B., Johnson. M., Gould. S. Spice: Semantic propositional image caption evaluation. In: *European Conference on Computer Vision*, Springer, 2016, pp. 382-398.
- [20] Shu. X., Wang. S., Guo. L., Zhao. Y. Survey on multi-source information fusion: progress and challenges. *Information Fusion*, 2018, Vol. 42, pp. 146-169.
- [21] Wang. X. Y., Hu. Z. W., Bai. R. J., et al. Review on Concepts, Processes, Tools and Methods Ontology Integration. *Library and Information Service*, 2011, Vol. 55(16), pp. 119-125.
- [22] Liu. J., Liu. Y., Wang. X., Wang. B. A survey on ontology integration techniques. *Future Generation Computer Systems*, 2020, Vol. 105, pp. 17-30.
- [23] Fujiwara. M., Kurahashi. T. Prenex normal form theorems in semi-classical arithmetic. *The Journal of Symbolic Logic*, 2022, pp. 1-31.
- [24] Pan. M., Ding. Z. A simple method for solving prenex disjunction (conjunction) normal forms. *Computer Engineering and Science*, 2008, Vol. 30(10), pp. 82-84.
- [25] Ge. L., Li. X. An overview of the methods for automated theorem proving in geometry. *Journal of Symbolic Computation*, 2016, Vol. 74, pp. 303-335.
- [26] Kryakin. A., Nguyen. P. Automated theorem proving in geometry using Gröbner bases. *Journal of Automated Reasoning*, 2014, Vol. 53(02), pp. 163-189.
- [27] Reiter. R. *A Logic for Default Reasoning*. *Artificial Intelligence*, 1980, Vol. 13(1-2), pp. 81-132.
- [28] Chou. S. C., Gao. X. S., Zhang. J. *Machine Proofs in Geometry: Automated Production of Readable Proofs for Geometry Theorems*. World Scientific, 1994.

Автоматизированный подход к проверке уровня знаний пользователей в интеллектуальных обучающих системах

Ли Вэньцзу

Данная работа посвящена проблеме автоматизации реализации быстрого тестирования знаний пользователей в интеллектуальных обучающих системах нового поколения. В данной работе подробно описывается основанный на семантике подход к автоматизации всего процесса от генерации тестовых вопросов и экзаменационных билетов до автоматической проверки ответов пользователей и автоматической оценки экзаменационных билетов.

На протяжении многих лет педагоги активно высказывают желание использовать компьютеры для автоматизации обучения и преподавания. С развитием технологии искусственного интеллекта в последние годы, это желание может наконец-то стать реальностью. Наиболее представительным продуктом, объединяющим искусственный интеллект и образование, являются интеллектуальные обучающие системы (ИОС), которые могут не только значительно повысить эффективность обучения пользователей, но и обеспечить справедливость и беспристрастность образовательного процесса.

Автоматическая генерация тестовых вопросов и автоматическая проверка ответов пользователей являются самыми основными и важными функциями ИОС. Использование этих двух функций в комбинации позволит реализовать весь процесс от автоматической генерации тестовых вопросов до автоматической оценки экзаменационных билетов пользователей. Это не только значительно сократит повторяющуюся работу педагогов, но и снизит стоимость обучения для пользователей, что позволит большему числу людей получить доступ к различным знаниям.

Хотя в последние годы благодаря развитию таких технологий, как семантические сети, глубокое обучение и обработка естественного языка (NLP), было предложено несколько подходов для автоматической генерации тестовых вопросов и проверки ответов пользователей, эти методы все еще имеют следующие основные недостатки:

- существующие подходы к генерации тестовых вопросов позволяют генерировать только самые простые объективные вопросы;
- некоторые из существующих подходов (например, сопоставление ключевых слов и использование статистической вероятности) для проверки ответов пользователей на субъективные вопросы не учитывают семантическое сходство между ответами;
- методы, использующие семантику для проверки ответов пользователей на субъективные вопросы, могут вычислять сходство только между ответами с простыми семантическими структурами;
- и т.д.

Поэтому на основе существующих методов и Технологии OSTIS в данной работе предлагается подход к разработке универсальной подсистемы для автоматической генерации тестовых вопросов и автоматической проверки ответов пользователя в обучающих системах, разработанных с использованием Технологии OSTIS (открытая семантическая технология проектирования интеллектуальных систем).

Received 29.03.2023

Model of Interoperability of Information Systems of Information and Communication Environment of Secondary Special Education Institution

Nikolai Listopad and Lizaveta Bushchik

Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus

Email: listopad@bsuir.by, e.bushchik@bsuir.by

Abstract—The paper proposes a model of interoperability of information systems of information and communication environment of secondary vocational education institution. It shows what should be understood under the interoperability of information systems in relation to the industry of education on the example of secondary special education institution. The system of business processes in the institution of specialized secondary education in the following directions: controlling, operating (basic, describing the educational process of the institution) and supporting is presented. The author's root model of business processes of educational institution is given. The well-known reference model of interoperability in the form of three levels - organizational, semantic and technical - is specified and supplemented by the parameters which were detailed for the college. The three-level model of interoperability of college information systems is refined on the basis of the processor approach, which allowed creating a flexible and adaptive management in the institution of secondary special education.

Keywords—digital transformation, interoperability, information and communication environment, process management

I. Introduction

At present, information systems, different in their composition, parameters and characteristics, are developing towards their integration and globalization. The term unified information space, which is understood as the interaction of various information systems for the exchange and use of information under common protocols, standards and rules, sounds more and more often. In this case, such knowledge-intensive areas as economics, industry, and defense are mainly analyzed.

Digital transformation, in the new era of digital economy, is impossible without the creation and development of a heterogeneous information and communication environment, the transparency of which is ensured through the use of open systems principles. One of the most important features of such open systems is interoperability, which is understood as the ability of systems and components to interact (exchange information and use

the information obtained from the exchange), based on the use of information and communication technologies (ICT) [1], [2]. The main property of interoperability is the seamless information integration of individual elements and systems as a whole. The relevance of ensuring interoperability increases significantly with the transition of all areas of ICT application (Fig. 1) to the stage of digital transformation.

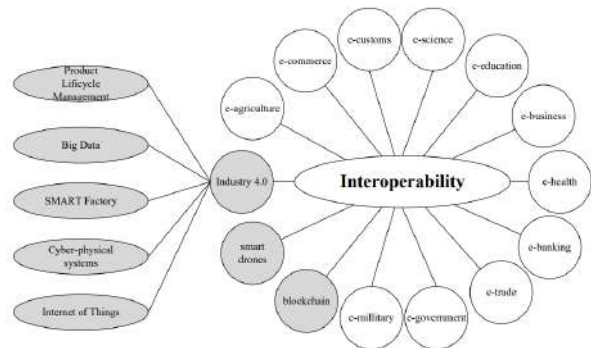


Figure 1. Interoperability — a key requirement for different applications.

All these domain areas should be seen as subsystems of the information society, and not isolated, but closely interacting. As it follows from the figure, the problem of interoperability extends to such subsystems of the information society as e-government, e-science, e-health, e-education, e-business and other areas, since virtually no area of knowledge and economy can develop today without the use of information and telecommunication technologies. The process integration of socio-electronic systems into the information society is carried out with the help of the Industry 4.0 platform. The provision of interoperability acts as one of the key factors of the industrial concept of Industry 4.0, which also includes:

- 1) Product Lifecycle Management (PLM) is the pro-

cess of managing a product at all stages, from idea, design and production to sales, service and withdrawal from the market.

- 2) Big Data is a variety of big data stored on digital media.
- 3) SMART Factory — this concept provides a flexible modular multi-platform production system with a high level of informatization and visualization, organized according to the principles of "lean manufacturing".
- 4) Cyber-physical systems is a system consisting of various physical entities of any kind, artificial subsystems, such as various sensors and sensors, and controllers, allowing to present such an entity as a whole.
- 5) The Internet of Things (IoT) is a global computing network, combining various kinds of physical objects, capable of interacting with each other and the outside world [3].

Industry 4.0 enables the creation of an efficient enterprise business model, where efficiency is achieved primarily through the rational management of automation systems for physical production operations and related processes integrated into a single information space.

Blockchain technology is crucial for the realization of a single information space with interoperability in mind. This decentralized, open-source technology allows the creation of interoperable products between blockchains, allowing more users, businesses and institutions to remain interconnected."

One new technology in industrial automation is "smart drones," which are unmanned aircraft with automatic or remote control capabilities. They are used for various purposes, including photo and video shooting, aerial photography, aerial scanning, assessment of different terrains, detection of security threats, field research, etc.

The key area of e-government development is the formation of the national information and educational environment, which should ensure the consistent implementation of digital transformation processes and their effective management. Transferability, interoperability and scalability should be the prerequisites for the successful implementation of the national information and educational environment.

When solving the problems of interoperability, organizational changes aimed at the introduction of flexible and adaptive information and education systems (IES) of different levels in educational institutions are of primary importance. The necessary flexibility in managing the process of educational activity can be provided by the process approach to its organization.

The process management of educational institution in this study is understood as an activity aimed at the implementation of business processes with the highest possible efficiency under given constraints (human, ma-

terial, immaterial, financial resources) [4].

To move to process management, it is necessary to formalize all business processes, determine which processes are the most demanded and most effective, how they are organized and how to control their effectiveness. The task of formalization is always solved by the introduction of a system of standards.

Process standardization in this case is understood as a set of measures, methods, tools and elements of the organizational structure that ensures the development, implementation, enforcement, maintenance and timely cancellation of outdated regulatory and procedural documents of the organization [5].

II. Interoperability of information systems in institutions of secondary special education

Let's consider interoperability in education on the example of a specialized secondary education institution. The activity of training specialists with specialized secondary education includes a system of business processes represented in the form of the following areas:

- *controlling* — manage the functioning of the educational system of the educational institution;
- *operational (basic)* — describe the educational process of the educational institution;
- *supporting* — serve the main activities of the educational institution (Fig. 2).



Figure 2. Business processes of secondary special educational institution.

These processes must be considered from a systemic approach, i.e., their interconnection and mutual influence on each other. The presented processes are basic, in particular educational institutions new modules can be added to them, new interrelations can be formed, so the presented scheme is open.

The implementation of the process approach must begin with the construction of a ROOT MODEL of business processes which are necessary for organizing and managing the activities of an educational institution. The ROOT MODEL of business processes is used to compile a classifier of business processes, and also shows the links between structural divisions, which allows at the output of the model to correlate business processes with structural divisions and their functions.

The scheme of the root business process model is shown in Figure 3.

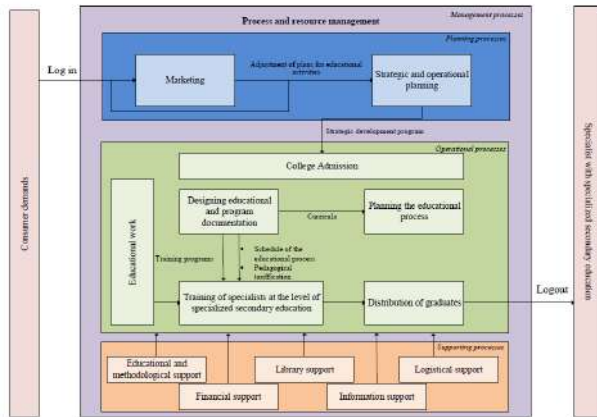


Figure 3. Root model of business processes of educational institution.

The ROOT MODEL of business processes includes three modules, each of which consists of generalized business processes and reflects their relationship through the flow of information.

The main module is operational processes. They determine the educational vector of the educational institution: the design of educational and program documentation, planning the educational process, the training of specialists at the level of secondary special education, ideological and educational work, the distribution of graduates.

The control processes module covers all business processes and is mainly focused on planning processes, whose activities are focused on providing educational services in accordance with the requirements of society and the state, which is the input data for the evaluation and subsequent adjustment of the strategic and operational plans of the educational institution.

The supporting processes module includes business processes that provide conditions for effective functioning of operational processes.

The result of the organized interconnected activity of all modules is a specialist with specialized secondary education. The root model of business processes can further serve as the basis for the classifier of business processes.

To ensure the interoperability of the information system modules it is reasonable to develop a problem-oriented interoperability model.

To date, there are many different models describing the interaction of information systems. In order to select a basic model by analogy with the approach implemented in the Russian Federation, let us distinguish four levels of interoperability (Fig. 4):

- 1) no interoperability;
- 2) technical level;
- 3) semantic level;

4) the organizational level [6].

Note that each level of interoperability should correspond to a set of standards and specifications, so that the system developers could create a profile that includes a set of harmonized standards from all the necessary levels.

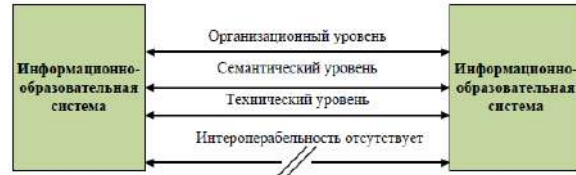


Figure 4. Basic levels of interoperability.

The first level of the interoperability model — no interoperability — means that all communication between information systems is done manually.

Thus, the college interoperability model can be based on the interoperability reference model presented in GOST R 55062-2012. This model consists of three levels: organizational, semantic, technical [7]. For each of the levels the interoperability parameters have been identified, which have been detailed to represent the college interoperability model. To form the sublevels and taking into account more parameters in the problem-oriented model of the college information system, the international experience of interoperability formalization, presented in the SCOPE-model, can be used.

As we know [8], SCOPE-model is designed for qualitative-quantitative assessment of interoperability of different aspects of the analyzed system at its different levels, according to a certain set of parameters.

Today there are many technologies for interoperability. However, these tools are used in isolation from each other and are not linked into a coherent methodological system. All the many known approaches to solving interoperability problems at the technological, semantic and organizational levels can be roughly divided into the following categories [8]:

- 1) bottom-up approach (bottom-up approach), which focuses primarily on solving the problems of technological interoperability of information systems by using common standards and technologies for transmitting, storing, representing and processing information at all levels of integration of these systems;
- 2) top-down approach (top-down approach), which focuses on the decomposition of the solution of interoperability problems from the perspective of the system architecture as a whole, and then from the perspective of individual subsystems and processes down to atomic elements;
- 3) system-wide approach, based on the analysis of internal communications between components within

the integrated system and focused on solving the problems of interoperability by forming a single environment of information interaction between them;

- 4) the interactive approach, which takes into account the nature of coupling and interaction of different systems between themselves and the external environment and focuses on achieving interoperability of those systems and their components that already have different technological implementation and use different standards for transmission, storage, presentation and processing of information;
- 5) The process approach, focused on solving interoperability problems, taking into account the identification, analysis and optimization of the full group of technological, organizational and organizational-technical factors that trigger, during the life cycle of systems, different processes that affect the functioning of systems as a whole and the change in the property of their interoperability.

To formalize the organizational level of interoperability we use the process approach, where the delegation of authority and responsibility is carried out through business processes, where a business process is a sustainable activity, which transforms resources (which are "inputs") into results (which are "outputs") [9].

The level of organizational interoperability, taking into account the process approach, is proposed to be decomposed as follows (Fig. 5):

- 1) normative legal acts regulating organizational issues of interoperability: these are the guiding documents regulating the activities of the educational institution at the state level, as well as local documents describing the interaction of departments of the educational institution.
- 2) parameters of organizational interoperability issues and indicators of their quality:
 - organizational context;
 - organizational management structure;
 - quality management system;
 - information support;
 - management techniques;
 - image.
- 3) parameters of the life cycle of organizational systems:
 - educational results: quality of education, participation in conferences, competitions, etc., continuation of education;
 - resource provision: staffing, equipment of educational process;
 - competitiveness of educational services: number of students, additional education.
- 4) parameters of organizational readiness of system objects to interact with each other:

- personnel;
- intellectual capital.

The semantic level of interoperability formalizes the functional interaction of information systems and their elements at the level of interpretation of the meaning of the information they exchange (Fig. 5):

- 1) The concept of semantic interoperability.
- 2) Semantic interoperability of interaction.
- 3) Mode of control.
- 4) Leadership style.
- 5) Adaptability and flexibility of semantic interoperability.
- 6) Parameters for the use of terminology, linguistic expressions and knowledge in highly specialized subject areas.
- 7) Dependence of semantic interoperability on human behavior and state, as well as human psyche.
- 8) Parameters of business culture affecting semantic interoperability.
- 9) Parameters of semantic interoperability of human-machine interfaces.

The semantic level acts as a link between the information and the agents that use it. At this level through the use of ontological approach and metadata should be achieved such a degree of integration of systems, which will provide a common understanding of the data within the interacting systems [10].

The technical level of interoperability formalizes interoperability at the level of technical means, hardware and software complexes that implement information and management processes, performing operations of input, search, processing, storage, delivery and delivery of information and management tools (Fig. 6):

- 1) Data and message format compatibility parameters.
- 2) Parameters of protocol compatibility and interfaces of information exchange in the network, as well as quality of service requirements.
- 3) Communication network — unity of used protocols, formats of service messages and other means used for organization of data and commands transmission through communication channels.
- 4) Infrastructure.
- 5) Formation, retrieval, transmission, storage, processing, and presentation of information in computing complexes.
- 6) Information security parameters.
- 7) Parameters of human-machine interface ergonomics.
- 8) Parameters of technological readiness of system objects to interact with each other.

Thus, organizational processes will reflect the peculiarities of interaction of subjects on creation and application of **network-centric** systems using information resource management tools and knowledge bases at the semantic level.

The semantic level describes the concept of semantic interoperability.

Technical-technological processes reflect the features of interaction at the level of technical means and applied technologies at the technical level of interoperability.

Based on the above, we can conclude that the process approach can be used not only as a basis for formalizing the processes of the organizational level of interoperability, but also, in part, for formalizing the individual aspects at the semantic and technical levels.

The final version of the interoperability model of a secondary vocational education institution is presented in Figures 4, 5, 6.

At the organizational level of interoperability (Fig. 4) business processes and approaches to their coordination are coordinated. This is achieved through the state regulations and internal documents governing the activities of the educational institution, the quality management system. The quality management system defines the mission, goals and objectives of the organization, as well as describes the scenarios of business processes of the educational institution.

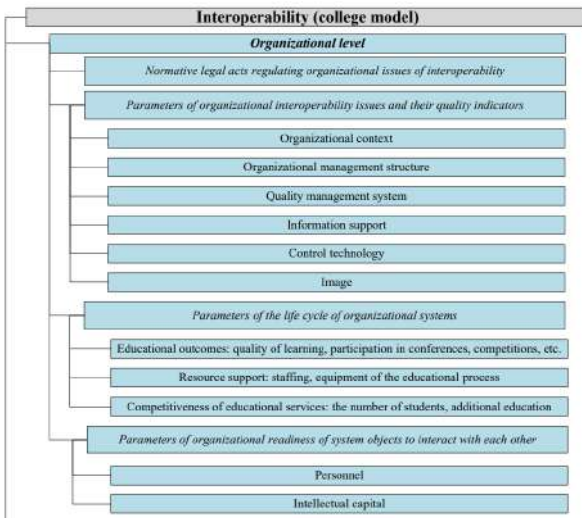


Figure 5. Organizational level of the root model of business processes of the educational institution.

Semantic interoperability allows systems to combine received information with other information resources and process its semantic content (Fig. 5). Semantic interoperability is achieved through the use of national standards and classifiers, dictionaries and directories, unified for industry and regional segments of the digital scientific and educational environment.

The technical level describes the formats of the transmitted information, where special attention is paid to how the information is presented in the communication environment (Fig. 6). The technical level includes such key aspects as: open interfaces, communication services,

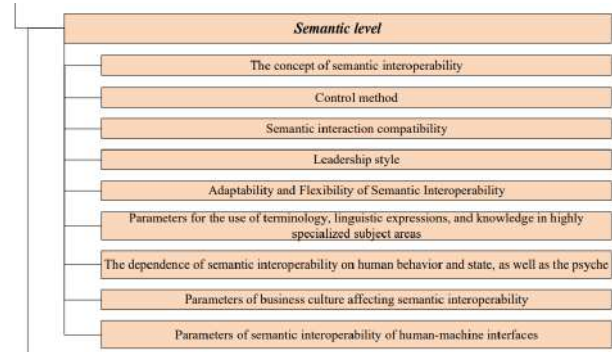


Figure 6. Semantic level of the root model of business processes of the educational institution.

information and educational environment, data integration and software middleware layer, electronic educational resources, distance learning, online repositories of electronic educational resources, automated workstations, information accessibility and protection services.

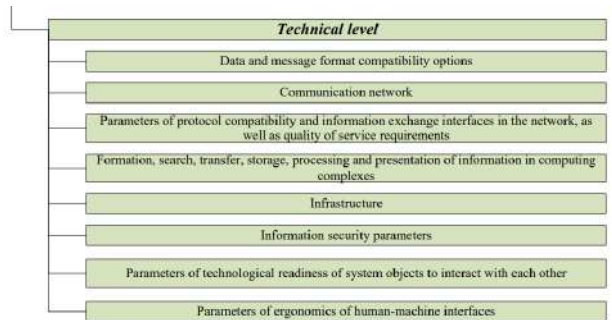


Figure 7. Technical level of the root model of business processes of the educational institution.

III. Conclusion

This article presents a three-level model of college IOS interoperability based on the processor approach, which allows you to create a flexible and adaptive management in the institution of secondary special education.

The main business processes of a secondary vocational education institution are described, the author's root model of business processes is presented, taking into account input and output data flows.

The features of the interoperability model at the organizational, semantic and technical levels are shown.

The graphical representation of the proposed model, taking into account the peculiarities of the development and implementation of the IES in the institutions of secondary special education of the Republic of Belarus, is presented.

As a recommendation for further research, it seems appropriate to develop a methodology for building an educational environment, which should include the construction of the model, an algorithm for building the

profile, its implementation and maintenance in order to develop a unified approach to ensure interoperability in educational systems.

References

- [1] ISO/IEC/IEEE 24765:2017. Systems and software engineering — Vocabulary. ISO, 2017. 522 p.
- [2] New European Interoperability Framework. Promoting seamless services and data flows for European public administrations. Luxembourg, Publications Office of the European Union, 2017. 48 p. Available at: https://ec.europa.eu/isa2/sites/isa/files/isa_eif_brochure_final.pdf (accessed 21 December 2023).
- [3] A. N. Nifant'ev Klyuchevye tekhnologii cifrovoj ekonomiki. *Magazine young scientist*, 2020, no. 50 (340). Available at: <https://moluch.ru/archive/340/76246> (accessed 10 February 2023) (in Russian).
- [4] V. V. Repin, V. G. Eliferov Protssessnyi podkhod k upravleniiu. *Modelirovanie biznes-protsessov [The process approach to management. Business process modeling]*. Moscow, Mann, Ivanov i Ferber, 2013, 544 p.
- [5] Yu. V. Gulyaev, E. E. Zhuravliov, A. Ya. Oleinkov Standardization methodology to ensure interoperability of wide class information systems. *Analytical Review. Journal of Radio Electronics*, 2012, no. 3. Available at: [text.pdf\(cplire.ru\)](http://text.pdf(cplire.ru)) (accessed 11 November 2022) (in Russian).
- [6] A. YA. Olejnikov, K. A. Ruban Modeli i standarty obespecheniya interoperabel'nosti. *Journal of Informatization of Education and Science*, 2009, no. 3. Available at: http://opensys.info/files/data_20100427173750.pdf (accessed 15 February 2023) (in Russian).
- [7] GOST R 55062-2012. Informacionnye tekhnologii (IT). Sistemy promyshlennoj avtomatizatsii i ih integratsiya. Interoperabel'nost'. Osnovnye polozheniya [Information technology (IT). Industrial automation systems and their integration. Interoperability. Fundamentals]. Moscow, Standartinform Publ., 2014. 12 p. (in Russian).
- [8] Systems, Capabilities, Operations, Programs, and Enterprises (SCOPE) Model for Interoperability Assessment. Version 1.0. NCOIC, 2008. 154 p.
- [9] S. V. Kozlov, A. N. Kubankov O napravleniyah integratsii informacionnyh, upravlyayushchih i telekommunikatsionnyh sistem na processnoy osnove [On the directions of integration of information, control and telecommunication systems on a process basis]. *Journal of Comm: Telecommunications and Transportation*, 2018, Tom 12, no. 9. Available at: <https://cyberleninka.ru/article/n/o-napravleniyah-integratsii-informatsionnyh-upravlyayuschih-i-telekommunikatsionnyh-sistem-na-protsessnoy-osnove> (accessed 2 February 2023) (in Russian).
- [10] E. V. Frangulova Klassifikatsiya podhodov k integratsii i interoperabel'nosti informacionnyh sistem [Classification of approaches to integration and interoperability of information systems]. *Journal of Bulletin of Astrakhan State Technical University*, 2010, no.2. Available at: <https://www.mathnet.ru/links/3394e634886fe20a86acc3065eaf\protect\discretionary{\protect\protect\leavevmode@ifvmode\kern+.1667em\relax\OMS/cmsy/m/n/8\char2}{}}78c9/vagu201.pdf> (accessed 24 February 2023) (in Russian).

Модель интероперабельности информационных систем информационно–коммуникационной среды учреждения среднего специального образования

Листопад Н. И., Бущик Е. А.

В работе предлагается модель интероперабельности информационных систем информационно–коммуникационной среды учреждения среднего специального образования. Показано, что следует понимать под интероперабельностью информационных систем применительно к отрасли образования на примере учреждения среднего специального образования. Представлена система бизнес–процессов в учреждении среднего специального образования в виде следующих направлений: управляющие, операционные (основные, описывающие образовательный процесс учреждения образования) и поддерживающие. Приведена авторская корневая модель бизнес–процессов учреждения образования. Известная эталонная модель интероперабельности в виде трех уровней — организационного, семантического и технического — уточнена и дополнена параметрами, которые были детализированы для колледжа. Трехуровневая модель интероперабельности информационных систем колледжа уточнена на основании процессорного подхода, позволившего создать гибкое и адаптивное управление в учреждении среднего специального образования.

Received 16.03.2023

About Creation of the Intelligent Transportation Control System in Railway Transport

Aleksandr Erofeev

Belarusian State University of Transport

Gomel, Belarus

Email: erofeev_aa@bsut.by

Abstract—The relevance of the development of an intelligent system for managing the transportation process is determined. The structure of the system construction theory is given. The experience of developing automated systems on the Belarusian railway is described and the effectiveness of their implementation is evaluated. It has been established that the main condition for the interaction of automated systems with each other is the use of a single ontology of the transportation process. It is indicated that the OSTIS technology is an effective tool for describing the process-object ontology of the transportation process. The advantages and limitations of using OSTIS technology in ITCS are established.

Keywords—Intelligent Transportation Control System, Ontology, process-object approach, transportation process, OSTIS technology

Automation of individual tasks of managing the transportation process (TP) was one of the first areas of informatization of the railway transport activity. However, in modern conditions, the efficiency of previously developed automated systems (AS) has decreased due to significant fluctuations in the power and structure of traffic flows and changes in the technologies of the transportation process. Further development of existing AS has significant limitations: the exact mathematical model of the object may be too complex or unconstructible; changes in the external object environment lead to the action on the object of a number of perturbations, which are an additional source of uncertainty about the state of the object; performance requirements can be loosely formalized and inconsistent. It is proposed to overcome these shortcomings by moving from information-reference and settlement systems to intellectual ones.

BeLSUT has developed a theory for building an Intelligent Transportation Control System (ITCS), the use of which in the development, implementation and operation will increase the adaptability of transportation process technologies to a changing operational environment, solve new operational problems, ensure coordination and continuity of control decisions, improve system manageability, which Together, it will ensure the efficient functioning of the railway in the face of changes in the volume and structure of traffic flows and optimize the operating costs for the organization of transportation activities [1], [2].

A graphical interpretation of the methodology for creating an ITCS is shown in “Fig. 1”.

The creation of the ITCS is aimed at:

- implementation of a coordinated integrated transportation process management system (TPMS) using by all participants in this activity a single digital model of the transportation process (DMTP), which describes the transport processes, covering the activities of all involved departments and all levels of management;
- improving the quality of information in the TPMS;
- formation of services for operational information and technological interaction of participants in the transportation process within the framework of a single long-term, medium-term, shift-daily and current planning, execution and control of agreed and approved plans;
- implementation of adaptive automatic control of technological processes for operational work and control over the execution of control decisions (CD);
- operational step-by-step and process assessment of CD.

The functioning of the ITCS is aimed at improving the efficiency of the TP by:

- increasing the speed of traffic flows;
- reducing the turnaround time of the wagon, including by reducing the time spent by the wagon in an empty state;
- reducing operating costs, including by increasing the productivity of locomotives in freight traffic, increasing the efficiency of using the car fleet;
- implementation of rational options for passing train flows in a changing operational environment;
- reducing the number of overtimes at technical stations during the turnover of a freight car and increasing the transit capacity of car traffic.

Currently, the following functional modules of the ITCS have been implemented or are being implemented at the Belarusian Railways. The AS “Graphist” software package (“Fig. 2”) allows you to develop train schedules (DTS) [3], [4]. Currently, it is in commercial operation at the Transportation Control Center of the Belarusian Railways.

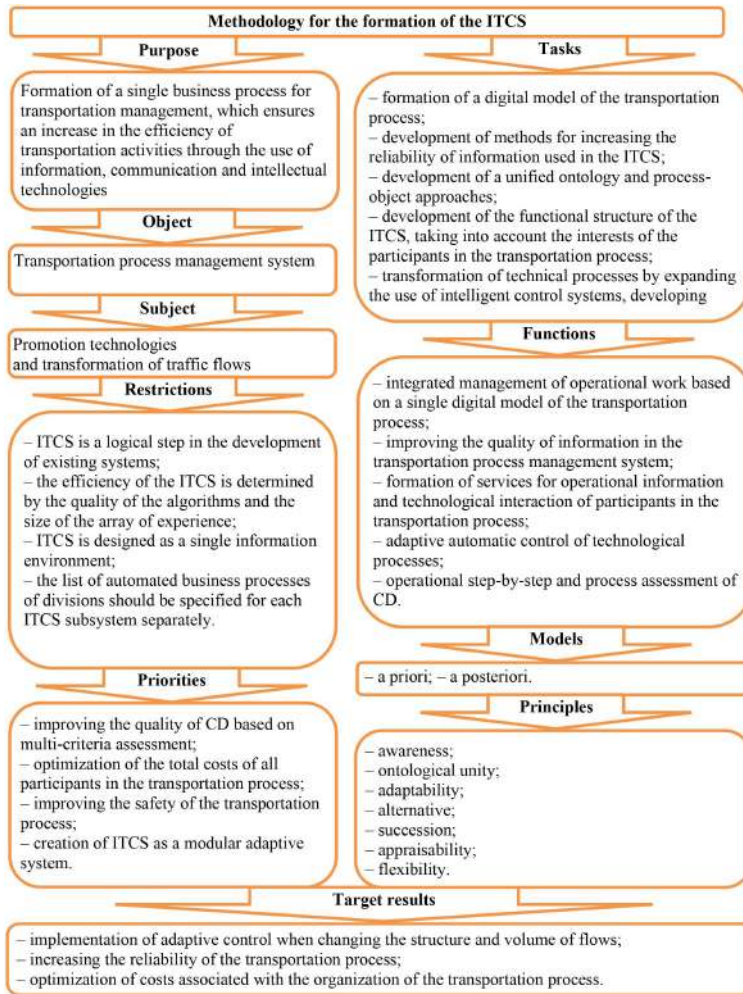


Figure 1. Graphical interpretation of the methodology for the formation of ITCS.

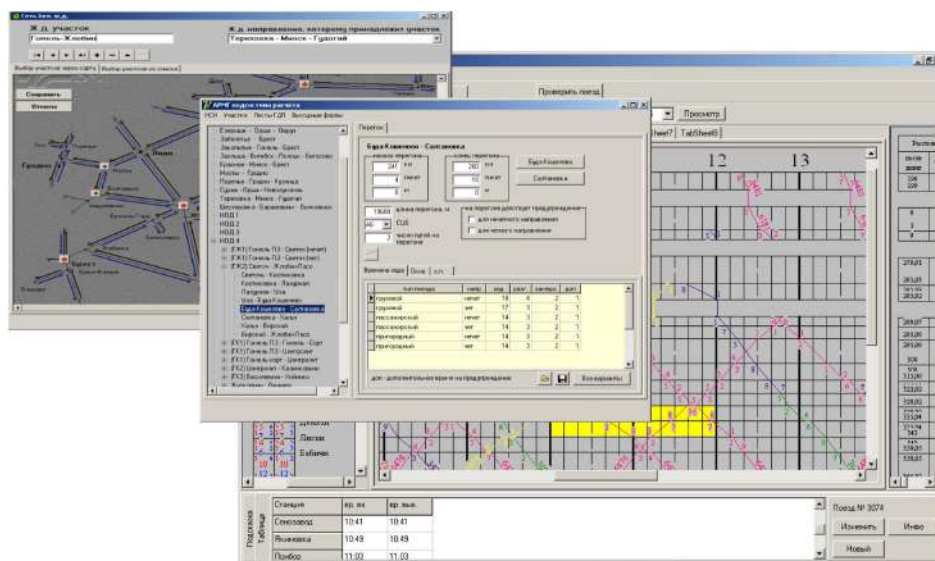


Figure 2. Modules for automatic construction of the DTS and its adjustment in AS "Graphist".

The intelligent algorithm for the development of the DTS is designed in such a way that, depending on the relative position of trains in the DTS and their categories, determine which of the station intervals should be used in each specific case. The calculation is made taking into account the mutual influence of the stations of the section. The solution of the problem is envisaged at the range of any length and configuration. Intellectualization of the functions of the development of the DTS allowed to reduce the workload of engineering personnel by 20-30%. The introduction of AS "Graphist" made it possible to increase the sectional speed by 7-11.5% in some sections and reduce the specific energy costs for train traction by 3-6%.

An automated system for shift-daily scheduling of cargo work (AS SDS) has been developed and put into commercial operation, which for the first time in the world provides end-to-end scheduling of railway freight work for the entire polygon of the road, all levels of management (road, departmental, linear) and all planning periods [5].

The AS SDS ("Fig. 3") implements the functions of intelligently linking wagons to requests (taking into account their condition, location, expiration category, owner and other features), as well as other elements of intelligent technologies: forecasting the time of arrival of wagons at the station, adjusting planned indicators for the second shift depending on their implementation for the first; formation of plans taking into account the directive establishment of an increased task for loading, etc.

Based on the results of the operation of the AS SDS, it was found that by improving the accuracy of planning, the share of unscheduled loading decreased by 20-30%. For the first time, a system of number-wise planning of cargo work with high planning accuracy (91-94%) has been implemented. The use of intelligent technologies in the planning system made it possible to increase the ratio of double operations by 8-12%; reduce the downtime of a local car at individual freight stations of the station by 6-9%.

An automated system for linking train formation with a train schedule (LTFDTS) has been developed and put into commercial operation ("Fig. 4") [6].

The main output decisions are: the schedule for the departure of freight trains from train stations for the forecast period; a plan for processing trains at train stations during the forecast period; dislocation of trains and wagons at train stations at the end of the forecast period. An intelligent solution of LTFDTS is also an abbreviated predictive DTS, in which, by means of multifactorial selection, all trains participating in train formation are linked to the threads of the predictive DTS.

Intellectualization of the train formation planning process at the Belarusian Railway range using LTFDTS made it possible to increase the efficiency of dispatching

control by increasing the reliability and automating the development of a predictive train schedule with further use in the automated train traffic control system (autodispatcher). This made it possible to enlarge the ranges of train traffic control by 1.3-1.5 times and optimize costs; reduce the time spent by trains and locomotives at technical stations by reducing the waiting time of technological operations from 15 to 20 percent; to ensure the coordination of the predictive DTS with the train and locomotive model of the road, reducing non-production losses of locomotive crews up to 20 percent.

Various development companies participated in the creation of these and a number of other systems. One key condition for their creation was to ensure the exchange of CD between different systems of the ITCS. For these purposes, a unified ontology of the transportation process was formalized.

The ontology of the transportation process presupposes the existence of unified ways of describing the system and the processes occurring in it. This task is inextricably linked with the formation of a digital model of the transportation process (DMTP). Actual mechanisms for the formation of the CMPP should allow for the real-time simulation of the state of the TP. This requires the unification of requirements for the content and form of presentation of information about the parameters of the functioning of objects [7].

DMTP may include:

- 1) models of objects (including resources) of the TPMS;
- 2) process models – a description of the processes occurring both in the TPMS and in the external object environment;
- 3) models of the external object environment, describing the external impact on the objects of the transportation process;
- 4) situation forecasting models – the study of options for the development of the transport situation in case of emergency changes in the state of the elements of the transport system, the external environment, with changes in the characteristics of information flows;
- 5) CD formation models that provide an analysis of the operational environment and the formation of effective CD;
- 6) assessment models that provide an assessment of the effectiveness of the implemented CD, the state of objects and the parameters of the software.

CMPP is focused on the implementation of dual control, i.e. adaptive control, in which not only the goals are achieved, but also the model is refined.

All DMTP objects are divided into the following subgroups:

- static objects (infrastructure objects);
- dynamic objects (tracking objects) ("Tab. 1").

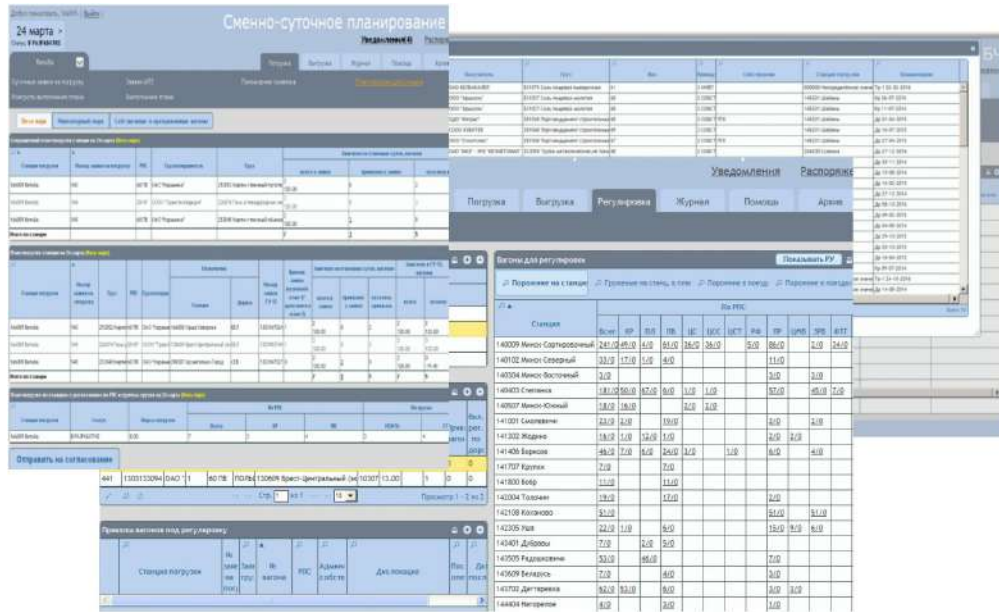


Figure 3. Modules for intelligent planning, linking applications to wagons and monitoring the implementation of plans in AS SDS.

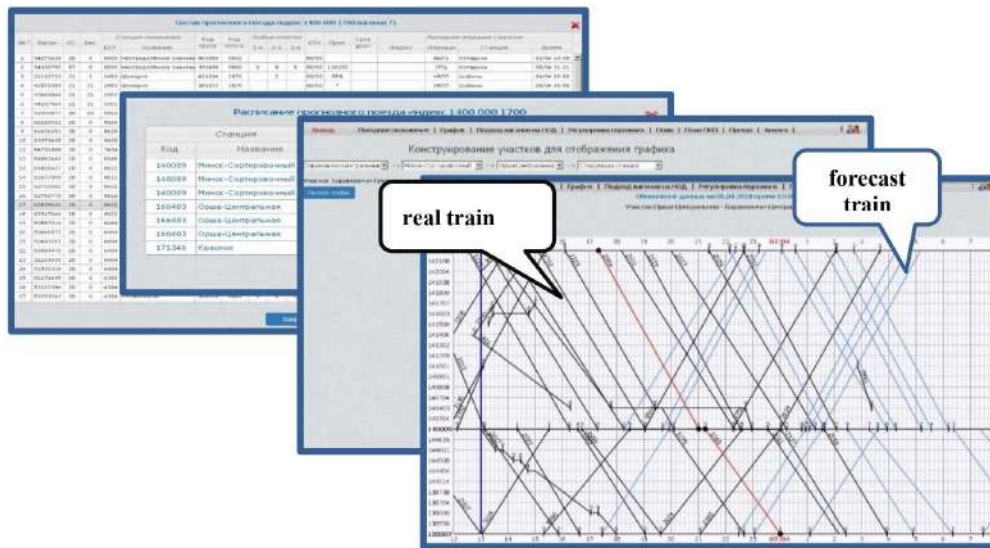


Figure 4. Modules of intelligent planning of composition formation and predictive DTS in LTFDTS.

Table I
DMTP FACILITIES

static objects (infrastructure objects)	dynamic objects (tracking objects)
<ul style="list-style-type: none"> Railway; – department; – railroad station; – interstate crossing point; – control area; – control room; – railway direction; – railway section; – haul; – block section; – depo. 	<ul style="list-style-type: none"> – train; – railway carriage; – container; – sending; – locomotive; – brigade; – document.

Based on the digital transformation of the logical entities of infrastructure objects, an ontology of static objects is formed. Each logical entity consists of several connected objects, which in the database are divided into the following sub-schemes (groups of tables): railway, departments, railway stations, railway sections, depots and auxiliary tables. This subgroup of modeling objects also includes such logical entities of the TP as a train schedule, a train formation plan, a local cargo delivery scheme, etc.

The given subschemes consist of a number of separate interconnected tables with a description of elementary (topographic) objects. For example, the digital model

of the "railway station" object includes the following elementary objects:

- paths;
- turnouts;
- traffic lights;
- passenger platforms;
- building;
- artificial constructions;
- subgrade, etc.

Based on the digital transformation of logical entities such as "train", "car", "container", "dispatch", "locomotive", "document", an ontology of dynamic TP objects is formed. Each logical entity consists of several logically connected objects. In accordance with this, the set of dynamic objects is logically divided into the following subschemes: trains, wagons, containers, shipments, locomotives, crews. When formalizing an ontology in the form of a database, the set of dynamic objects is supplemented by documents and auxiliary tables that describe the properties of each object.

Tables of descriptions of control objects are logically interconnected. Information about each tracking object is entered into the corresponding logical subcircuit, which consists of several levels. Each database table of the next level is a child of the table of the previous level.

Modeling of infrastructure objects and vehicles for solving operational problems of the ITCS is based on semantic and ontological links between these TP objects.

In the ontological scheme of TP, objects are interconnected within their group. The "station" object is an element (attribute) of higher-level objects – "department" and "railway section". In turn, the "railway section" is an element of the "department" object, and its attributes are determined based on the parameters of the station object. The topmost level in the group of static objects is the "railway".

The positioning parameters of dynamic objects (objects of tracking) are connected with each other and with elementary infrastructure objects. Depending on the stage of the SP, the parameters of the tracking objects are combined into one top-level tracking object, or vice versa, the top-level object is divided into lower-level objects. After the loading process, the "dispatch" object is included in the temporary group of the "wagon" object. The technological lifetime of this group ends after the unloading operation is completed. A similar procedure is performed when transforming the objects "car" and "locomotive" into the top-level object "train".

Traditionally, AS are focused on collecting, aggregating and presenting information to a person. A distinctive feature of the ISMS should be its focus on solving operational problems (problem orientation).

The object-oriented approach is based on the representation of each object of the software domain in the form of a classifier and its description by a set of

properties – characteristics. To describe the relationship between objects, a special unit of data is used – a relation. Combinations of these elements form models of objects and situations. The classifier is a set of initial units of information (concepts of the selected subject area) systematized according to classification criteria and their groupings, representing generalized concepts. Creation of a model of objects of the required subject area of TP allows you to adapt the basic knowledge of the ITCS to solve the necessary ET.

The process-oriented approach is based on the formation of a model for the execution of technological operations of the software, aimed at achieving the ultimate goal of solving the operational and (or) completing the solution of the problem within the time limits established by the technology. A process is generally understood as "a set of interrelated and interacting activities that transforms inputs into outputs."

To describe the TPMS, a process-object ontology of TP is proposed as a symbiosis of ontological description, object-oriented and process-oriented approaches [1].

An effective tool for describing the process-object ontology of TP, in our opinion, is the OSTIS technology. This choice is due to the following important properties of the technology [8]–[10].

- 1) OSTIS technology is a technology of component (modular) and platform-independent design of semantically compatible intelligent systems that have knowledge bases of any complexity and implement parallel models of knowledge processing. This allows you to effectively integrate systems of various developers with each other and feel complex CD.
- 2) The knowledge base of an ostis-system, i.e., a system built using the OSTIS Technology, is a semantic network that generally has a complex hierarchical structure, in which there are elements denoting not only external entities and relationships between them, but also various classes elements of the semantic network, various fragments of this network, various connections between the indicated classes, between the indicated fragments. All this provides unlimited opportunities for the transition from knowledge to metaknowledge. This approach makes it possible to ensure the solution of new ET, including those based on the formation of CD under conditions of uncertainty in the initial data.
- 3) The basis of parallel models of knowledge processing in systems built using OSTIS Technology is the model of asynchronous knowledge management. The essence of this model is that all (!) knowledge processing processes performed by a certain set of agents are initiated by the corresponding situations or events that occur in the semantic memory during the processing of the knowledge base. At the same

time, any computer system, incl. and the one that does not solve intellectual problems can be built on the basis of the OSTIS Technology, i.e., in the form of an ostis-system. This property makes it possible to develop IS through the phased implementation of intelligent functions in automated systems.

However, the use of OSTIS Technology has significant limitations:

- a relatively small number of developers own this technology, and those companies that know this technology do not have the competence in building management systems in transport;
- the study of the OSTIS Technology is quite problematic, since a single description standard has not been approved.

The Republic of Belarus has the conditions to take the lead positions in the world in the development of intelligent control systems for railway transport:

- there is a team of scientists-experts who have been dealing with the issues of informatization of transport processes for decades. Most experts work at BelSUT or are its graduates;
- scientific schools in the field of artificial intelligence (BSUIR, BSU, etc.) are effectively functioning in a number of universities;
- there are many software development companies with a staff of highly qualified programmers, most of which are concentrated in the Hi-Tech Park.

In this regard, an important task is the formation of scientific and practical consortiums that have the necessary knowledge and competencies to build an ITCS using OSTIS technology.

REFERENCES

- [1] A. A. Erofeev *Intellektual'naya sistema upravleniya perevozhnym processom [Intelligent transportation process control system]*. Monografiya, Gomel', UO BelGUT, 2022, P. 407 (In Russ.)
- [2] P. M. Dulub, V. G. Kuznecov *Adaptaciya innovacionnyh tekhnologij v organizacii perevozhnogo processa belorusskoj zheleznoj dorogi [Adaptation of innovative technologies in the organization of the transportation process of the Belarusian railway]*. *Vestnik Belorusskogo gosudarstvennogo universiteta transporta: nauka i transport [Bulletin of the Belarusian State University of Transport: science and transport]*, 2015, No. 2 (31), pp. 12–17 (In Russ.)
- [3] A. A. Erofeev, E. A. Erofeeva *Sistema avtomatizirovannogo proektirovaniya grafika dvizheniya poezdov [Train schedule computer-aided design system]*. Ucheb.-metod. posobie. M-vo obrazovaniya Resp. Belarus', Belarus. gos. un-t transp, Gomel': UO BelGUT, 2012, P. 150 (In Russ.)
- [4] V. G. Kuznecov, O. A. Tereshchenko, E. A. Fedorov, N. A. Starinskaya, S. E. Sakovich *Avtomatizirovannaya sistema smennosutochnogo planirovaniya poezdnoj i gruzovoj raboty belorusskoj zheleznoj dorogi [Automated system for shift-daily planning of train and freight work of the Belarusian railway]*. *Vestnik Belorusskogo gosudarstvennogo universiteta transporta: nauka i transport [Bulletin of the Belarusian State University of Transport: science and transport]*, 2015, No. 2 (31), pp. 53–56 (In Russ.)

- [5] V. G. Kuznecov, T. V. Pil'gun, O. A. Tereshchenko, E. A. Fedorov *Sozdanie informacionno-analiticheskoy modeli smennosutochnogo planirovaniya ekspluatacionnoj raboty zheleznoj dorogi [Creation of an information-analytical model of shift-daily planning of the operational work of the railway]*. *Informacionnye tekhnologii i sistemy 2016 (ITS 2016) : materialy Mezhdunar. nauch. konf., Minsk, 28 okt. 2015 g.*. L. YU. SHilin (eds.). Minsk, BGUIR, 2016, pp. 54–55 (In Russ.)
- [6] O. A. Tereshchenko *Information Support for Operational Management Technology of Railway Local Work. Problemy bezopasnosti na transporte [Transport security issues]. Materialy hi mezhdunarodnoj nauchno-prakticheskoy konferencii. V 2 chastyah, 2021, Gomel', pp. 64–65*
- [7] V. G. Kuznecov, E. A. Fedorov, V. G. Kozlov, S. V. Grigor'ev *Cifrovaya model' gis-tekhnologij dlya resheniya zadach operativnogo upravleniya perevozhnym processom [Digital model of GIS technologies for solving the problems of operational management of the transportation process]*. *Vestnik Belorusskogo gosudarstvennogo universiteta transporta: Nauka i transport [Bulletin of the Belarusian State University of Transport: Science and Transport]*, 2018, No. 2 (37), pp. 66–70 (In Russ.)
- [8] V. V. Golenkov, N. A. Guliakina *Printsipy postroeniya massovoi semanticheskoi tekhnologii komponentnogo proektirovaniya intellektual'nykh sistem [Principles of building mass semantic technology component design of intelligent systems]*. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, Minsk, BSUIR, 2011, pp. 21–58 (In Russ.)
- [9] V. V. Golenkov, N. A. Guliakina *Semanticheskaya tekhnologiya komponentnogo proektirovaniya sistem, upravlyayemyh znaniyami [Semantic technology of component design of knowledge-driven systems]*. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, Minsk, BSUIR, 2015, pp. 57–78 (In Russ.)
- [10] V. V. Golenkov *Ontology-based design of intelligent systems. Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, Minsk, BSUIR, 2017, pp. 37–56

О создании интеллектуальной системы управления перевозками на железнодорожном транспорте

Ерофеев А. А.

Показана актуальность разработки интеллектуальной системы управления перевозочным процессом. Представлена структура теории построения систем. Описан опыт разработки автоматизированных систем на Белорусской железной дороге и оценена эффективность их внедрения. Установлено, что основным условием взаимодействия автоматизированных систем между собой является использование единой онтологии перевозочного процесса. Показано, что Технология OSTIS является эффективным инструментом описания процессно-объектной онтологии транспортного процесса. Установлены преимущества и ограничения использования Технологии OSTIS в ИТКС.

Received 13.03.2023

Adaptive Control System for Technological Process within OSTIS Ecosystem

Valery Taberko and Dzmitry Ivaniuk

JSC «Savushkin Product»

Brest, Belarus

Email: tab@pda.savushkin.by,

id@pda.savushkin.by

Viktor Smorodin and Vladislav Prokhorenko

Department of Mathematical Problems

of Control and Informatics

Francisk Skorina Gomel State University

Gomel, Belarus

Email: smorodin@gsu.by, snysc@mail.ru

Abstract—In this paper an approach to building a hybrid intellectual computer system for adaptive control of a technological production cycle is being proposed in the form of an ostis-system solver based on the ontology of the "technological production processes with probabilistic attributes" domain knowledge. The idea of development and implementation of mathematical models of neural network regulators for control optimization problems is the basis for the system solver. Such an implementation makes it possible to integrate the proposed solution with other developed solutions as well as the company's software in order to allow building intellectual systems for automated control, recommendation systems and decision support systems, information support systems for the company's personnel.

Keywords—technological production process, adaptive control, neural network, reinforcement learning, Industry 4.0, standard

I. INTRODUCTION

The constant efforts of scientific and technological advances in the sphere of optimization of production systems operation require the development of up-to-date approaches to adaptive control of production processes that would include elements of artificial intelligence, neural network modeling and development of intellectual computer systems of new generation.

In this paper an approach is being proposed for building a hybrid intellectual computer system for adaptive control of a technological production cycle within the framework of OSTIS Ecosystem. The formal description of the control object is based on ontologies of the "technological production processes with probabilistic attributes" domain knowledge and ISA 5.1, ISA-88, ISA-95 standards, implemented through the means of OSTIS Technology.

Control adaptation is implemented using the means of software-hardware coupling of neural network regulators and hardware control system of the technological object in real-time.

The idea of the hybrid intellectual system for adaptive control is based upon development and implementation of mathematical models of neural network regulators for solving problems of control optimization, implementation of methods and algorithms of synthesis of feedback

control for technological cycle, depending on changing parameters of the control object operation.

II. TERMS AND DEFINITIONS

The central concepts within the knowledge domain under consideration are technological process (technological cycle) and probabilistic technological process [2].

- 1) A technological process (TP) (technological cycle) of production is a sequence of interconnected operations, that is defined by technological documentation, directed at the object of process with purpose of producing the required output. Within ISA-88 standard's procedural control module TP is defined as a procedure, that produces a batch of product (that may be a final product or intermediate product used in the further stages of production of the final product).
- 2) A technological operation is part of a technological process, that is being run continuously at one workplace on one or more simultaneously processed or assembled products by one or more operators. Within ISA-88 standard it is defined as operation that results in substantially altered product properties. Operations can be run by an operator who can start, pause and resume them. At each moment of time only one operation can be active within a machine (unit).
- 3) Microtechnological operation is a finite sequence of elementary operations which constituent the contents of a technological operation, that is run continuously at one workplace. Within ISA-88 standard it is a phase that results in small changes in the properties of a product. Those steps can be run in parallel, in sequence or as a combination of two. Operator can't directly control the steps (start/pause/resume).
- 4) A Probabilistic technological process is a technological process that has probabilistic parameters of operation; a technological process with a structure that may change during its operation.

- 5) A Control system is a well-defined set of hardware-software means for control of a technological object, that makes it possible to collect readings of its state and to influence its operation in order to achieve the given goals.
- 6) Adaptive control is a set of methods and algorithms that allow synthesizing the control feedback connections that can change parameters (control structure) of the neuroregulator based on the control actions and external disturbances.
- 7) Automated control system of a technological process (ACSTP) is a complex of technical and software means that makes it possible for technological units to operate in automated mode based upon the chosen control criteria.

technological process

- := [technological cycle]
- := [is a sequence of interconnected operations, that is defined by technological documentation, directed at the object of process with purpose of producing the required output]
- := [a set of technological operations $\{TCO_{ij}\}$, where $i, j = \overline{1, N}$, as well as the resources consumed by those operations]

probabilistic technological process

- := [PTP]
- \subset *technological process*
- := [technological process that has probabilistic parameters of operation]
- := [a technological process with a structure that may change during its operation]

technological operation

- := [TCO]
- \subset *technological process*
- := [subset of a technological process, that is being run continuously at one workplace on one or more simultaneously processed or assembled products by one or more operators]

microtechnological operation

- := [MTCO]
- \subset *technological operation*
- := [finite sequence of elementary operations which constituent the contents of a technological operation, that is run continuously at one workplace]

III. PROBLEMS OF ADAPTIVE CONTROL OF PRODUCTION PROCESSES IN THE CONTEXT OF NEW GENERATION COMPUTER SYSTEMS DEVELOPMENT

Enterprise automation tools must be able to adapt to any changes in the production process itself with minimal costs and time delays. Such changes may include expansion or reduction in production volumes, changes in production nomenclature, replacement or changes of technological units, alteration of the overall production structure, changes of interactions between suppliers and consumers, changes of the legal acts and standards, as well as other unforeseen circumstances of different nature.

Analysis of the field in the sphere of the modern controlled production systems research demonstrates that the problem of determining the real-time operation parameters of such research objects emerges primarily in the cases of complex technical products production that requires high manufacturing precision and high labor productivity.

In such cases when solving a multicriteria control optimization problem high standards need to be applied to the algorithms of the operation of the production process, minimization of human factor impact on the technological production cycle operation quality, prevention of occurrence of technogenic emergencies. Such a case is typical for robotic production systems that operate under control of the hardware and software controller that administers the functioning of the technological cycle control system according to the given programs.

At the same time the arising emergency situations due to equipment failures, random external control actions, including human factor, lead to deviation of the operation parameters of the production system from the desired values, which leads to the necessity of their correction in real-time based on the neuroregulator models that operate within the hardware-software coupling means of the technological production cycle.

The existing special artificial intelligence models such as neural networks have unique properties and can be used as universal approximators that have a capability to generalize the data against which they were trained. Such features make it practical to use such models when solving complex problems in the domain of adaptive control.

The modern convergence tendencies in the sphere of intellectual systems development [1] requires the development of the appropriate software that would feature elements of cognitive abilities based on semantically compatible technologies of artificial intelligence. This sphere also includes the development of computer systems that are able to provide intellectualization of the processes of making analytical control decisions (which are directly related to adapting control processes for complex dynamic systems (technological objects) in real-time), building semantically compatible knowledge bases in the domain

of dynamic systems operation analysis and optimization of the operation of complex control systems based upon them through open-source intellectual decision support systems development.

IV. BUILDING A MATHEMATICAL MODEL OF A PRODUCTION SYSTEM IN THE CONTEXT OF INDUSTRY 4.0

Implementation of Industry 4.0 concept at industrial enterprises is accompanied by creation of a unified ontological model of the production process. This model is the core of integrated information service for the company [13].

In order to allow a wide implementation of the artificial intelligence technologies for automating the company, all corporate knowledge must be transformed into the formal language for knowledge representation. Such knowledge may be obtained from the existing documentation that describes the enterprises' operation within the framework of accepted international standards.

It is necessary to transform each existing standard into a knowledge base based on the appropriate set of ontologies related to the standard. Such an approach allows to significantly automate the processes of standard development and its application.

As an example let us consider *ISA-88* standard [16] (the base standard for prescription production). Even though this standard is widely used by American and European companies (and is being actively introduced in Republic of Belarus) it has a number of flaws that are listed below. Authors' experiences with *ISA-88* and *ISA-95* have demonstrated the following problems related to versions of the standard (see [8]):

- 1) American version of standard – *ANSI/ISA-88.00.01-2010* – has been updated and is in its 3rd edition (as of 2010);
- 2) *ISA-88.00.02-2001* – includes data structures and guidelines of languages;
- 3) *ANSI/ISA-TR88.00.02-2015* – describes an implementation example of *ANSI/ISA-88.00.01*;
- 4) *ISA-88.00.03-2003* – an activity that describes the use of common site recipes with and across companies;
- 5) *ISA-TR88.0.03-1996* – shows possible recipe procedure presentation formats;
- 6) *ANSI/ISA-88.00.04-2006* – structure for the batch production records;
- 7) *ISA-TR88.95.01-2008* – describes how *ISA-88* and *ISA-95* can be used together;
- 8) At the same time, the European version approved in 1997 – *IEC 61512-1* – is based on the older version *ISA-88.01-1995*;
- 9) Russian version of the standard – *GOST R IEC 61512-1-2016* – is identical to *IEC 61512-1*, that is also outdated. Also it raises a number of questions

related to the not very successful translation of the original English terms into Russian.

Another standard that is often used in the context of Industry 4.0 is *ISA-95* [17]. *ISA-95* is an industry standard for describing high level control systems. Its main purpose is to simplify the development of such systems, abstract from the hardware and provide a single interface to interact with ERP and MES layers. It consists of the following parts (see [8]):

- 1) *ANSI/ISA-95.00.01-2000*, Part 1: «Models and terminology» — it consists of standard terminology and object models that can be used to determine what information is exchanged;
- 2) *ANSI/ISA-95.00.02-2001*, Part 2: «Object Model Attributes» — it consists of attributes for each object defined in Part 1. Objects and attributes can be used to exchange information between different systems and can also be used as the basis for relational databases;
- 3) *ANSI/ISA-95.00.03-2005*, Part 3: «Models of Manufacturing Operations Management» — it focuses on Level 3 (Production/MES) functions and activities;
- 4) *ISA-95.00.04* Part 4: «Object models and attributes for Manufacturing Operations Management». The SP95 committee is yet developing this part of *ISA-95*. This technical specification defines an object model that determines the information exchanged between MES Activities (defined in Part 3 of *ISA-95*). The model and attributes of Part 4 form the basis for the design and implementation of interface standards, ensuring a flexible flow of cooperation and information exchange between various MES activities;
- 5) *ISA-95.00.05* Part 5: «Business to manufacturing transaction (B2M transactions)». Part 5 of *ISA-95* is still in development. This technical specification defines operations among workplace and manufacturing automation structures that may be used along with Part 1 and Part 2 item models. Operations join and arrange the manufacturing items and activities described within the preceding a part of the standard. Such operations arise in any respect ranges within the organization, however the attention of this technical specification is at the interface among the organization and the management system;
- 6) *ISA-95.00.06* Part 6: «Transactions between Manufacturing Operations»;
- 7) *ISA-95.00.07* Part 7: «Model of Service Messages»;
- 8) *ISA-95.00.08* Part 8: «Profiles of Information Exchange».

Models help define boundaries between business and control systems. They help answer questions about which functions can perform which tasks and what information must be exchanged between applications.

The first phase of building a digital twin model requires

embedding data at lower levels of production, such as production processes and equipment (see [15]). The connection with other software components that solve different tasks (acquiring, saving, displaying, analyzing data, etc.) should also be taken into account, but in general they form a digital twin (see [23]). The P&ID production scheme serves as the source of this data. Therefore the ISA 5.1 standard [18] has to work with the P&ID scheme and is widely used in control systems along with the ISA 88 standard to fully describe the lower production levels. This standard is useful when a reference to equipment is required in the chemical, petroleum, power generation, air conditioning, metal refining, and many other industries. The purpose of this standard is to establish a consistent method of naming instruments and instrumentation systems used for measurement and control. For this purpose, a designation system is presented that includes symbols and identification codes.

One of the important problems when implementing a standard at a company is the possibility of ambiguous interpretation of various parts of standard, as well as the necessity to constantly alter and improve this interpretation in such a way that it becomes closer to original. Moreover, there are specific features of implementing a standard at a particular company, the necessity to update the standard that is being used (because each standard constantly evolves) followed by changes to structure and organization of the company's activities to ensure compliance with the standard.

In order to build a mathematical model of a production system it is required to formalize the technological processes of this production system. The formalization of the mathematical models of the object under study and the control contour is based upon the results of authors' research in the area of simulating modeling of complex technical systems [2].

Traditionally two groups of technological processes are being considered:

- continuous;
- discrete.

The first group of technological processes is usually implemented during the production in real-time. These technological processes are a subject of an automated control system of a technological process (ACSTP). For example, ACSTP can be used to control the process of steelmaking, control the flow of raw materials into open-hearth furnaces and automatic pouring of metal into molds. The second group of technological processes is characterized by a graph structure of a technological cycle organization, that operates as set of interconnected operations $\{TCO_i\}$. Some TCO_i may in their turn consist of a set of microtechnological operations $\{MTCO_{ij}\}$.

Depending on how $\{TCO_i\}$ include $\{MTCO_{ij}\}$ the following types of technological processes can be identified:

- single-level processes, which consist of $\{MTCO_{ij}\}$ that can run in parallel or sequentially based on the graph structure that describes connections between $MTCO_{ij}$;
- hierarchical processes, that feature a main technological branch that is divided into several child-branches, that will then merge again; in this case as part of $MTCO_{ij}$ of any level there are operations of "splitting" of technological lines and operations of "merging" of technological lines;
- iterative, that feature child operations nested within the main operations; in this case the operation execution results at some nesting level are used to execute child technological branches.

Modeling of all types of technological processes is performed based on critical or average values of resource consumption. Probabilistic technological processes of the second and the third type are considered to be the most difficult to model.

When $MTCO_{ij}$ is running resources of the technological cycle are being spent. Based on the nature of resource consumption two groups of processes can be considered:

- deterministic processes (DTP) that have resources consumed based on critical or average values;
- probabilistic processes (PTP) that have some resources consumed in deterministic way (defined by lists of resources and their volumes) while other resources are being spent in a probabilistic way (defined by probability distribution functions for the resource consumption) by its $MTCO_{ij}$.

Some $MTCO_{ij}$ of a PTP may not only consume resources during operation but also perform control activities that change control variables U_s of the PTP.

During PTP's normal operation each component of the U_s control variables set must be within acceptable ranges of minimal (U_s^-) and maximal (U_s^+) values. These changes in U_s values may also be of probabilistic nature. Some $MTCO_{ij}$ may alter control variables values in such a way that U_s are returned into acceptable ranges ($U_s^- \leq U_s \leq U_s^+$).

Let us consider a task of modeling reliability characteristics for the technological cycle hardware.

PTP uses in its operation technological units (machines) that may have various reliability characteristics. The technological units have some resource of operation that would gradually decrease depending on the time of active usage of the given unit. When a hardware unit reaches a threshold for this resource during active operation the probability of failure increases dramatically and therefore in a real world situation it is necessary to act in a timely manner and switch to a reserve unit (if possible) or perform maintenance in order to restore the unit's resource.

The moments of time at which a unit will fail are determined by probability distribution functions for the machine's operation time before failure. Units' failures may result in emergencies of different types that require quick reactions in order to eliminate their consequences, which may lead to the expenditure of material resources and time delays.

Reliability characteristics of hardware units are defined the following way [2]:

- moments of hardware failures are defined by the probability distribution function $F(\tau_{NOw})$ of normal operation for the appropriate unit; when the value is exceeded by the operation time a failure will occur;
- if a simple failure happens normal operation can be restored after an interval (τ_{ROw});
- some failures may lead to a simple emergency with probability (P_{f1}); liquidating an emergency requires (τ_{EM1w}) additional time and (c_{EM1w}) of additional costs to cover the restoration of normal operation;
- some failures may lead to a complex emergency with probability (P_{f2}); liquidating of such an emergency requires execution of a sequence of special technological operations; each of those operation may require additional time (τ_{liq}) and additional costs (c_{liq}); therefore liquidation of such a complex emergency may lead to operation time delay of $\sum \tau_{liq}$ and increases in costs of $\sum \tau_{liq}$.

A separate type of PTP can be considered where emergency may lead to stop of operation for the whole PTP. Such emergencies are liquidated as the complex emergencies but the whole TP operation halts until the liquidation is finished.

Thus project modeling of the PTP is a complex scientific problem due to the probabilistic nature of the resource consumption by $\{MTCO_{ij}\}$ and presence of equipment failures that may also be of probabilistic nature.

V. FORMALIZATION OF THE TECHNOLOGICAL CYCLE CONTROL CONTOUR BASED ON OSTIS TECHNOLOGY

In line with the knowledge domain ontology for the probabilistic technological processes, the technological production cycle means a sequence of actions and operations that result in some final (or semi-final for this stage) product.

The functional interaction between the components of the control complex and the operating in real-time technological production cycle is based on continuous monitoring of the equipment states and the control parameters through registers-indicators and means of technical coupling.

In order to build a robust semantically compatible intellectual system for control adaptation OSTIS Technology will be used as a base for the control complex. The system includes agents to address the tasks of interacting

with the means of technical coupling and making control decisions.

In OSTIS Technology problem solvers are implemented based on multi-agent approach. According to this approach the solver is constructed as a set of agents, that are called sc-agents. All such agents share memory and may exchange data through special semantic structures (sc-texts). It is important to note that some agents may not be atomic (i.e. may consist of two or more sc-agents).

The full problems solver for the control task can be implemented as a decomposition of an abstract non-atomic sc-agent.

abstract non-atomic sc-agent of cycle recommendation system

⇒ *decomposition of abstract sc-agent**:

- {• *abstract sc-agent of interaction with the observation system*
- *abstract sc-agent of forming recommendations*
- *abstract sc-agent of forming requests*
- }

- 1) abstract sc-agent of interaction with the observation system — is used to extract observations from the means of hardware-software coupling in the technological cycle; it initiates operation of agent for forming recommendations;
- 2) abstract sc-agent of forming recommendations — based on the received observations initializes controller operation in order to get recommendation for control actions;
- 3) abstract sc-agent of forming requests — based on the data received from the agent of forming recommendations forms requests for changing the control variables of the PTP through the appropriate hardware-software means of coupling.

Agents of interaction with the observation system and forming requests must correspond with the technical system under consideration and allow communicating with the means of technical coupling.

When developing a control adaptation system within the task of building an ostis-system solver of the given structure an agent for forming recommendations based on the received observations must be created.

Construction of the controller used by the sc-agent of forming recommendations can be based on neural networks as well as other modern machine learning models and algorithms.

VI. DEVELOPING NEUROREGULATOR MODELS FOR ADAPTIVE CONTROL PROBLEMS

Several approaches may be proposed when considering the adaptive control problem:

- 1) direct control (controller modeling), when a neural network is trained on a database of existing "optimal"

signals (of an existing controller) [5] that lead to the desired trajectories in the phase space of the system and thus the neurocontroller for the control system is built [12]. Such a scheme is one of the most simple ones, but it has flaws such as a requirement to have a representative set of the existing physical controller statistics;

- 2) direct inverse control [4], in this case during modeling of the control contour operation the neurocontroller learns to reproduce a relation between the control signal at the current moment of time and observations of the control object at the next moment of time [6] [7];
- 3) Schemes may be proposed where the neurocontroller is built using reinforcement learning, particularly:
 - neurocontroller trained through reinforcement learning procedure for a task of finding an optimal trajectory (optimal in some «geometrical» sense) in the phase space of the control system states [3] [4] [9];
 - neurocontroller trained through reinforcement learning procedure for a task of implicit optimal trajectory search when some estimation of the control quality R is used [10] [11].

Since the problem of selecting an appropriate neural network structure for the particular task is in each case complex and difficult to formalize, it may be useful to consider application of the evolutionary methods for its automation. [14] .

When controller modeling is used the neurocontroller is trained in such a way that it would be able to reasonably generalize over the collected data of the control process for the technological cycle. In this case from the machine learning theory point of view, supervised learning occurs, when the collected data will be used as pairs of desired input and output (control signals) vectors of the controller. From the point of view of the dynamical systems theory a phase space of the technological process control system is an environment where a decision about control actions must be made based on the observed state. And therefore the process of neural network training may be viewed as a process of learning the correct trajectories in the phase space that correspond to optimal controls.

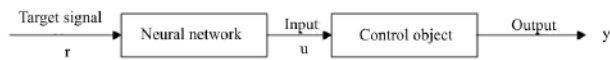


Figure 1. Direct control scheme

Training process with inverse scheme is similar from the machine learning point of view. The difference is that the desired observation signals of the control object are used as a source for building the loss function that would be optimized.

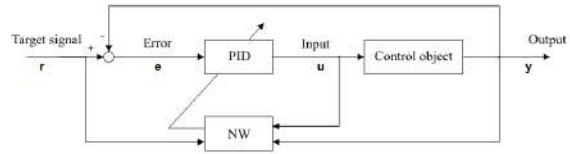


Figure 2. An example of a scheme with inverse control

The schemes where reinforcement learning is implemented have a potential for building an effective controller due to the presence of an exploratory element in them when searching for a control strategy. This may have a positive effect when solving problems with complex structure of the control decisions.

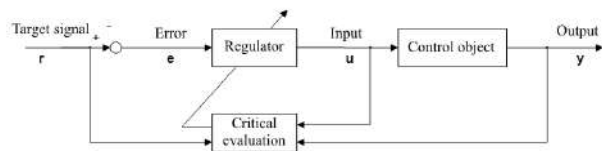


Figure 3. Reinforcement learning scheme

Let us consider some functional R for estimating the quality of the control (estimation of the quality of the control policy (control actions selection by agent (control system)) π) that is calculated during the time period of the TP operation $[0, t]$. (For example this functional may «reward» for lowering the production costs and «penalize» for equipment downtime, occurrence of hardware failures or emergencies).

Decision making in this case can be managed by a neural network (agent operating under the neural network control).

When agent performs certain actions this leads to some sort of trajectory being built in the phase space (in this context the control construction is considered to be a trajectory construction in the phase space of the technological process control system states).

The problem of search for an optimal trajectory in the phase space of the system, thus is equivalent to maximizing R (at this moment of time and at the future moments - meaning that the controller for the technological cycle is required to produce such an action selection policy π , that would maximize the estimate for control quality R).

Application of reinforcement learning methods assumes the existence of environment, in which a critical evaluation of agent's actions happens. An environment in which an agent operates in the context of the technological cycle control is the system of control of technological production cycle. This environment makes it available for agent's observations signals of registers indicating

the equipment states and control parameters. Based on the agent's decisions the decision making system forms requests for altering the control parameters.

For example reinforcement learning schemes based on Q-learning and policy gradient can be implemented to solve the task of searching for an optimal hardware maintenance strategy for the technological cycle of production [10] [11]. In this case neuroregulators are trained in a simulation that models technological cycle operation with given hardware reliability characteristics and structure, and R functional that is used to estimate model's performance is constructed in such a way that it would satisfy system user's requirements (e.g. costs minimization while avoiding serious hardware failures).

As another example to illustrate an adaptive control problem solved by neural network methods let us consider a temperature stabilization task during the operation of a pasteurizer machine. Pasteurizer is a machine for heat treatment and cooling of milk products in a continuous thin-layer closed flow. For automating the temperature regime adjustment the pasteurizer includes a control system based on an industrial controller. Quality of the final product directly depends upon the implemented algorithms.

The main reasons for temperature fluctuations t_{mn} during milk heating are flow G_m volatility of the product, temperature t_0 fluctuations in the source milk, flow G_p changes in steam that exist because of pressure P_p fluctuations, change in heat transfer coefficient K because of protein deposits on heat transfer surfaces. To stabilize the temperature t_{mn} of milk heating mainly steam flow G_p is controlled. It is controlled by a controlled valve.

A neural network based scheme with a proportional-integral-derivative (PID) regulator can be proposed to solve this task. The general structure of the self-adjusting neuro-PID regulator is shown on figure. Neural network outputs are proportional (K_P), integral (K_I), derivative (K_D) coefficients.

In discrete time PID-regulator can be described using the following formula:

$$u_n = u_{n-1} + K_P(e_n - e_{n-1}) + K_I e_n + K_D(e_n - 2e_{n-2} + e_{n-2}),$$

where K_P , K_I and K_D — proportional, integral and derivative coefficients respectively, u_n defines the output of the control object at moment $t = nT_0$ and e_n — error between desired output value r_n and the real one, meaning

$$e_n = r_n - y_n,$$

where T_0 defines a singular time interval.

To adjust K_P , K_I and K_D during the operation a 3-layer perceptron will be used. Each layer has N_1 , N_2 and N_3 neurons respectively. The number of neurons is selected based on the researcher's expertise and complexity of the control object, N_3 equals 3 - number of coefficients in PID. In order to use the backpropagation

algorithm a function E must be chosen for minimization. Control error function e_n is used at moment nT_0 .

$$E_n = \frac{e_n^2}{2}$$

For error accumulation the previous data is saved — $E_{n-p}, \dots, E_{n-2}, E_{n-1}, E_n$, where p defines the number of previously saved data points for neural network training.

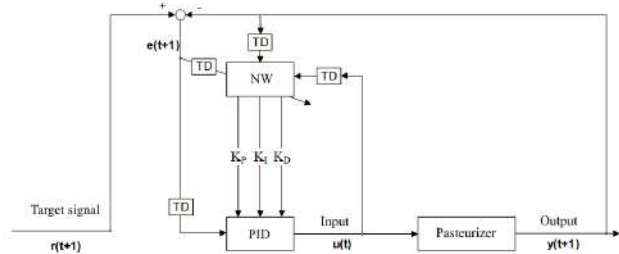


Figure 4. Neuro-PID regulator, TD means time-delay elements

Therefore approaches based on the modern machine learning algorithms such as neural network allow solving complex control adaptation tasks for the technological processes and can be used as a base for constructing problem solvers for the appropriate ostis-systems.

VII. CONCLUSION

Lately algorithms and methods are being actively developed and improved for the class of applied problems related to control of robotic production when external random influences on the control object are present.

In this paper a procedure is being proposed for synthesizing feedback control connections for technological production cycle and adaptation methods are proposed based on formalization within the framework of OSTIS Ecosystem and building artificial neural network models for solving applied tasks of control optimization for complex technological systems.

The obtained results make it possible to develop new hybrid intellectual computer systems with the purpose of solving a wide class of applied problems of control adaptation in real-time for technological objects using ostis-systems and technical means of hardware-software coupling of the control object.

REFERENCES

- [1] V. Golenkov, N. Gulyakina, N. Grakova, I. Davydenko, V. Nikulnka, A. Ereemeev, V. Tarasov. *From Training Intelligent Systems to Training Their Development Tools. Open Semantic Technologies for Intelligent Systems (OSTIS)*, Minsk, Belarussian State University of Informatics and Radioelectronics Publ., 2018, iss. 2, pp. 81–98.
- [2] V. S. Smorodin, I. V. Maximey *Methods and means of simulation modeling of technological production processes*, Gomel, F. Skorina State University, 2007.
- [3] M. T. Hagan, H. B. Demuth *Neural networks for control. Proceedings of the American Control Conference*. San Diego, USA, 1999, Vol. 3, pp. 1642–1656.

- [4] D. A. White, D. A. Sofge. Handbook of Intelligent Control. Neural, Fuzzy, and Adaptive Approaches. Van Nostrand Reinhold, 1992.
- [5] O. Omidvar, D. L. Elliott (eds.) Neural Systems for Control. Academic Press, New York, 1997, P. 358
- [6] D. S. Ivanyuk POU neyro-PID-regulyator. *Vestnik Brestskogo gosudarstvennogo tehničeskogo universiteta. Series: Physics, Mathematics, Computer science*, 2014, No. 5, pp. 35–40.
- [7] D. Ivaniuk Neuro-PID Controller for a Pasteurizer. *XV International PhD Workshop OWD*, 2013.
- [8] V. Taberko, D. Ivaniuk Ontological approach to batch enterprise within Industry 4.0. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2022, pp. 395–400.
- [9] V. Smorodin, V. Prokhorenko Control Of A Technological Cycle Of Production Process Based On A Neuro-Controller Model, *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2019, pp. 251–256.
- [10] V. Smorodin, V. Prokhorenko Adaptive Control Of Robotic Production Systems. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2020, pp. 161–166.
- [11] V. Smorodin, V. Prokhorenko Software-Technological Complex For Adaptive Control Of A Production Cycle Of Robotic Manufacturing. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2022, pp. 401–404.
- [12] V. Smorodin, V. Prokhorenko (2019) Application of Neuro-Controller Models for Adaptive Control. In: Chertov O., Mylovanov T., Kondratenko Y., Kacprzyk J., Kreinovich V., Stefanuk V. (eds) Recent Developments in Data Science and Intelligent Analysis of Information. ICDSIAI 2018. Advances in Intelligent Systems and Computing, Springer, Cham, vol. 836, pp. 30–38.
- [13] V. Taberko, D. Ivaniuk, D. Shunkevich, O. Pupena Principles for enhancing the development and use of standards within Industry 4.0. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2020, pp. 167–174.
- [14] Stanley, Kenneth O.; Miikkulainen, Risto (2002-06-01). Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation*. 10 (2): 99–127. CiteSeerX 10.1.1.638.3910. doi: 10.1162/106365602320169811. ISSN 1063-6560. PMID 12180173. S2CID 498161.
- [15] N. Lutska, O. Pupena, A. Shyshak, V. Taberko, D. Ivaniuk, M. O. Nikita Zotov, and L. Vlasenko Ontological model of digital twin in manufacturing. *Open semantic technologies for intelligent systems*. BSUIR, Minsk, 2022, pp. 310–335.
- [16] ISA-88 standard. Available at: <https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa88/> (accessed 2023, Jan).
- [17] ISA-95 standard. Available at: <https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa95/> (accessed 2023, Jan).
- [18] ISA5.1 Standard. <https://www.isa.org/standards-and-publication/s/isa-standards/isa-standards-committees/isa5-1/> (accessed 2023, Jan)
- [19] EasyEPLANner project on GitHub. Available at: <https://github.com/savushkin-r-d/EasyEPLANner/> (accessed 2023, Jan).
- [20] EasyServer-4.0 project on GitHub. Available at: <https://github.com/savushkin-r-d/EasyServer-4.0> (accessed 2023, Jan).
- [21] s88-ostis project on GitHub. Available at: <https://github.com/savushkin-r-d/s88-ostis> (accessed 2023, Jan).
- [22] isa-5.1-ostis project on GitHub. Available at: <https://github.com/savushkin-r-d/isa-5.1-ostis> (accessed 2023, Jan)
- [23] L.V. Massel, A.G. Massel Semanticheskoe modelirovanie pri postroenii tsifrovyykh dvoynikov energeticheskikh ob"ektov i sistem [Semantic modeling in the construction of digital twins of energy objects and systems] , *Ontologiya proektirovaniya [Design ontology]*, Vol. 13, No 1 (47), 2023, pp. 45–54.

Система адаптивного управления технологическим циклом производства в рамках Экосистемы OSTIS

Таберко В. В., Иванюк Д. С.,
Сморodin В. С., Прохоренко В. А.

В настоящей работе предлагается подход к построению гибридной интеллектуальной компьютерной системы адаптивного управления технологическим циклом производства в виде решателя соответствующей ostis-системы на основе онтологии предметной области «технологические процессы производства с вероятностными характеристиками». Основой для создания решателя системы служит идея разработки и использования математических моделей нейросетевых регуляторов для решения задач оптимизации управления. Такая реализация позволяет обеспечить возможность интеграции предлагаемого решения с другими разработками, программными средствами предприятия для обеспечения построения интеллектуальных систем автоматизированного управления, рекомендательных систем и систем поддержки принятия решений, систем информационного обеспечения персонала предприятия.

Received 25.03.2023

Support of The Life Cycle of Intelligent Geoinformation Systems for Various Purposes

Sergei Samodumkin

Belarusian State University of Informatics and Radioelectronics

Minsk, Belarus

Email: Samodumkin@bsuir.by

Abstract—The article is dedicated to the particular technology for intelligent geoinformation systems design built on the principles of ostis-systems. The structural and semantic interoperability of geoinformation systems built using the proposed technology is ensured by the transition from the map to the semantic description of map elements.

Keywords—OSTIS, intelligent geoinformation system, intellectualization, map, interoperability

I. INTRODUCTION

In geoinformatics, fundamental knowledge about space, time, and the Earth is systematically organized on the basis of information encoding.

Unlike other classes of *information systems*, in *geoinformation systems*, the main object of research is knowledge and data about *terrain objects*, which are not only considered as spatial data and knowledge, but also are the integration basis for various *subject domains*. At the same time, the formalization of such *knowledge* and their representation in *knowledge bases of intelligent systems* requires the establishment of relations to describe the properties and patterns inherent in the *subject domain* under consideration and using *terrain objects*, the establishment of geometric characteristics capable of binding *terrain objects*, and also taking into account the temporal nature of the existence of *terrain objects*, which allows for a retrospective analysis. Taking into account the fact that *intelligent systems* are designed to meet *information needs of users*, this fact contributes to the expansion of *subject domains* and the addition of new functionality within the proposed particular *Technology for intelligent geoinformation systems design*.

According to the generally accepted definition, a *geoinformation system* is a *computer software system*, which provides input, manipulation, analysis, and output of spatially correlated data (geodata) about the territory, social and natural phenomena in solving problems related to inventory, analysis, modeling, forecasting, and environmental management, as well as territorial organization of the society [1].

Consequently, the very definition of the *geoinformation system* implies the need to implement intelligent problems:

- analysis problem

- modeling problem
- forecasting and environmental management problem

All these problems are intelligent and require decision-making support when they are implemented.

Within this article, fragments of structured texts in the SCn-code will often be used, which are simultaneously fragments of the source texts of the knowledge base, understandable to both human and machine. This allows making the text more structured and formalized, while maintaining its readability. The symbol “:=” in such texts indicates alternative (synonymous) names of the described entity, revealing in more detail certain of its features.

II. REQUIREMENTS FOR INTELLIGENT GEOINFORMATION SYSTEMS OF A NEW GENERATION

intelligent geoinformation system

:= [information system, the main object of research of which is knowledge and data about terrain objects, acting as an integration basis for solving applied problems in various subject domains]

⊃ *intelligent geoinformation ostis-system*

:= [intelligent geoinformation system developed according to the principles of the OSTIS Technology]

An important point that reduces the development time of *intelligent systems* on the one hand, and on the other hand increases the functionality of *intelligent systems* that use knowledge about *terrain objects* as integration is the availability of design technology and tools. At the same time, the *Technology for intelligent geoinformation systems design* should ensure the reuse of information and functional components of the system in order to reduce the design and development time for applied systems. Thus, it is about creating a particular *Technology for intelligent geoinformation systems design*. In this connection, the relevant problems are:

- design of spatial ontologies and the solution on its basis of the problem with *semantic compatibility* of *knowledge of subject domains*;
- solving the problem of metadata management and improving search, access, and exchange in the context of growing volumes of spatial information

and services provided by numerous sources of *geoinformation*;

- implementation of knowledge output using spatial and thematic information as components of *knowledge* about *terrain objects* using the *Question Language*;
- implementation of a *cartographic interface* in *intelligent ostis-systems* as a natural way for a human to represent information about *terrain objects*.

The constant evolution of models and means of ontological description of subject domains, using spatial and temporal components, their heterogeneity, and ambiguity, poses new challenges in terms of interaction, integration, and compatibility of various applied systems due to:

- the integration of *subject domains* and their corresponding ontologies (vertical level);
- expanding the functionality of the systems using reusable components of these systems (horizontal level), in particular, designing components for new territories and/or in a new time interval.

In order to implement the requirements represented, it is proposed to consider the map as an *information construction*, the elements of which are *terrain objects*, and to offer:

- the *Subject domain and ontology of terrain objects*;
- *Map Language Syntax*;
- *Denotational semantics of the Map Language*.

The transition from maps to their *meaning* is based on:

- the formal description of the *Map Language Syntax*;
- the formal description of the *Denotational semantics of the Map Language*.

At the same time, *semantic compatibility of geoinformation systems* and their components are provided due to the common ontology of *terrain objects*, which is necessary for the interoperability of *geoinformation systems* for various purposes and their components.

Thus, structural and semantic interoperability of geoinformation systems is ensured due to the transition from the map to the semantic description of map elements, that is, terrain objects and connections (spatial relations) between them.

The presence of these circumstances determines the existence of a scientific and technical problem of *intellectualization of geoinformation systems* and the creation of the *Technology for intelligent geoinformation systems design*, which are based on the principles of designing *ostis-systems*.

III. SYSTEMATIZATION OF PROBLEMS SOLVED BY INTELLIGENT GEOINFORMATION SYSTEMS

One of the ways to increase the efficiency of using information and computing tools is *intellectualization of geoinformation systems*.

Intellectualization of geoinformation systems implies:

- the possibility of end-user communication with the system on the *Question Language*;
- the use of various interoperable problem solvers with the possibility of explaining the solutions obtained;
- the use of *cartographic interface* to visualize the source data and results.

The implementation of the capabilities of *intelligent geoinformation systems* can be carried out using:

- *knowledge base management systems*;
- *multimedia knowledge and databases* by application areas;
- *interoperable problem solvers*;
- an *intelligent cartographic interface*;
- *expert systems* in various fields of human activities;
- *decision support systems*;
- *intelligent assistance systems*.

Intellectualization of geoinformation systems involves solving the following problems:

- the use of digital cartographic material and data from *remote sensing of the Earth* in problem-oriented areas [2];
- planning actions in a dynamically changing situation in conditions of incomplete or fuzzy data using expert knowledge [3];
- analysis of emergency situations and preparation of materials for decision-making on prevention or elimination of their consequences;
- creation of decision support systems for applied *geoinformation systems* of territorial planning and management [4];
- development of diagnostic expert systems for geological exploration activities with remote access to them;
- logistics planning, creation of expert systems and enterprise management software;
- creation of control and navigation systems;
- creation of *expert systems* for forecasting the occurrence and development of technogenic and natural situations: floods, earthquakes, extreme weather conditions (precipitation, temperature), epidemics, spread of radionuclides, chemical emissions, meteorological forecast, etc.;
- creation of *expert systems* for the selection of terrain compartments for the construction of various objects;
- creation of *expert systems* for planning the efficient use of agricultural land;
- creation of *expert systems* and software tools for geodata analysis;
- creation of image and picture recognition systems based on data from *remote sensing of the Earth*;
- creation of banks of digital cartographic information with means of remote access to them;
- image processing;
- retrospective analysis of events (see [5], [6];

- creation of *information search systems* for Earth sciences and geoinformatics;
- development of educational systems for training specialists and experts with means of remote access to them.

The complete solution of the above problems requires the use of open system standards and the use of ontologies of *terrain objects* as integrating elements of various *subject domains*.

IV. MAIN COMPONENTS OF FORMAL ONTOLOGIES USED IN GEOINFORMATION SYSTEMS

The main approach to ensuring semantic interoperability is the development of ontologies. The most frequently used ontologies in geoinformatics are usually considered as domain ontologies, which are commonly called geographical ontologies, or geontologies [7], [8]. One of the problems in the development of ontologies is a clear and unambiguous definition of the semantics of primitive terms (atomic elements that cannot be further separated). To solve this problem, the researchers proposed to justify the primitive terms of geontologies on the basis of geographical phenomena [9], [10].

In relation to subject domains, an ontology is the formalization of a certain area of knowledge based on a conceptual scheme with a structure containing classes of objects, their relations, and rules that allows for computer analysis. Accordingly, the ontology of the subject domain includes instances, concepts, attributes, and relations.

The subject domains for which the development of geoinformation systems is appropriate involve the construction of an ontology, which we will call geontology.

geontology

$:=$ [ontology of subject domains, the object instances of which includes geosemantic elements]

$:=$ [ontology of subject domains, object instances of which use spatially correlated data about the territory, social and natural phenomena]

\subset *ontology*

\ni *terrain object class*[^]

\ni *spatial relation*

terrain object class[^]

$:=$ [a class of geospatial concepts of natural or artificial origin, natural phenomena having common features (semantic attributes) that are characteristic of a certain terrain object class and describe the internal characteristics of the concept]

terrain object

$:=$ [a certain element of the Earth surface of natural or artificial origin, a natural phenomenon that actually exists at the time under consideration within the localization area, for which the location is known or can be established, including

the size and position of the boundaries, and signs are set, reflecting the semantic attributes of such an element, characteristic of a certain *terrain object class*, with set *spatial relations* reflecting connections with other *terrain objects*]

A feature of the *geontology* is the use of special elements for the formalization of subject domains that clarify the spatial characteristics of terrain objects, which we will call **geosemantic elements**.

geosemantic element

\Rightarrow *subdividing*^{*}:

- *coordinate location of the terrain object*[^]
- *spatial relation*
- *spatial relation of the main directions*
- *dynamics of the state of the terrain object*[^]

geocoding

$:=$ [establishing a connection between a *terrain object* and its *location*]

\subset *action*

spatial relation

$:=$ [class of relations that define the semantic properties of a terrain object in relation to other terrain objects]

\Rightarrow *subdividing*^{*}:

- *topological spatial relation*
- *spatial ordering relation*
- *metric spatial relation*

spatial ordering relation

\Rightarrow *subdividing*^{*}:

- *relation of location of terrain objects*
- *relation of the main directions of terrain objects*

relation of location of terrain objects

\subset *oriented relation*

$:=$ [allows determining what position one *terrain object* occupies in relation to another *terrain object*]

\supset *terrain object is located in front of another terrain object*^{*}

\supset *terrain object is located behind another terrain object*^{*}

\supset *terrain object is located to the left of another terrain object*^{*}

\supset *terrain object is located to the right of another terrain object*^{*}

\supset *terrain object is located above another terrain*

- object*
- ⊃ terrain object is located under another terrain object*
- ⊃ terrain object is located closer than another terrain object*
- ⊃ terrain object is located further than another terrain object*

relation of the main directions of terrain objects

- ⊂ oriented relation
- := [allows determining which main direction one terrain object occupies in relation to another terrain object]
- ⊃ terrain object in relation to another terrain object occupies the main north direction*
- ⊃ terrain object in relation to another terrain object occupies the main north-east direction*
- ⊃ terrain object in relation to another terrain object occupies the main east direction*
- ⊃ terrain object in relation to another terrain object occupies the main south-east direction*
- ⊃ terrain object in relation to another terrain object occupies the main south direction*
- ⊃ terrain object in relation to another terrain object occupies the main south-west direction*
- ⊃ terrain object in relation to another terrain object occupies the main west direction*
- ⊃ terrain object in relation to another terrain object occupies the main north-west direction*

metric spatial relation

- := [characterizes information about the distance between terrain objects]
- ⇒ measurement*:
kilometer
- ⇒ measurement*:
meter
- ⊃ scale metric spatial relation

geodetic coordinate system

- := [coordinate system used to determine the location of objects on the Earth]
- ⊃ example':
WGS84
- := [The world system of geodetic parameters of the Earth, 1984, which includes a system of geocentric coordinates, and unlike local systems, it is a single system for the entire planet]
- ⊃ example':
CK-95

V. FORMALIZATION OF TOPOLOGICAL SPATIAL SEMANTIC RELATIONS IN GEOINFORMATION SYSTEMS

Between instances of *terrain objects*, it is possible to establish *topological spatial relations*:

topological spatial relation

- := [spatial relation class, defined over terrain objects that are in relation of connectivity and adjacency between terrain objects]
- ⊃ inclusion*
 - ⊃ inclusion of a point terrain object in an area terrain object*
 - ⊃ inclusion of a linear (multilinear) terrain object in an area terrain object*
 - ⊃ inclusion of an area terrain object in an area terrain object*
- ⊃ border*
- ⊃ intersection*
 - ⊃ intersection of two linear (multilinear) terrain objects*
 - ⊃ intersection of linear (multilinear) and area terrain objects*
- ⊃ adjacency*

The “inclusion*” relation will be set between *area* and *linear*, *area* and *point*, *area terrain objects*. The “intersection*” relation will be set between *linear* and *area* and *linear terrain objects*. The “border*” relation will be established between *area terrain objects*. The “adjacency*” relation is established between *linear terrain objects*. For all *cartographic relations*, there are structures for storing them.

VI. SUBJECT DOMAIN AND ONTOLOGY OF TERRAIN OBJECTS

For the purpose of *integration* of *subject domains* with spatial components of *geoinformation systems*, respectively increasing *interoperability* of these systems, a *hybrid knowledge model* is proposed. By this model we will understand a *stratified model of the information space of terrain objects* described in the work [11].

terrain object

- ⇒ subdividing*:
- Typology of terrain objects by topic[^]**
- = {
 - water terrain object (facility)
 - populated terrain object
 - industrial (agricultural or socio-cultural) terrain object
 - road network (facility)
 - vegetation cover (soil)
- }

The basis for building the ontological model of *terrain objects* is grounded on the classifier of topographic information displayed on topographic maps and city plans developed and currently functioning in the *Republic of Belarus* [12]. In accordance with this circumstance, the objects of classification are the terrain objects to which the map objects correspond, as well as the signs (characteristics) of these objects. For this purpose, in the ontological model, *terrain objects* are divided by

localization type into: *area objects*[^], *linear (multilinear) objects*[^], and *point objects*[^].

At the next stage of developing the ontology of *terrain objects*, we will set the subdivision of *terrain objects* on orthogonal bases, which corresponds to the placement of objects in accordance with thematic layers in *geoinformation systems*.

For each *terrain object*, the main semantic characteristics inherent only to it are highlighted. It should be particularly noted that metric characteristics do not have such a property. According to this classifier, each class of *terrain objects* has a unique unambiguous designation. The classifier hierarchy has eight classification stages and consists of the *class code*, *subclass code*, *group code*, *subgroup code*, *order code*, *suborder code*, *species code*, *subspecies code*. Thus, thanks to the coding method, generic relations have already been defined, reflecting the correlation of various *terrain object* classes, and the characteristics of a specific *terrain object* class have also been established. Due to the fact that the basic properties and relations are set not of specific *physical objects* but of their classes, then such information is meta-information in relation to specific *terrain objects*, and the totality of this meta-information is an ontology of *terrain objects*, which in turn is part of the *knowledge base* of the *intelligent geoinformation system*.

terrain object

⇒ *subdividing*^{*}:

Typology of terrain objects by localization[^]

- = {
 - *point terrain object*
 - ⇒ *inclusion*^{*}:
 - *well*
 - *light post*
 - *linear terrain object*
 - ⇒ *inclusion*^{*}:
 - *bridge*
 - *multilinear terrain object*
 - ⇒ *inclusion*^{*}:
 - *river*
 - *road*
 - *area terrain object*
 - ⇒ *inclusion*^{*}:
 - *lake*
 - *administrative area*

VII. SPECIFICATION OF THE MAP LANGUAGE

The *Map Language* belongs to the family of semantic compatible languages – *sc-languages* – and is intended for the formal description of *terrain objects* and the relations between them in *geoinformation systems*. Therefore, the *Map Language Syntax*, like *syntax* of any other *sc-language*, is the *Syntax of the SC-code*. This approach allows:

- using a minimum of means to interpret the specified *terrain objects* on the map;
- using the *Question Language for ostis-systems*;
- reducing the search to most of the given *questions* to searching for information in the current state of the *ostis-system knowledge base*.

Denotational semantics of the Map Language includes the *Subject domain and the ontology of terrain objects* and their *geosemantic elements*.

VIII. AUTOMATION TOOLS FOR THE INTELLIGENT GEOINFORMATION SYSTEMS DESIGN

The design of intelligent geoinformation systems is carried out in stages. At the first stage, the knowledge base of the subject domain is formed and for this purpose an electronic map (voluntary cartographic information) is analyzed and translated into the knowledge base of terrain objects with the establishment of geosemantic elements for the corresponding territory. At this stage, it is determined, firstly, to which class the terrain object under study belongs and, further, depending on the type of object, the concept of a knowledge base corresponding to a specific physical terrain object is formed. Thus, many concepts are created that describe specific terrain objects for each class of terrain objects. It should be noted that it is at this stage of the formation of the knowledge base that semantic elements are established. At the second stage of designing an intelligent geoinformation system, the knowledge base obtained at the first stage is integrated with external knowledge bases. At this stage, in addition to geographical knowledge, knowledge of related subject domains is added, thereby it becomes possible to establish interdisciplinary connections. An illustrative example is integration with biological classifiers, which in implementation represent an ontology of flora and fauna objects. Such integration expands the functional and intelligent capabilities of the applied intelligent geoinformation system. Note that at this stage, homonymy is removed in the names of geographical objects belonging to the classes of settlements. For settlements of the Republic of Belarus, this is achieved by using the *system of designations of administrative-territorial division objects and settlements* and semantic comparison of geographical terrain objects is carried out according to the following principle:

- the terrain object class is determined;
- the terrain object subclass, species, subspecies, etc. is determined in accordance with the classifier of terrain objects, i.e. types of terrain objects in the ontology;
- the attributes and characteristics that are inherent in this terrain object class are determined;
- the values of the characteristics for this object class are determined;
- the homonymy of identification is eliminated;

- appropriate connections are established between the map object, the concept in the knowledge base with the established geosemantic elements;
- spatial relations are established between terrain objects assigned to certain classes.

CONCLUSION

Let us list the main provisions of this article:

- the development of geoinformation systems consists in their intellectualization, thereby expanding the range of applied problems using knowledge about terrain objects;
- it is proposed to consider the map as an *information construction*, the elements of which are *terrain objects*, thereby ensuring the structural and semantic interoperability of geoinformation systems due to the transition from the map to the semantic description of map elements, that is, terrain objects and connections (spatial relations) between them;
- ensuring semantic interoperability is achieved through the development of ontologies of subject domains, and the establishment of *geosemantic elements* allows setting spatial characteristics of terrain objects;
- availability of a particular *Technology for intelligent geoinformation systems design* provides the process of designing intelligent geoinformation systems built on the principles of ostis-systems.

ACKNOWLEDGMENT

The author would like to thank the research group of the Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics for its help in the work and valuable comments.

REFERENCES

- [1] A. Kryuchkov, S. Samodumkin, M. Stepanova, and N. Gulyakina, *Intellektual'nye tekhnologii v geoinformatsionnykh sistemakh [Intelligent technologies in geoinformation systems]*. BSUIR, 2006, p. 202 (In Russ.).
- [2] S. Ablameyko, G. Aparin, and A. Kryuchkov, *Geograficheskie informatsionnye sistemy. Sozdanie cifrovyykh kart [Geographical information systems. Creating digital maps]*. Institute of Technical Cybernetics of the National Academy of Sciences of Belarus, 2000, p. 464 (In Russ.).
- [3] Ya. Ivakin, "Metody intellektualizatsii promyshlennykh geoinformatsionnykh sistem na osnove ontologii [Methods of intellectualization of industrial geoinformation systems based on ontologies]," Doct. diss.: 05.13.06, Saint-Petersburg, 2009, (In Russ.).
- [4] M. Belyakova, *Intellektual'nye geoinformatsionnye sistemy dlya upravleniya infrastrukturoy transportnykh kompleksov [Intelligent geoinformation systems for infrastructure management of transport complexes]*. Taganrog : Southern Federal University Press, 2016, p. 190 (In Russ.).
- [5] A. Gubarevich, O. Morosin, and D. Lande, "Ontologicheskoe proektirovaniye intellektual'nykh sistem v oblasti istorii [Ontological design of intelligent systems in the field of history]," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for designing intelligent systems]*, pp. 245–250, 2017, (In Russ.).
- [6] A. Gubarevich, S. Vityaz, and R. Grigyanets, "Struktura baz znaniy v intellektual'nykh sistemakh po istorii [Structure of knowledge bases in intelligent systems on history]," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for designing intelligent systems]*, pp. 347–350, 2018, (In Russ.).
- [7] A. Bliskavitsky, *Konceptual'noye proektirovaniye GIS i upravleniye geoinformatsiej. Tekhnologii integratsii, kartograficheskogo predstavleniya, veb-poiska i rasprostraneniya geoinformatsii [Conceptual GIS design and geoinformation management. Technologies of integration, cartographic representation, web search, and distribution of geoinformation]*. LAP LAMBERT Academic Publishing, 2012, p. 484 (In Russ.).
- [8] —, "Semantika geoprostranstvennykh ob'ektov, funktsional'naya grammatika i intellektual'nye GIS [Semantics of geospatial objects, functional grammar, and intelligent GIS]," in *Izvestiya vysshikh uchebnykh zavedeniy. Geologiya i razvedka [News of higher educational institutions. Geology and exploration]*, no. 2, 2014, pp. 62–69, (In Russ.).
- [9] Y. Hu, "Geospatial semantics," *Comprehensive Geographic Information Systems*, pp. 80–94, 2018.
- [10] K. Janowicz, S. Simon, T. Pehle, and G. Hart, "Geospatial semantics and linked spatiotemporal data — past, present, and future," *Semantic Web*, vol. 3, pp. 321–332, 10 2012.
- [11] S. Samodumkin, "Next-generation intelligent geoinformation systems," *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, 2022.
- [12] *Cifrovyye karty mestnosti informatsiya, otobrazhaemaya na topograficheskikh kartakh i planakh naseleennykh punktov : OKRB 012-2007 [Digital maps of the area information displayed on topographic maps and plans of settlements : NKRБ 012-2007]*, Minsk, 2007, (In Russ.).

Поддержка жизненного цикла интеллектуальных геоинформационных систем различного назначения

Самодумкин С.А.

Работа посвящена частной технологии проектирования интеллектуальных геоинформационных систем, построенных по принципам ostis-систем. Структурная и семантическая интероперабельность геоинформационных систем, построенных по предлагаемой технологии, обеспечивается за счет перехода от карты к семантическому описанию элементов карты.

Received 01.03.2023

Ensuring Information Security of the OSTIS Ecosystem

Valery Chertkov
*Euphrosyne Polotskaya State
University of Polotsk*
Polotsk, Belarus
v.chertkov@psu.by

Vladimir Zakharau
*Belarusian State University of
Informatics and Radioelectronics*
Minsk, Belarus
zakharau@bsuir.by

Abstract—The development of artificial intelligence systems, associated with the transition to working with knowledge bases instead of data, requires the formation of new approaches to ensuring information security systems. The article is devoted to the review of approaches and principles of ensuring security in intelligent systems of the new generation. The current state of methods and means of ensuring information security in intelligent systems is considered and the main goals and directions for the development of information security ostis-systems are formed. The information security methods presented in the article are extremely important when designing the ostis-systems security system and analyzing their security level.

Keywords—information security, new generation intelligent system, Information security threats

I. INTRODUCTION

A wide variety of information security models, the growing amount of data that needs to be analyzed to detect attacks on information systems, the variability of attack methods and the dynamic change in protected information systems, the need for a rapid response to attacks, the fuzziness of the criteria for detecting attacks and the choice of methods and means of responding to them, the lack of highly qualified security specialists entails the need to use artificial intelligence methods to solve security problems.

II. THE SPECIFICS OF ENSURING INFORMATION SECURITY OF INTELLIGENT SYSTEMS OF A NEW GENERATION

Information security of intelligent systems should be considered from two points of view:

- application of artificial intelligence in information security;
- organization of information security in intelligent systems.

The use of artificial intelligence in information security

Artificial intelligence is actively used to monitor and analyze security vulnerabilities in information transmission networks [1]. The artificial intelligence system allows machines to perform tasks more efficiently, such as:

- visual perception, speech recognition, decision making and translation from one language to another;
- invasion detection - artificial intelligence can detect network attacks, malware infections and other cyber threats;
- cyber analytics - artificial intelligence is also used to analyze big data in order to identify patterns and anomalies in the organization's cyber security system in order to detect not only known, but also unknown threats;
- secure software development - artificial intelligence can help create more secure software by providing real-time feedback to developers.

Artificial intelligence is used not only for protection, but also for attack, for example, to emulate acoustic, video and other images in order to deceive authentication mechanisms and further impersonation, deceive checking a person or robot captcha, etc.

Currently, it is possible to define the following classes of systems in which artificial intelligence is used [2]:

- UEBA (User and Entity Behavior Analytics) — a system for analyzing the behavior of subjects (users, programs, agents, etc.) in order to detect non-standard behavior and use them to detect potential threats using threat templates (patterns);
- IP (Threat Intelligence Platform) — platforms for early detection of threats based on the collection and analysis of information from indicators of compromise and response to them. The use of machine learning methods increases the efficiency of detecting unknown threats at an early stage;
- EDR (Endpoint Detection and Response) — attack detection systems for rapid response at the end points of a computer network. Can detect malware, automatically classify threats and respond to them independently;
- SIEM (Security Information and EventManagement) — systems for collecting and analyzing information about security events from network devices and applications in real time and alerts;
- NDR (Network Detection and Response) — sys-

tems for detecting attacks at the network level and promptly responding to them. AI uses the accumulated statistics and knowledge base about threats;

- SOAR (Security Orchestration and Automated Response) — systems that allow you to identify information security threats and automate incident response. In solutions of this type, unlike SIEM systems, AI helps not only to analyze, but also automatically respond appropriately to identified threats;
- Application Security — systems that allow you to identify threats to the security of application applications, manage the process of monitoring and eliminating such threats;
- Antifraud — platforms detect threats in business processes and fraudulent transactions in real time. AI is used to identify deviations from identified business processes in order to detect intrusions or process vulnerabilities and increase adaptability to changing business process logic and metrics.

The paper [3] proposes a method for constructing a neuroimmune system for analyzing information security incidents that combines data collection and storage (compression) modules, an information security event analysis and correlation module, and a network attack detection subsystem based on convolutional neural networks. The use of machine learning technologies in information security creates bottlenecks and system vulnerabilities that can be exploited and has the following disadvantages [4]:

- data sets that must be formed from a significant number of input samples, which requires a lot of time and resources;
- requires a huge amount of resources, including memory, data and computing power;
- frequent false positives that disrupt the operation and generally reduce the effectiveness of such systems;
- organized attacks based on artificial intelligence (semantic viruses).

Organization of information security in intelligent systems of a new generation

Let's define the goals of ensuring the information security of new generation systems.

From the monograph [5] the objectives of ensuring the information security of traditional intelligent systems are:

- ensuring the confidentiality of information in accordance with the classification;
- ensuring the integrity of information at all stages of related processes (creation, processing, storage, transfer and destruction) in the provision of public services;
- ensuring timely availability of information in the provision of public services;

- ensuring observability aimed at capturing any activity of users and processes;
- ensuring the authenticity and impossibility of refusal of transactions and actions performed by participants in the provision of public services;
- accounting for all processes and events related to the input, processing, storage, provision and destruction of data.

Since intelligent systems of the new generation will interact with similar systems while understanding what the request is about, the goals of the provision will look different. The goals of ensuring the information security of new generation intelligent systems are:

- ensuring the safety of the semantic compatibility of information;
- protection of reliability and integrity of information;
- ensuring the availability of information at different levels of the intellectual system;
- minimization of damage from events that pose a threat to information security.

Currently, classical approaches and principles have been developed to ensure the security of knowledge bases (data), communication interfaces (information exchange) between the components of intelligent systems, such as encryption of transmitted data, filtering of unnecessary (redundant) content, and data access control policy.

The information security system should be created on the following principles:

- the principle of equal strength - means ensuring the protection of equipment, software and control systems from all types of threats;
- the principle of continuity - provides for continuous security of information resources of the system for the continuous provision of public services;
- the principle of reasonable sufficiency - means the application of such measures and means of protection that are reasonable, rational and the costs of which do not exceed the cost of the consequences of information security violations;
- the principle of complexity - to ensure security in all the variety of structural elements, threats and channels of unauthorized access, all types and forms of protection must be applied in full;
- the principle of comprehensive verification - is to conduct special studies and inspections, special engineering analysis of equipment, verification studies of software. Emergency messages and error parameters should be continuously monitored, hardware and software equipment should be constantly tested, as well as software integrity control, both during software loading and during operation;
- the principle of reliability - methods, means and forms of protection must reliably block all penetration routes and possible channels of information

leakage; for this, duplication of means and security measures is allowed;

- the principle of universality - security measures should block the path of threats, regardless of the place of their possible impact;
- the principle of planning – planning should be carried out by developing detailed action plans to ensure the information security of all components of the system for the provision of public services;
- the principle of centralized management – within a certain structure, the organized and functional independence of the process of ensuring security in the provision of public services should be ensured;
- the principle of purposefulness – it is necessary to protect what must be protected in the interests of a specific goal;
- the principle of activity - protective measures to ensure the safety of the service delivery process must be implemented with a sufficient degree of persistence;
- the principle of service personnel qualification – maintenance of equipment should be carried out by employees who are trained not only in the operation of equipment, but also in technical issues of information security;
- the principle of responsibility - the responsibility for ensuring information security must be clearly established, transferred to the appropriate personnel and approved by all participants as part of the information security process.

III. THE PRINCIPLES UNDERLYING THE INFORMATION SECURITY OF OSTIS SYSTEMS

The OSTIS ecosystem is a collective of interacting:

- ostis-systems;
- users of ostis systems (end users and developers);
- other computer systems that are not ostis-systems, but are additional information resources or services for them.

The core of OSTIS technology includes the following components:

- semantic knowledge base OSTIS, which can describe any kind of knowledge, while it can be easily supplemented with new types of knowledge;
- OSTIS problem solver based on multi-agent approach. This approach makes it easy to integrate and combine any problem solving models;
- ostis-system interface, which is a subsystem with its own knowledge base and problem solver.

The presented architecture of the OSTIS Ecosystem implements:

- all knowledge bases are united into the Global Knowledge Base, the quality of which (logicality, correctness, integrity) is constantly checked by many agents. All problems are described in a single

knowledge base, and specialists are involved to eliminate them, if necessary;

- each application associated with the OSTIS Ecosystem has access to the latest version of all major OSTIS components, components are updated automatically;
- each owner of the OSTIS Ecosystem application can share a part of their knowledge for a fee or for free.

It is important to note that information security is closely related to the architecture of the built system: a well-designed and well-managed system is more difficult to hack. Therefore, it is very important to develop an information security system at the stage of designing the architecture and structure of a future next-generation intelligent system.

The OSTIS Ecosystem is a community where ostis systems and users interact, where rules must be established and controlled. Illegal and destabilizing actions by all members of the community should not be allowed. The user cannot directly interact with other ostis systems, but only through a personal agent. This agent stores all personal data of the user and access to them should be limited.

In the OSTIS Ecosystem, all agents must be identified. It should be noted that the personal user agent in the Ecosystem solves the problem of identifying the user himself.

In the considered OSTIS Ecosystem, it is required to organize information security at each of the levels of interaction: data exchange, data access rights, authentication of Ecosystem clients, data encryption, obtaining data from open sources, ensuring the reliability and integrity of stored and transmitted data, monitoring the violation of communications in knowledge base, tracking vulnerabilities in the system.

threat in ostis-system

- ⊃ *threat. breach of confidentiality of information*
⇒ *explanation**:
[unauthorized access to read information]
- ⊃ *threat. violation of the integrity of information*
⇒ *explanation**:
[unauthorized or erroneous change, distortion or destruction of information, as well as unauthorized impact on technical and software information processing tools]
- ⊃ *threat. accessibility violation*
⇒ *explanation**:
[blocking access to the system, its individual components, functions or information, as well as the impossibility of obtaining information in a timely manner (unacceptable delays in obtaining information)]
- ⊃ *threat. violation of semantic compatibility*
⇒ *explanation**:

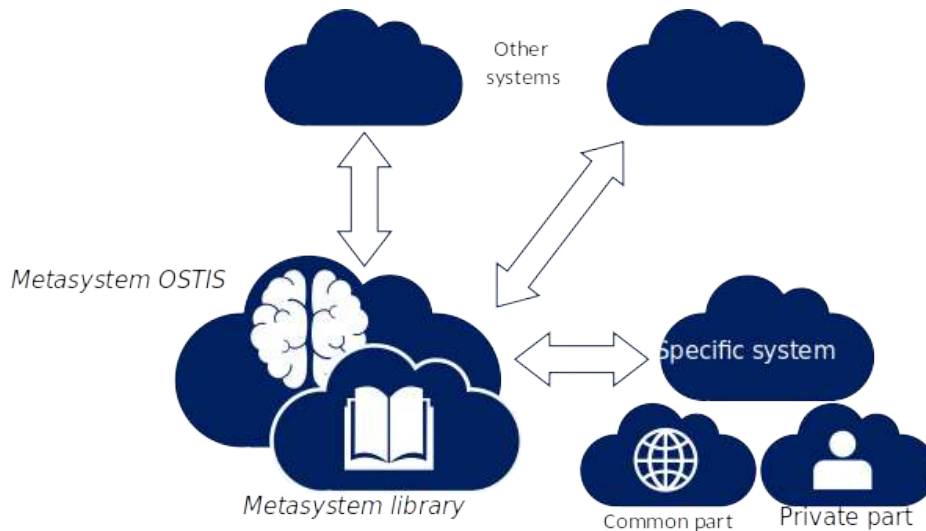


Figure 1. OSTIS Ecosystem Architecture

- [violation of the generality of concepts and in the generality of basic knowledge]
 - ⊃ *threat. destruction of knowledge base semantics (semantic viruses)*
 - ⇒ *explanation**:
 - [substitution or removal of nodes and links between them in the knowledge base]
 - ⊃ *threat. excessive amount of incoming information*
 - ⊃ *threat. breach of non-repudiation*
 - ⇒ *explanation**:
 - [issuance of unauthorized actions as legal, as well as concealment or substitution of information about the actions of subjects]
 - ⊃ *threat. breach of accountability*
 - ⇒ *explanation**:
 - [unauthorized or erroneous change, distortion or destruction of information about the performance of actions by the subject]
 - ⊃ *threat. violation of authenticity (authenticity)*
 - ⇒ *explanation**:
 - [performing actions in the system on behalf of another person or issuing unreliable resources (including data) as genuine]
 - ⊃ *threat. breach of credibility*
 - ⇒ *explanation**:
 - [intentional or unintentional provision and use of erroneous (incorrect) or irrelevant (at a specific point in time) information, as well as the implementation of procedures in violation of the regulations (protocol)]

Let's present the main directions of ensuring the information security of ostis-systems to prevent emerging threats:

- limitation of information traffic analyzed by the

- intelligent system;
- policy of differentiation of access to the knowledge base;
- connectivity;
- introduction of semantic metrics;
- semantic compatibility;
- activity.

It should be noted that at the design stage of the OSTIS technology itself, the basic principles of ensuring information security were already laid down as part of the design of individual components of the system. So already initially, support for semantic compatibility and cohesion is provided in ostis systems due to the system's ability to detect malicious processes in the knowledge base.

Restriction of information traffic analyzed by the intelligent system

The exponential growth of the volume of information circulating in information flows and resources under the conditions of well-defined quantitative restrictions on the capabilities of the means of its perception, storage, transmission and transformation forms a new class of information security threats characterized by the redundancy of the total incoming information traffic of intelligent systems.

As a result, the overflow of information resources of an intelligent system with redundant information can provoke the spread of distorted (destructive semantic) information. The general methodology for protecting intelligent systems from excessive information traffic is carried out through the use of axiological filters that implement the functions of numerical assessment of the value of incoming information, selection of the most valuable and screening (filtering) of less valuable (useless or harmful) using well-defined criteria.

Active means of destroying the semantics of knowledge bases (semantic viruses) should also be singled out as a separate category of information security threats [6].

Knowledge base access control policy

Mandatory access control (MAC) is based on mandatory (forced) access control, which is determined by four conditions: all subjects and objects of the system are identified; a lattice of information security levels is specified; each object of the system is assigned a security level that determines the importance of the information contained in it; each subject of the system is assigned an access level that determines the level of trust in him in the intellectual system. In addition, the mandate policy has a higher degree of reliability. The implementation of this policy is based on the developed algorithm for determining the agreed security levels for all elements of the ontology.

Since semantic knowledge bases, unlike a relational database, allow executing rules for obtaining logical conclusions, it is relevant to ensure data security by developing algorithms and methods that can only receive data that have security levels less than the access levels of the subjects who requested them [7].

Connectivity

All information stored in the semantic memory of the intelligent system is systematized in the form of a single knowledge base. Such information includes directly processed knowledge, interpreted programs, formulations of tasks to be solved, plans and protocols for solving problems, information about users, a description of the syntax and semantics of external languages, a description of the user interface, and much more [8]. In the information knowledge base between fragments of information (units of information), the possibility of establishing links of various types should be provided. First of all, these links can characterize the relationship between information units. Violation of connections leads to an incorrect logical conclusion, or to obtaining false knowledge, or to incompatibility of knowledge in the base.

Introduction of semantic metric

On a set of information units, in some cases it is useful to set a relation that characterizes the semantic proximity of information units, i.e. the force of the associative connection between information units [9]. It could be called the relevance relation for information units. This attitude makes it possible to single out some typical situations in the knowledge base. The relevance relation when working with information units allows you to find knowledge that is close to what has already been found.

Semantic Compatibility

Internal semantic compatibility between the components of an intelligent computer system (i.e., the maximum possible introduction of common, coinciding concepts for various fragments of a stored knowledge base), which is a form of convergence and deep integration within

an intelligent computer system for various types of knowledge and various problem solving models, which ensures effective implementation of the multimodality of an intelligent computer system. External semantic compatibility between various intelligent computer systems, which is expressed not only in the commonality of the concepts used, but also in the commonality of basic knowledge and is a necessary condition for ensuring a high level of socialization of intelligent computer systems [10].

Activity

In an intellectual system, the knowledge available in this system contributes to the actualization of certain actions. Thus, the execution of activities in an intelligent system should be initiated by the current state of the knowledge base. The appearance in the database of facts or descriptions of events, the establishment of links can become a source of system activity [11]. Including deliberate distortion of information and connections can become a source of deliberate distortion of information.

For new generation intelligent systems, there are a number of aspects that require the development of new algorithms and methods for ensuring information security in addition to existing mechanisms:

- multi-level access to individual parts of the knowledge base, as information can be public, personal, confidential;
- monitoring of changes in the meanings of words over time, as well as the meanings of translation from a foreign language that may influence decisions;
- protection against unauthorized use by using cryptosemantic ciphers;
- constant monitoring of vulnerabilities in the system;
- logging of actions (interactions) of the system.

To solve the tasks set, an expert ostis system can be used, which is capable of detecting abuses and anomalies in the behavior of all participants in the OSTIS Ecosystem based on continuous monitoring and the introduction of protocols for the interactions of participants.

The creation and application of expert systems is one of the important stages in the development of information technology and information security [12]. Accordingly, the solution to the problems of ensuring information security can be obtained based on the use of expert systems:

- it becomes possible to solve complex problems with the involvement of a new mathematical apparatus specially developed for these purposes (semantic networks, frames, fuzzy logic);
- the use of expert systems can significantly improve the efficiency, quality and efficiency of decisions through the accumulation of knowledge.

IV. CONCLUSION

For effective information protection of the system at the present stage, a symbiosis of traditional technologies

and technologies implemented within the framework of OSTIS is required. It should also be noted that ensuring information security based on OSTIS technology is much easier, because many aspects have already been implemented at the design stage of the technology itself. It is important to note that a new generation intelligent information system is an independent entity that can consciously, purposefully and constantly take care of itself, including its own security.

REFERENCES

- [1] S. Isoboev, D. Vezarko, and A. Chechel' nitskii, "Intellektual' naya sistema monitoringa bezopasnosti seti besprovodnoi svyazi na osnove mashinnogo obucheniya [intelligent system for monitoring the security of a wireless communication network based on machine learning]," *Ekonomika i kachestvo sistem svyazi*, no. 1(23), pp. 44–48, 2022.
- [2] A. Skrypnikov, V. Denisenko, E. Khitrov, I. Savchenko, and K. Evteeva, "Reshenie zadach informatsionnoi bezopasnosti s ispol'zovaniem iskusstvennogo intellekta [solving information security problems using artificial intelligence]," *Sovremennye naukoemkie tekhnologii*, no. 6, pp. 277–281, 7 2021.
- [3] V. Chastikova and A. Mityugov, "Metodika postroeniya sistemy analiza intsidentov informatsionnoi bezopasnosti na osnove neuroimmunnogo podkhoda [methodology for building an information security incident analysis system based on the neuroimmune approach]," *Elektronnyi Setevoi Politematicheskii Zhurnal «Nauchnye Trudy Kubgtu»*, no. 1, pp. 98–105, 2022.
- [4] D. D. Abdurakhman, "Iskusstvennyi intellekt i mashinnoe obuchenie v kiberbezopasnosti [artificial intelligence and machine learning in cybersecurity]," *Sovremennye problemy lingvistiki i metodiki prepodavaniya russkogo yazyka v vuze i shkole*, no. 34, pp. 916–921, 2022.
- [5] A. Ostroukh, *Intellektual' nye sistemy: monografiya*. Krasnoyarsk: Nauchno-innovatsionnyi tsentr, 2020.
- [6] A. Baranovich, "Semanticheskie aspekty informatsionnoi bezopasnosti: kontsentratsiya znaniy," *Istoriya i arkhivy*, no. 13(75), pp. 38–58, 2011.
- [7] V. Khoang and A. Tuzovskii, "Resheniya osnovnykh zadach v razrabotke programmy podderzhki bezopasnosti raboty s semanticheskimi bazami dannykh [solving the main tasks in the development of a security support program for working with semantic databases]," *Doklady TUSURa*, no. 2(28), pp. 121–125, 2013.
- [8] V. Glenkov, N. Guliakina, I. Davydenko, and D. Shunkevich, "Semanticheskaya model' predstavleniya i obrabotki baz znaniy [semantic model for representation and processing of knowledge bases]," L. Kalinichenko, Y. Manolopoulos, N. Skvortsova, and V. Sukhomlina, Eds. FIC IU RAN, 10 2017, pp. 412–419.
- [9] A. Dement'ev, "Metriki semanticheskikh dannykh [semantic data metrics]," *Molodoi uchenyi*, no. 24(419), pp. 48–51, 6 2022.
- [10] V. Golenkov, N. Guliakina, I. Davydenko, and A. Ereemeev, "Methods and tools for ensuring compatibility of computer systems," in *Open semantic technologies for intelligent systems*, ser. 4, V. Golenkov, Ed. BSUIR, Minsk, 2019, pp. 25–52.
- [11] V. Druzhinin and D. Ushakov, *Kognitivnaya psikhologiya. Ucheb-nik dlya vuzov [Cognitive psychology. Textbook for universities]*. M.: PER SE, 2002.
- [12] E. Sozinova, "Primenenie ekspertnykh sistem dlya analiza i otsenki informatsionnoi bezopasnosti [the use of expert systems for the analysis and evaluation of information security]," *Molodoi uchenyi*, no. 10(33), pp. 64–66, 10 2011. [Online]. Available: <https://moluch.ru/archive/33/3766/>

Обеспечение информационной безопасности Экосистемы OSTIS

Чертков В. М., Захаров В. В.

Большое разнообразие моделей обеспечения информационной безопасности, всё возрастающий объем данных, которые необходимо анализировать для обнаружения атак на информационные системы, изменчивость методов атак и динамическое изменение защищаемых информационных систем, необходимость оперативного реагирования на атаки, нечеткость критериев обнаружения атак и выбора методов и средств реагирования на них, нехватка высококвалифицированных специалистов по защите влечет за собой потребность в использовании методов искусственного интеллекта для решения задач безопасности.

В статье рассмотрены подходы к использованию искусственного интеллекта для обеспечения безопасности традиционных информационных систем, особенности обеспечения информационной безопасности интеллектуальных систем нового поколения и основные угрозы и принципы, лежащие в основе обеспечения информационной безопасности ostis-систем.

Received 13.03.2023

Semantic Approach to Designing Applications with Passwordless Authentication According to the FIDO2 Specification

Anton Zhidovich and Alexei Lubenko and Iosif Vojteshenko and Alexey Andrushevich
Belarusian State University
Minsk, Belarus
{anton.zhidovich, alexeilubenko02}@gmail.com, {voit, andrushevich}@bsu.by

Abstract—In this paper, a semantic approach to designing applications with the FIDO2 specification-based passwordless authentication using OSTIS technology is proposed. Obtained results will improve the efficiency of the component approach to the development of applications with passwordless authentication, as well as provide the ability to automatically synchronize different versions of components, increasing their compatibility and consistency.

Keywords—FIDO2 technology, passwordless authentication, OSTIS technology, biometrics

I. INTRODUCTION

When building an intelligent system, it's necessary to accord special priority to the issue of access to system resources and the differentiation of user rights. The key concept here is authentication – a procedure of identity verification to ensure that the user is the subject whose identifier he uses. The issue becomes more complicated when designing different semantically compatible intelligent systems [1], requiring a unified authentication apparatus: easy to use and integrate, as well as the most secure.

The authentication system is only a component, and therefore the development of a unified approach to its design is required. There are various authentication standards, many of which may not provide a high level of security and, moreover, may be compatible only with a certain software class or be proprietary.

II. ANALYSIS OF AUTHENTICATION METHODS

Let's take a look at the most common authentication methods. These methods can be encountered both in everyday life, with the use of messengers, online banking or other online services, and within corporate systems where data is accessed by company employees and delineated according to their position.

A. Password-based authentication

Being the most common because of its ease of implementation, password-based authentication method is vulnerable to the most types of attacks: brute-force, range attacks, dictionary attacks, key-logging, social engineering such as phishing, man-in-the-middle and replay attacks.

B. Trusted third-party authentication

The method is based on the fact that the service (provider) that owns the user's data, with his permission, provides third-party applications with secure access to this data. The provider is usually a service such as Google, GitHub, Facebook or Twitter. The most common implementation is the OAuth 2.0 protocol.

The OAuth 2.0 specification defines a protocol for delegating user authentication to the service that hosts a user account and authorising third-party applications to access that user account [2].

In [2] the main participants of OAuth 2.0 authentication and their interaction are described. Although this method is one of the most user-friendly and, moreover, implemented in most online resources, it is still vulnerable to a man-in-the-middle attack, which is a common and effective way to gain unauthorised access to a system.

C. One-time password (OTP)

OTP, which is used in many systems as a second or first authentication factor, can be a number or some string that is generated for a single login process. When authenticating, the OTP can be sent to the user via SMS-message, push notification or in a special application. The most secure tool for generating one-time passwords is a token (software, such as Google Authenticator, or hardware).

OTP quickly becomes invalid, which provides resistance to replay attacks. However, most attacks on authentication systems with OTP target the way the user receives it. For example, OTP transmitted via SMS can be intercepted by software such as FlexiSPY or Reptilicus.

One-time passwords are protected against phishing in the classic sense: users cannot reveal long-term credentials. However, the man-in-the-middle attack can be used to retrieve a currently valid one-time password.

D. Passwordless authentication methods

Passwordless authentication allows a user to access an information system without entering a password or answering security questions. Instead, the user provides

some other form of evidence such as a fingerprint, proximity badge, or hardware token code [3].

The essence of passwordless authentication is to never reveal any secrets. That is, everything about the user's identity and sensitive data remains protected.

There are several standards for passwordless authentication, depending on which factor is used and the available hardware, software and other features of the information system and its users. For example, biometric authentication involves comparing a user's unique biometric characteristics (facial features, retinal structure, etc.) with previously recorded samples of those characteristics. Biometric characteristics are inseparable from their owner, which ensures that it is impossible to refuse to log on and perform certain actions in the system. However, these authentication methods are also susceptible to hacking, such as face spoofing with a photo, 3D head model, etc. One way to combat such attacks on system security is liveness detection technology, which consists in checking the presented identifier for belonging to a "live" user and is designed to strengthen the identification procedure and protect against hacking in biometric authentication [4], [5].

The modern approach to passwordless authentication is the open standard FIDO2, jointly developed by the FIDO Alliance and the W3C consortium. The FIDO2 specification uses public-key cryptography and consists of two groups of standards. One of these is called the W3C WebAuthn standard. The second part is the Client to Authenticator Protocol (CTAP, CTAP1, CTAP2). The FIDO Alliance states that FIDO2 "reflects the industry's answer to the global password problem" by addressing legacy authentication's challenges as they pertain to security, usability, privacy, and scalability [6].

FIDO2-based authentication has advantages, primarily related to usability and security. FIDO2 can be used for passwordless login to the application or as an additional authentication factor, while ensuring a sufficient level of security for most tasks [7], [8]. The following advantages can be pointed out:

- The use of public-key cryptography provides resistance to phishing and man-in-the-middle attacks. Indeed, even by intercepting the public key or any data during registration, a fraudster cannot create a digital signature without access to the authenticator. The user authenticator itself is either built into the operating system, where the protection of the credentials (private key) is organised at the hardware level, or is an external device.
- Generating a special random byte buffer each time the server communicates with the authenticator prevents replay attacks.
- Users have a simple built-in mechanism like a fingerprint scanner to provide fast, secure, and convenient access to online services [9].

- Many operating systems and browsers have built-in support for WebAuthn API, which greatly simplifies the implementation of the technology in information systems.
- Cryptographic keys are unique for every website, providing users with enhanced privacy as sites cannot track the users across the web [9].

Thus, FIDO2-authentication is a modern, secure and convenient phishing-resistant method based on open standards and implemented in browsers and operating systems. The method provides ease of use by allowing users to register their device with a given online service through the selection of a local authentication mechanism. Speaking into the microphone, looking into the camera, inputting a PIN, or swiping a finger could all be valid local authentication mechanisms [10], [11].

III. SEMANTIC DEFINITION OF FIDO2 SPECIFICATIONS AND SECURITY KEYS

In order to increase the level of convergence and subsequent integration of a unified authentication system with next-generation intelligent computer systems, this paper proposes a semantic approach to their designing based on OSTIS technology. The OSTIS technology is a set of models, methods and tools permanently developed as part of an open project oriented on the ontological design, production, operation and re-engineering of semantically compatible hybrid intelligent computer systems that can independently interact with each other [12]. A number of languages are used to represent knowledge bases of ostis-systems (systems built on OSTIS Technology). Among them are external languages [12]:

- SCg (Semantic Code graphical) – a language whose texts represent a graph structures of a general type with precisely defined denotational semantics.
- SCn (Semantic Code natural) – a language for the structured external representation of SC-code texts.

A. Definition of FIDO2-authentication specifications using SC-code

W3C WebAuthn

:= [a specification developed by the FIDO alliance and W3C that allows an application to register and authenticate users using public-key cryptography instead of a password.]

:= [WebAPI built into platforms and browsers of all common operating systems to support passwordless authentication]

CTAP

⇒ *decoding**:

[Client to Authenticator Protocol]

:= [a specification that describes how the client (mobile app or web browser) and operating

system interact with cross-platform (physical) authenticators via USB, BLE, and NFC]

⇒ *levels**:

- {• *Authenticator API*
:= [level that represents a specific set of authenticator functions used for generating new credentials, confirming authentication and cancelling current operations]
- *Message Encoding*
:= [level at which all requests to the Authenticator API level are generated and encrypted]
- *Transport-specific Binding*
:= [level at which requests and responses to the external authenticator are transmitted via USB, BLE, NFC]

⇒ *subdividing**:

- {• *CTAP1*
:= [a protocol that describes client interaction with legacy authenticators as a second authentication factor]
- *CTAP2*
:= [a protocol that describes client interaction with new authenticators for passwordless access, two-factor authentication or multi-factor authentication]

The interaction of multiple parties is defined by the specifications included in FIDO2. Each party performs specific functions depending on its role in the authentication process:

FIDO2-authentication process participants

- = {• *user*
- *client*
- *credentials*
:= [the pair of private and public keys associated with the user account]
- *security key*
- *relying party*
:= [a server that stores the public key associated with the user account, makes requests to the WebAuthn client, and verifies the authentication signature]

B. Definition of security keys in the WebAuthn specification using SC-code

In general, a security key (also called authenticator) is assumed to have only one user. If multiple natural persons share access to an authenticator, they are considered to represent the same user in the context of that authenticator. If an authenticator implementation supports multiple users in separated compartments, then each compartment is considered a separate authenticator with a single user with no access to other users' credentials [13].

Platform authenticators have a built-in Trusted Platform Module (TPM) used to secure any generated private keys and are often biometric in nature, although this is certainly not a requirement. When present however, the biometric element provides a mechanism for the platform to match against the device's identity profile of a user, and in turn, use the stored cryptographic credentials to authenticate against a relying party. When it is not present, the same outcome can be achieved with other methods such as PINs for instance, although potentially, may be less secure [14].

The following definition of security keys can be made in SC-code.

Security key

- ⇒ *synonyms**:
[authenticator, WebAuthn key]
- := [a software component built into the operating system or external device that supports FIDO2 authentication]
- ⇒ *subdividing**:
 - {• *cross-platform (roaming) security key*
:= [an external physical device, not tied to a specific platform (operating system), used for authentication on multiple devices]
 - ⇒ *example**:
 - {• *YubiKey, developed by Yubico*
 - *Titan Security Key, developed by Google*
 - *OneKey, developed by CryptoTrust*
 - *platform (internal) security key*
:= [a software module, implemented either as a separate application or at the operating system level, used for authentication on a single device]
 - ⇒ *example**:
 - {• *Windows Hello for Windows 10/11*
 - *Touch ID, Face ID for IOS/macOS*

- *Graphic key, iris scanner, PIN code, etc. for Android*

C. Definition of FIDO certification process using SC-code

At the moment, all common operating systems support passwordless FIDO2-authentication methods using internal and roaming authenticators, certified by FIDO. The authentication tool undergoes a multi-step FIDO certification, confirming compliance of the implementation with FIDO2 specifications and defining a security level (L1, L2, L3) depending on resistance to various types of cyberattacks and built-in software and hardware protection of credentials [15]. The FIDO certification process can be represented as follows in SC-code:

FIDO certification process

- ```
⇒ steps*:
{
• Confirmation of compliance with FIDO specifications
• Functional compatibility testing
• Obtaining an authenticator security certificate of at least L1 level
• Certification submission
• Obtaining a trademark FIDO® Certified
• FIDO Metadata Service Registration
}
```

The FIDO Authenticator Certification levels can be represented in the following SC-code:

#### **Authenticator Certification Levels**

- ```
⇒ {
• L1
⇒ defence*:
[phishing and majority of scalable attacks]
⇒ example*:
[YubiKey 5 developed by Yubico]
• L2
⇒ defence*:
[remote software attacks]
⇒ example*:
[vFido developed by SecureMetric Technology]
• L3
⇒ defence*:
[remote software and local hardware attacks]
⇒ example*:
[de.fac2 developed by German Federal Office for Information Security]
}
```

As shown in [16], authenticators with L1 certification level can be vulnerable to timing attacks on FIDO2-authentication, which can allow attackers to link user accounts on multiple resources. Such authenticators may not guarantee a high level of security and privacy. This vulnerability has now been patched for major browsers (Mozilla Firefox [17], Google Chrome [18]).

IV. REGISTERING A WEBAUTHN KEY AND AUTHENTICATING WITH IT

The WebAuthn API allows applications to create and use secure credentials based on public-key cryptography with limited scope for user authentication. These credentials are created and saved in protected memory by the authenticator.

There are two operations defined in the WebAuthn API: creation of new user credentials and authentication using existing ones.

A. Registering new user credentials

Registration of new credentials can be done either for an authorised user (using a password, other authenticator or any available factor) or when creating a new account. Figure 1 shows the sequence of actions of the participants in the process of creating and registering new credentials. Let's look at the implementation using a WEB application as an example.

The user initiates the creation of new credentials while being on the WEB application page. The relying party's server sends a request to the client. The request contains a challenge - a buffer of cryptographically random bytes generated on the server, as well as information about the user (the identifier) and information about itself. The request data goes to the JavaScript program executed by the client (browser). The protocol for communicating with the server, as well as creating the request in the required format, are beyond the scope of the WebAuthn specification and depend on the platform on which the server runs and the libraries used.

The JS program makes a request to the authenticator to create new credentials. The request must contain data received from the server, information about the authenticator being used (internal or roaming), information about the expected credentials (such as the type of cryptographic algorithm used by the authenticator to create the signature), and the name of the new credentials (displayed for the user). The request is passed as parameters to the authenticator via the WebAuthn API built-in to the browser.

After the authenticator receives the data, the user authentication phase begins. If the verification is successful, the authenticator creates a new asymmetric key pair, saves the private key in protected storage on the device and generates a message to the client. The message contains the unique identifier of the created credentials, the address of the original web application, the challenge buffer, and

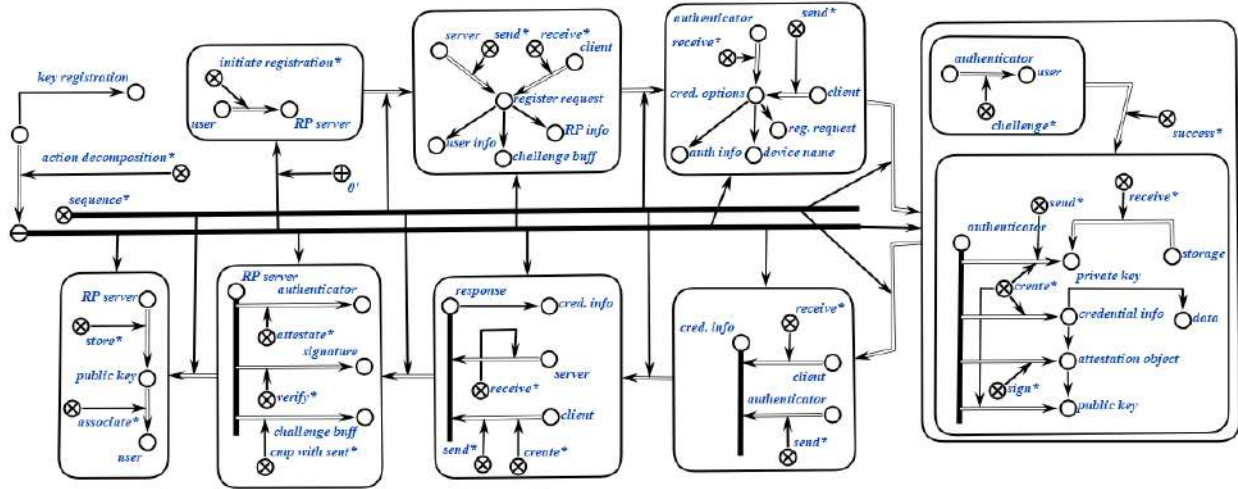


Figure 1. Key registration flow.

a special attestation object containing the public key and other information to be verified by the authenticator. The attestation object is signed with the private key.

The authenticator returns the data to the browser and the data is sent to the JavaScript program. The program creates a response to the server from the received data and then sends it to the server.

After receiving the data, the server must perform a series of checks to ensure that the registration process was passed and the data have not been modified: compare the received and sent values of the challenge buffer (they must be the same), verify the signature and perform attestation using the certificates set by the specific authenticator. If all the steps are successful, the server will store the received public key and associate it with the user account. The public key will be used later in user authentication .

B. Authentication using registered credentials

After a user has registered the WebAuthn key, it can be used in passwordless authentication. The authentication flow is much the same as the registration, but has its own peculiarities. The main difference is that the authenticator creates a specific authorization response signed with the private key instead of creating a new key pair. In a simplified form, the authentication process is shown in Figure 2.

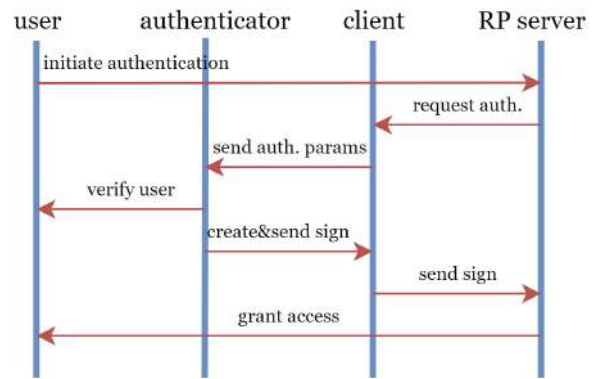


Figure 2. Authentication flow.

V. ARCHITECTURE OF APPLICATION WITH FIDO2-AUTHENTICATION

Based on the analysis above, and using [19], we propose to formalise the architecture of an application with passwordless authentication based on the FIDO2 specification in the form of the following tree:

- (0) Application architecture $S = A * B * C * D$
- (1) Client platform $A = E * F * G$
- (1.1) Client $E = H * I$

- (1.1.1) Client-side JS-program H : H_1 (Processing of the RP request), H_2 (Accessing the WebAuthn API)
- (1.1.2) WebAuthn API in the client I : I_1 (Authenticator registration), I_2 (Signature creation)
- (1.2) Authentication API built into the platform F : F_1 (Locating the authenticator), F_2 (Connecting to the authenticator)
- (1.3) Internal authenticator platform G : G_1 (Windows), G_2 (Android), G_3 (IOS), G_4 (MacOS)
- (2) Roaming authenticator standard $B = P * T$
- (2.1) Protocols P : P_1 (CTAP1), P_2 (CTAP2)
- (2.2) Transports T : T_1 (Bluetooth Low Energy (BLE)), T_2 (NFC), T_3 (USB)
- (3) Relying party $C = J * K * L$
- (3.1) Server application J : J_1 (Relying party operations), J_2 (Basic logic)
- (3.2) Library API interfaces K : K_1 (RSK FIDO2 Lib for ASP.NET), K_2 (Yubico libfido2)

- (3.3) FIDO2 server $L = M * N$
- (3.3.1) User & key storage M : M_1 (internal), M_2 (external)
- (3.3.2) Attestation trust store N : N_1 (Full basic), N_2 (Surrogate basic), N_3 (With a privacy certification centre)
- (4) External metadata service D : D_1 (Metadata Service v2.0 (MDS2)), D_2 (MDS3)

VI. CONCLUSION

In this paper a semantic approach to designing an application with FIDO2 authentication is described. The most common authentication methods have been analyzed and their common vulnerabilities identified. Using the capabilities of the OSTIS technology – SCn- and SCg-languages – the main components of FIDO2, as well as the WebAuthn key registration and authentication flows were described. The proposed approach enhances the efficiency of integration of FIDO2-authentication components in multi-agent intelligent systems.

Most modern operating systems and browsers have built-in support for FIDO2-authentication, in particular, have support for WebAuthn API, which is user-friendly, accessible, and protected from many types of attacks. In combination with the proposed approach, the technology can be easily integrated into OSTIS-systems, providing a single authentication apparatus.

In the context of authentication, an ontology that defines concepts — user, client, authenticator, certificates and levels of security, credentials, relying party, and its actions — and the relationships between them are presented. This will help to standardise the authentication process and ensure that the different devices and systems interact correctly. Standardisation of the authentication process improves functional compatibility and interoperability of systems and helps in the development and integration of new components of the OSTIS ecosystem.

ACKNOWLEDGMENT

The authors thank the teams of the Department of Software Engineering of the Belarusian State University and the Department of Intellectual Information Technologies of the Belarusian State University of Informatics and Radioelectronics for their help and valuable comments.

REFERENCES

- [1] V. Chertkov, "Information security in intelligent semantic systems," *Otkrytye semanticheskie tehnologii proektirovaniya intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, pp. 417–420, 2022.
- [2] (2023, Feb) An Introduction to OAuth 2. [Online]. Available: <https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>
- [3] (2022, Dec) Passwordless Authentication. [Online]. Available: <https://www.cyberark.com/what-is/passwordless-authentication/>
- [4] S. Policepatil and S. M. Hatture, "Face liveness detection: An overview," *International Journal of Scientific Research in Science and Technology*, vol. 8, no. 4, pp. 22–29, Jul. 2021. [Online]. Available: <https://doi.org/10.32628/IJSRST21843>

- [5] V. Zolotarev, A. Povazhnyuk, and E. Maro, "Metody usileniya procedury identifikatsii pol'zovatelej na osnove tekhnologii liveness detection [Methods of strengthening the user identification procedure based on liveness detection technology]," *Izvestiya SFedU. Engineering Sciences*, vol. 14, no. 2, pp. 212–225, May 2022. [Online]. Available: <https://doi.org/10.18522/2311-3103-2022-2-212-225>
- [6] (2023, Jan) FIDO2 Web Authentication. [Online]. Available: <https://www.hypr.com/security-encyclopedia/fido2-web-authentication>
- [7] A. M. Kadan and E. R. Kirichonok, "Authentication module based on the protocol of zero-knowledge proof," *Selected Papers of the V International Scientific and Practical Conference "Distance Learning Technologies"*, pp. 365–373, 2020. [Online]. Available: <https://CEUR-WS.org/Vol-2914/paper34.pdf>
- [8] A. Pathak, T. Patil, S. Pawar, P. Raut, and S. Khairnar, "Secure authentication using zero knowledge proof," in *2021 Asian Conference on Innovation in Technology (ASIANCON)*, 2021, pp. 1–8.
- [9] (2022, Nov) What is FIDO2 and Its Benefits. [Online]. Available: <https://www.kensington.com/news/security-blog/what-is-fido2-and-its-benefits/>
- [10] (2023, Jan) FIDO Authentication with WebAuthn. [Online]. Available: <https://auth0.com/docs/secure/multi-factor-authentication/fido-authentication-with-webauthn>
- [11] (2023, Dec) Pros and Cons of FIDO authentication. [Online]. Available: <https://www.securemetric.com/2019/05/17/pros-and-cons-of-fido-authentication/>
- [12] V. Golenkov, N. Guliakina, and D. Shunkevich, *Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems*. BSUIR, Minsk, 2021.
- [13] (2022, Jan) Webauthn Specification. [Online]. Available: <https://w3c.github.io/webauthn/>
- [14] (2023, Feb) Platform and Roaming Authenticators. [Online]. Available: https://developers.yubico.com/Developer_Program/WebAuthn_Starter_Kit/Platform_and_Roaming_Authenticators.html
- [15] (2022, Dec) Functional Certification. [Online]. Available: <https://fidoalliance.org/certification/functional-certification/>
- [16] M. Kepkowski, "How not to handle keys: Timing attacks on fido authenticator privacy," *22nd Privacy Enhancing Technologies Symposium*, pp. 705–726, 2022.
- [17] (2023, Jan) FIDO2/WebAuthn privacy leak through a timing attack using silent authentications. [Online]. Available: https://bugzilla.mozilla.org/show_bug.cgi?id=1730434
- [18] (2023, Jan) Stable Channel Update for Desktop. [Online]. Available: <https://chromereleases.googleblog.com/2021/11/stable-channel-update-for-desktop.html>
- [19] (2023, Feb) WebAuthn Introduction. [Online]. Available: <https://developers.yubico.com/WebAuthn/>

Семантический подход к проектированию приложений с беспарольной аутентификацией по спецификации FIDO2

Жидович А. А., Лубенько А. А.,
Войтешенко И. С., Андрушевич А. А.

В работе предложен семантический подход к проектированию приложений с беспарольной аутентификацией по спецификации FIDO2 на основе использования компонентов и средств технологии OSTIS. Приведено формальное описание спецификаций и ключей безопасности FIDO2.

Полученные результаты позволяют повысить эффективность компонентного подхода к разработке приложений с беспарольной аутентификацией, а также обеспечить возможность автоматической синхронизации различных версий компонентов, повышая совместимость и согласованность.

Received 14.03.2023

Mathematical methods for assessing information security risks

Alexsander Sobol and Viktor Kochyn
Center of Information technologies
Belarussian State University
Minsk, Belarus
Email: sobolam@bsu.by, kochyn@bsu.by

Natalia Grakova
Department of Intelligent Information Technologies
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: grakova@bsuir.by

Abstract—The article discusses international standards for assessing information security risks, as well as mathematical methods. The semantic model of information security risk assessment of the organization is described. For each element the calculation of the value is given.

Keywords—risk analysis, information security, mathematical model

I. INTRODUCTION

Nowadays, more and more organizations are exposed to cyberattacks. In order to reduce the impact of cyber attacks on the organization, it is necessary to assess information security risks. Although there are different approaches to assessing these risks, all methods require human involvement in any case.

The purpose of a risk assessment system is to establish an objective measurement of the level of risk that allows organizations to understand the business risks associated with critical information and assets, both qualitatively and quantitatively. Ultimately, risk assessment systems provide the tools necessary to make business decisions regarding investments in people, processes, and technology in order to reduce risk to an acceptable level.

This paper presents an overview of various current methodologies and models for information security risk assessment (ISRA). Particular attention is paid to game-theoretic and probabilistic-graphic methods and the construction of a semantic model of risk assessment, depending on the mathematical method used. The purpose of this article is to establish the rules of assessment, the objectives for the actors involved, the terminology used to describe risk, and the quantitative and qualification criteria [1]. In addition, the risk assessment methodology allows the comparison of risk degrees and defines the documentation to be collected and prepared based on the results of the assessment and follow-up.

II. CLASSIC SECURITY RISK ASSESSMENT METHODOLOGIES

To determine the necessary evaluation criteria, let's review the international standards related to risk assessment and use them to formulate the necessary criteria for assessing information security risks.

A. ISO 31000

ISO 31000 is a security risk assessment standard whose universal approach makes it applicable to a wide range of organizations and systems, regardless of their type or size [2].

The ISO 31000 process consists of several steps, as shown "Fig. 1".

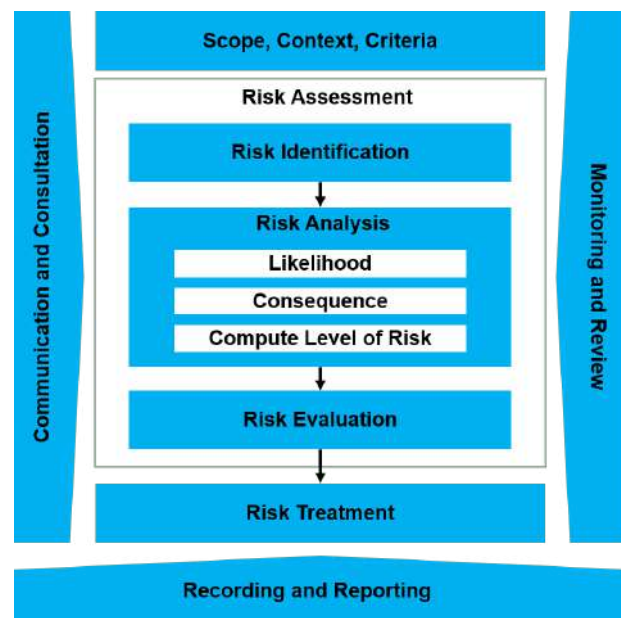


Figure 1. Risk management process.

Let's define each step:

- Establishing the context. This step identifies the internal and external contexts. The internal context defines the management, organizational structure, processes, roles and responsibilities, policies, project objectives, assets and other characteristics of the organization. The external context includes social, cultural, political, legal, regulatory, financial, technological, economic and natural aspects. They are used to build an understanding of the interests of stakeholders inside and outside the organization,

determine the purpose of the risk assessment, and identify risk criteria (what risks to consider and how to assess them).

- Risk Assessment. This step involves a complete process of risk identification, analysis, and assessment. Risk identification - defines sources of risk, areas of influence, risk events and their consequences. The main result of this part is a list of all possible risks. The next step is risk analysis. For each item of the list the corresponding probability of occurrence and potential consequences (impact) is defined. Risk analysis focuses on identifying risk sources, communicating with stakeholders to gather the information needed to make a decision, and evaluating risks against risk criteria. Risk analysis can be quantitative, semi-quantitative or qualitative in nature. During the assessment phase, risks are ranked for further processing.
- Risk treatment. This step involves selecting and implementing one or more options to change (mitigate) risks. ISO 31000 identifies a number of possible strategies that include deploying additional safety controls, shifting responsibility for risk mitigation to other organizations, changing the specifics of the organization, and accepting risk as it is. Each step of the assessment should involve communication with stakeholders and subject matter experts. The approach is considered iterative and should be repeated periodically.

ISO 31000 offers a high-level description of the process. This level of precision allows it to be adopted for a large number of different systems/organizations. However, most of the process steps are not detailed and rely solely on stakeholder decision-making and peer review. This standard does not contain any explicit suggestions for possible automation of the decision-making process at each step, nor does it define how to quantify risks. The standard assumes that the relevant relationships between systems and possible post-attack consequences must be taken into account in the risk assessment. However, the standard does not provide a clear procedure or algorithm to be followed.

B. ISO 27005

ISO 27005 [1] belongs to the ISO 27000 family of standards and is an extended risk management process that is specifically adapted to information security requirements. The standard itself contains a description of the ISRA process, which is still applicable in information security and infrastructure security. The ISO 27005 process itself is shown in “Fig. 2”. The first step of the analysis is to establish the context. It identifies the following objects of analysis:

- Definition of impact criteria, such as financial, human or reputational losses. Impact criteria may vary

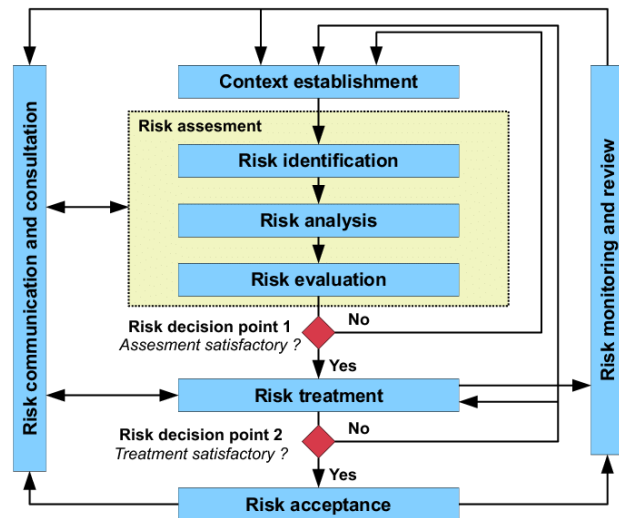


Figure 2. Information security risk management process.

depending on the specifics of the protected system and are defined as real parameters describing the level of damage caused. The usual range of impact criteria is from 0 to 5, where the scale is determined by the security personnel conducting the analysis.

- Defining the scope, which is the organization’s overall view of the assets to be protected. Assets can be both tangible (buildings, computers or personnel) and intangible (data, functions).
- If the assessment is planned as part of some standard process within the organization, it is important to identify the unit responsible for carrying it out.

The next step in the analysis is the risk assessment, which has three parts.

- Risk identification examines what incidents are likely to occur and which ones will result in a loss. This includes both those incidents that are under the organization’s control and those that are based on external influences. Identifying possible threats includes listing critical system assets (which can be tangible or intangible); identifying major vulnerabilities and how they are exploited; creating a set of possible “controls,” that is, means to mitigate or prevent an attack; identifying the various types of consequences of an attack, such as money, reputation, human loss, or any other relevant criteria.
- Risk analysis analyzes threats in terms of their probability of occurrence and numerically evaluates them against a set of criteria defined in the previous step. Risk analysis can be either qualitative or quantitative, depending on the type of criterion under consideration. For example, some criteria can be defined as an exact numerical value or range of values (financial losses), some criteria can be

characterized using descriptive scales (e.g., "low," "medium," "high"). Probability assessment looks at how often a particular threat is likely to occur and how easily the vulnerabilities associated with it can be exploited; it is conducted by experts in the field using an event log.

- The risk assessment re-examines the list of threats given the assessed criteria and probabilities. Based on the expert assessment, risks are ranked and a list of priority threats is compiled.

The next step in the process is risk treatment, described as a complex step consisting of four possible options: risk modification, risk retention, risk avoidance, and risk sharing. These options are not mutually exclusive; a combination of strategies is possible. Risk modification involves choosing (new) controls to reduce risk. This decision must be made in the light of available resources. The remaining strategies identify different measures to transfer responsibility for handling risk to another strategy or to accept low-impact risks as they are.

The final step in the analysis is risk acceptance, where all identified risks are reassessed against the chosen treatment strategies and accepted or not based on residuals (estimated exposure levels after treatment). ISO 27005 provides a high-level risk assessment algorithm and defines the vocabulary and basic elements of the ISRA process. It is more detailed than ISO 31000, defining the exact types of entities to be analyzed during the contextualization phase. However, it does not define a meaningful approach for quantifying risks - the standard relies heavily on expert judgment at each stage of analysis and provides no suggestions or guidance for detailed formalization of attacks or any means for automating decision-making during risk identification and risk analysis.

III. ISRA CRITERIA

Based on the international standards considered, the following features can be identified, which are important for the assumed mathematical model of ISRA. The described criteria are considered as necessary conditions, for simplicity of simplification of experts in the field of ISRA.

1. General description of the system. The approach must be applicable to any type of system and must interact with heterogeneous entities, which can be adjusted according to the information security expert's area of interest.
2. General descriptions of attacks. The approach should allow the description of arbitrary threat scenarios and attack types. Attacks can be interdependent, and the scenario can include several stages of development.
3. Versatility of protection description. The approach must be capable of implementing a variety of information security controls and information protection tools.

This list is not necessarily sufficient for a quality ISRA. However, meeting these requirements will allow the model to be technically applied to ISRA problems. Many additional constraints and requirements can be added, such as the need to explicitly model sequential countermeasures, deflect simultaneous attacks, and consider different types of exposure. Fulfilling these criteria will require a certain level of simplification compared to models that consider specific attacks on a particular system. In addition to defining the modeling criteria, it is necessary to describe the specific application of the model being developed (i.e., the group of end users and the qualifications required). The end users of the model are information security personnel and specialists who perform ISRA.

IV. ISRA MATHEMATICAL METHODS

The mathematical methods used for ISRA are diverse. They cover a wide range of methods, including formalized techniques [3], evolving into intermediate expert assessments with supporting formal structures (e.g., attack trees and correlation diagrams), [4] and converging to fully expert decision tree methods used to classify or quantify security levels.

A. Main Families of Models

The main methods of applied mathematics in the field of ISRA are: big data analysis and statistical learning, algorithms of operations research and statistical graph models. Let us consider them in more detail.

1. Data mining and statistical learning. These methods have been successfully applied in intrusion detection. The methods developed are usually formalized as anomaly detection problems, solved by methods such as principal component analysis [5], single-class support vector machines (SVMs) [6], or kernel density estimators [7]. The advantage of these methods is that there is no need to develop a detailed system model, since the training procedure aims to adapt the parametric model depending on the specifics. However, these algorithms are used for specific tasks with a narrow domain, and exploitation is performed at the system component level using special equipment (e.g., network analyzers, protocol scanners and malware detectors or botnets). Algorithms based on statistics depend on the training data and on the structure of the chosen algorithm, which significantly narrows the scope of their application. An exception in this case may be self-learning online models that can adjust to new data. Nevertheless, these methods still require the data to be presented in the standard format of generic descriptive information security "trait-vectors", which is a separate problem that needs to be solved. Another problem with statistical methods, especially the more complex examples (e.g., deep neural networks), is the poor interpretation of results and difficult error analysis/debugging.

2. Algorithms for information security problems based on operations research. They are usually developed for large-scale information security systems. These methods take as inputs a formal description of the system, a set of asset variables to be optimized (e.g., security control allocation), and one (potentially multiple) objective function. The target function must be optimized for the model variables. The adversary is usually represented explicitly as an entity, and its goals can be explicitly modeled using formalization by means of game theory or decision theory. Existing methodologies may vary depending on the asset and system variables modeled, but within operations research (and game theory in particular) provides a general concept that can be used to support decision making during ISRA in large-scale systems. In addition, the explicit definition of the target function simplifies the interpretation of the results.

3. A family of statistical graph models [8]. This method can be understood as a combination of statistical analysis and operations research algorithms. They are used in security tasks to identify failures of subsystems [9], as well as for cybersecurity [10]. In contrast to big data analysis models, statistical graph models have a preliminary design phase that aims to capture the specifics of the system being described. The models establish an explicit probabilistic relationship between the data observed during operation and the description of the system state (e.g., the failure of certain modules or the possible presence of an attack). Thus, graph models allow, for example, to derive a probabilistic distribution of system state variables depending on the observed data.

It should be noted that in recent years, the aforementioned distinction between the use of operations research methods and big data analysis has become increasingly flexible. One of the main reasons for this is related to the developing field of generative adversarial networks (GANs) [11]. GANs can be thought of as a combination of operations research methods and data mining. Some theoretical settings simulate a zero-sum game between a generator and a classifier [12], the intention being to teach the generator to produce objects that are indistinguishable from the "real" training dataset. However, this does not explicitly aim to model the behavior between the attacker and the defender, nor does it address the problem of poor interpretation. In addition to the methods already discussed, there are a number of approaches to solve general information security problems, including general attack detection methods, as presented in [13].

B. Semantic assessment of information security risks

The semantic model of information security risk assessment can be represented as the following function:

$$R = f(V, T, I) \quad (1)$$

where: V — probability of threat occurrence; T — severity of consequences; I — cost of risk reduction.

This function takes as input the three components of the model and calculates the information security risk borne by the system or a component of the system. Appropriate functions can be used to calculate each component, which can be determined from experience and statistical data.

The function for calculating the probability of the threat is as follows:

$$V = V(P, C, E) \quad (2)$$

where: P — theoretical threat probability (this value from 0 to 1 depends on the formula we choose: Bayesian, Monte Carlo, etc.); C — the degree of possibility of the threat occurrence (this value from 0 to 10 can be obtained based on the vulnerabilities of information systems, which are evaluated according to the CVSS metric); E — the motivation of the attacker.

The motivation can be varied, for example financial, political, hacking, revenge, or sabotage. All of these motivations can take different forms and manifest themselves in different types of cyber attacks. To defend against cyber threats more effectively, it is necessary to understand the motivations of attackers and take them into account when developing an information security strategy. In general, it can be difficult to formalize an assessment of an attacker's motivation using mathematical methods, because motivations can be very diverse and are often social, political or psychological in nature. However, the use of machine learning algorithms and big data analysis can help automate the process of assessing the motivation of attackers. An attacker's motivation score can be expressed as a numerical value on a particular scale, such as 0 to 10, where 0 is no motivation and 10 is the maximum motivation. A numerical estimate of motivation can be derived from an analysis of various factors affecting the attacker, such as:

- the potential benefit of successfully executing the attack (e.g., financial gain, gaining a competitive advantage, revenge, etc.);
- complexity of the attack (e.g., skills, tools available to the attacker, etc);
- whether the attacker can be detected and punished (e.g., the probability that the attack will be detected and the attacker will be eliminated);
- value of the target to the attacker (e.g., value of sensitive data, company reputation, etc.).

Based on the analysis of these factors, a motivation factor can be determined to represent the numerical value of the attacker's motivation. For example, a formula to estimate an attacker's motivation might look like this:

$$E = \frac{B * Diff * ValueTarg}{Detect} \quad (3)$$

where: B — the potential benefit of successfully executing the attack; $Diff$ — the difficulty of the attack;

ValueTarg — the value of the target to the attacker; *Detect* — the probability of identifying and punishing the attacker.

A function for calculating the severity of consequences could look as follows:

Calculation of consequence severity (T):

$$T = T(II, A, F) \quad (4)$$

where: *II* — is the importance of the information that may be stolen, lost, or damaged (importance); *A* — likelihood of the consequence occurring (likelihood); *F* — the degree of impact on business processes (impact factor).

Quantifying the importance of information (II) can be done by using mathematical modeling and data analysis techniques. One such method is Business Impact Analysis (BIA).

BIA is a process that is used to assess the importance of information to business processes and to determine the potential consequences of its loss or breach of confidentiality. The evaluation of information importance in BIA is based on two main criteria: business importance and confidentiality.

Different levels of importance can be defined for each criterion, which can be expressed as numbers from 1 to 10. For example, business importance levels may include the following values:

- critically important: 9-10;
- very important: 7-8;
- medium importance: 4-6;
- minor: 1-3.

Similar levels can be defined for the privacy criterion.

Further, for each business process, the importance of each element of information related to the process can be defined. The importance of each element can be expressed by a number from 1 to 10, where 1 is low importance and 10 is high importance.

The final importance of each information can be determined by multiplying the importance of each element by the importance of the business process and the importance of confidentiality. The probability of consequences refers to the likelihood that a particular event or threat will result in undesirable consequences for the system or organization.

Various methods can be used to estimate the probability of consequences, including statistics, expert judgment, and modeling. A quantitative estimate of probability can be expressed as a number, usually on an interval between 0 and 1, where 0 means that probability is impossible and 1 means that probability is absolutely certain.

The degree of impact on business processes can be estimated using the following formula:

$$Impact = \sum_{i=1}^n (Asset_i * \frac{Loss_i}{Revenue}) \quad (5)$$

where *n* is the number of assets, *Asset_i* is the value of *i*-th asset, *Loss_i* is the potential loss when consequences for *i*-th asset occur, *Revenue* is the company's annual income. Thus, the value of *Impact* shows what part of the annual income of the company can be lost as a result of the possible consequences for its assets. The higher the value of *Impact*, the more serious are the consequences for business processes.

The following formula can be used to quantify the probability of a consequence occurring:

$$P = Mot * U * II \quad (6)$$

where *Mot* is the probability of the attacker's motivation; *U* is the probability of system vulnerability, and *II* is the importance of the information.

Each of the components of the formula can be evaluated on an interval from 0 to 1, where 0 means that the probability is impossible and 1 means that the probability is absolutely certain.

Estimating the probability of an attacker's motivation can be done through expert judgment, historical data analysis, or research. An assessment of the likelihood of system vulnerability can be based on statistical data analysis, vulnerability studies, or expert assessments. An assessment of the importance of information can be based on business process analysis, expert evaluations, or data classification. Each of these components can be evaluated based on an analysis of the importance of the data stored in the system, the likelihood of consequences occurring, and the extent to which those consequences affect business processes.

Each of these components can be estimated based on an analysis of the importance of the data stored in the system, the likelihood of the consequences and the impact of these consequences on business processes.

Calculation of risk reduction cost (I):

$$I = I(Cost, RR) \quad (7)$$

where: *Cost* — the cost of taking risk reduction measures (cost); *RR* — the probability of risk reduction when taking action (effectiveness).

The cost of taking risk reduction measures can be estimated using the following formula:

$$Cost = \sum_{i=1}^n (Asset_i * \frac{Loss_i * Risk_i}{SecurityBudget}) \quad (8)$$

where *n* is the number of assets, *Asset_i* is the value of *i*th asset, *Risk_i* is the probability of threat occurrence for *i*th asset, *Loss_i* is the potential damage when consequences occur for *i*th asset, *SecurityBudget* is the budget for information security.

Thus, the value of *Cost* shows how much money needs to be spent on risk mitigation measures for all company assets. Estimating the cost helps to decide what risk

mitigation measures should be implemented based on the information security budget.

The probability of reducing risk by taking action can be estimated using the following formula:

The cost of taking risk reduction measures can be estimated using the following formula:

$$RR = \frac{Risk_{before} - Risk_{after}}{Risk_{before}} * 100 \quad (9)$$

where $Risk_{before}$ is the probability of threat occurrence before taking measures, $Risk_{after}$ is the probability of threat occurrence after taking measures.

Thus, the value of $RiskReduction$ shows by how much the probability of threat occurrence will decrease when measures to reduce risk are taken. This assessment helps to determine the effectiveness of the measures taken to reduce the risk and decide whether additional measures are needed.

REFERENCES

Information security risk assessment is an important element of information resource management in any organization, since information security is directly related to its value, confidentiality and integrity. There are many methods of risk assessment, and each of them has its own advantages and disadvantages.

One approach to risk assessment is the use of mathematical methods and formulas. They allow you to assess the likelihood of the threat, the motivation of the attacker, the importance of the information, the impact on business processes, the degree of impact of the consequences and other factors that may affect the security of information.

By using machine learning algorithms, it is possible to automate the risk assessment process and improve the accuracy of the assessment. In this case, machine learning models can be used to analyze large amounts of data, search for hidden patterns, determine parameters and identify dependencies between them. Machine learning can also be used to create predictive models and scenario analyses.

However, despite all the advantages, mathematical methods and machine learning models are not a universal solution to the problem of information security. It is necessary to take into account that each model has its own limitations and shortcomings, and cannot take into account all possible factors that may affect information security.

Therefore, a comprehensive approach to risk assessment is necessary, which includes the use of several assessment methods, analysis of results, regular updating and improvement of models based on new data and improvement of information security practices.

In general, mathematical methods and machine learning models are an important tool for information security

risk assessment, but only in combination with other information security methods and practices, such as physical security, access management, personnel training, etc.

REFERENCES

- [1] ISO/IEC 2018. Technical report
- [2] A. Dali and C. Lajtha. Iso 31000 risk management: "the gold standard". EDPACS, 45(5):1–8, May 2012.
- [3] M. Brown, B. An, C. Kiekintveld, F. Ordonez and M. Tambe. Multi-objective optimization for security games. In Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems. *International Foundation for Autonomous Agents and Multiagent Systems*, Richland, SC, 2012, Vol. 2, AAMAS '12, pp. 863–870.
- [4] B. Schneier. Attack Trees — Modeling security threats. *Dr. Dobbs's Journal*, 1999.
- [5] T. Morita, S. Yogo, M. Koike, T. Hamaguchi, S. Jung, I. Koshijima, and Y. Hashimoto. Detection of cyber-attacks with zone dividing and PCA. *Procedia Computer Science*, 22:727 – 736, 2013. 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems — KES2013.
- [6] Y. Wang, J. Wong and A. Miner. Anomaly intrusion detection using one class SVM. In Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004., pp. 358 – 364.
- [7] Y. Cao, H. He, H. Man and X. Shen. Integration of Self-Organizing Map (SOM) and Kernel Density Estimation (KDE) for network intrusion detection. Edward M. Carapezza (ed.), *Unmanned/Unattended Sensors and Sensor Networks VI*, International Society for Optics and Photonics, SPIE, 2009, vol. 7480, pp. 146–157.
- [8] C. Borgelt and R. Kruse *Graphical Models: Methods for Data Analysis and Mining*. John Wiley and Sons, Inc., USA, 01 2002.
- [9] A. Abdollahi, K. R. Pattipati, A. Kodali, S. Singh, S. Zhang and P. B. Luh. *Probabilistic Graphical Models for Fault Diagnosis in Complex Systems*, Springer International Publishing, Cham, 2016, pp. 109–139.
- [10] P. Xie, J. Li, X. Ou, P. Liu and R. Levy. Using Bayesian networks for cyber security analysis. In 2010 IEEE/IFIP International Conference on Dependable Systems Networks (DSN), 2010, pp. 211–220.
- [11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, 2020.
- [12] F. A. Oliehoek, R. Savani, J. Gallego-Posada, E. van der Pol, E. D. de Jong and R. Gross. GANGs: Generative adversarial network games, 2017.
- [13] F. Pasqualetti, F. Dörfler and F. Bullo. Attack detection and identification in cyberphysical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, Nov 2013.

Математические методы анализа риска информационной безопасности

Соболь А. М., Кочин В. П., Гракова Н. В.

В статье рассматриваются международные стандарты для оценки рисков информационной безопасности, а также математические методы. Описана семантическая модель оценки рисков информационной безопасности организации. Для каждого элемента приведен расчет значения.

Received 01.04.2023

AUTHOR INDEX

A		Aliaksandr Kroshchanka	127
Alexey Andrushevich	213, 311	Aliaksandr Kupo	219
B		Alexander Kurbatsky	33
Valeria Bahai	159	L	
Kseniya Bantsevich	213	Wenzu Li	269
Danila Bernat	159	Nikolai Listopad	279
Dzianis Bialiauski	165	Alexei Lubenko	311
Anatoliy Boriskevich	247	M	
Lizaveta Bushchik	279	Jun Ma	247
C		Natallia Malinovskaya	213
Vitalij Chachlou	165	Vadim Matskevich	133
Alena Cherkas	219	O	
Valery Chertkov	305	Maksim Orlov	191
E		Arkadzi Oskin	263
Aleksandr Erofeev	285	P	
G		Boris Panshin	27
Vladimir Golenkov	15, 39	Anton Paramonov	61
Natalia Grakova	257, 317	Vladislav Prokhorenko	291
Natalia Gulyakina	15, 39	Q	
H		Longwei Qian	171
Katsiaryna Halavataya	159	R	
Aliaksandr Halavaty	159, 257	Vadim Rodchenko	147
Juras Hetsevich	165	Vladimir Rostovtsev	237
I		S	
Dzmitry Ivaniuk	291	Mikhail Sadowski	153
Valerian Ivashenko	39, 95	Sergei Samodumkin	141, 299
K		Vladislav Savionok	181, 225
Liesia Kajharodava	165	Daniil Shunkevich	39, 115
Anna Karkanitsa	147	Viktor Smorodin	291
Alena Kazlova	257	Alexsander Sobol	317
Victar Kochyn	33, 317	T	
Mikhail Kovalev	121, 213	Valery Taberko	291
Victor Krasnoproshin	133, 147	Valery Taranchuk	181, 225

Mikhail Tatur	61
Aliaksandr Trafimau	165
Vitaly Tsishchanka	213
Viktar Tsviatkou	247
V	
Uladzimir Vishniakou	241
Iosif Vojteshenko	311
X	
Yiwei Xia	241
Y	
Chuyue Yu	241
Z	
Alexandr Zagorskiy	207
Vadim Zahariev	39, 165
Vladimir Zakharau	305
Kuanysh Zhaksylyk	165
Anton Zhidovich	311
Yauheniya Zianouka	165
Nikita Zotov	67, 77

АВТОРСКИЙ УКАЗАТЕЛЬ

А		Ивей С.	241
Андрушевич А. А.	213, 311	К	
Б		Кайгородова Л.	165
Багай В. Д.	159	Карканица А. В.	147
Банцевич К. А.	213	Ковалёв М. В.	121, 213
Белявский Д.	165	Козлова Е. И.	257
Бернат Д. А.	159	Кочин В. П.	33, 317
Борискевич А. А.	247	Краснопрошин В. В.	133, 147
Бущик Е. А.	279	Крощенко А. А.	127
В		Купо А. Н.	219
Вишняков В. А.	241	Курбацкий А. Н.	33
Войтешенко И. С.	311	Л	
Г		Ли В.	269
Гецевич Ю.	165	Листопад Н. И.	279
Голенков В. В.	15, 39	Лубенько А. А.	311
Головатая Е. А.	159	М	
Головатый А. И.	159, 257	Ма Ц.	247
Гракова Н. В.	257, 317	Малиновская Н. В.	213
Гулякина Н. А.	15, 39	Мацкевич В. В.	133
Е		О	
Ерофеев А. А.	285	Орлов М. К.	191
Ж		Оськин А. Ф.	263
Жаксылык К.	165	П	
Жидович А. А.	311	Паньшин Б. Н.	27
З		Парамонов А. И.	61
Загорский А. Г.	207	Прохоренко В. А.	291
Захаров В. В.	305	Р	
Захарьев В. А.	39, 165	Родченко В. Г.	147
Зеновко Е.	165	Ростовцев В. Н.	237
Зотов Н. В.	67, 77	С	
И		Савёнок В. А.	181, 225
Иванюк Д. С.	291	Садовский М. Е.	153
Ивашенко В. П.	39, 95	Самодумкин С. А.	141, 299

Сморозин В. С.	291
Соболь А. М.	317
Т	
Таберко В. В.	291
Таранчук В. Б.	181, 225
Татур М. М.	61
Тищенко В. Н.	213
Трофимов А.	165
Х	
Хохлов В.	165
Ц	
Цветков В. Ю.	247
Цянь Л.	171
Ч	
Черкас Е. О.	219
Чертков В. М.	305
Чуюэ Ю.	241
Ш	
Шункевич Д. В.	39, 115

Научное издание

**Открытые семантические технологии
проектирования интеллектуальных систем**

**Open Semantic Technologies
for Intelligent Systems**

Сборник научных трудов

Основан в 2017 году

Выпуск 7

В авторской редакции

Ответственный за выпуск *В. В. Голенков*
Компьютерная вёрстка *Н. В. Гракова*

Подписано в печать 13.04.2023. Формат 60*84 1/8.
Бумага офсетная. Гарнитура «Таймс». Ризография.
Усл. печ. л. 39,53. Уч.-изд. л. 34,65.
Тираж 105 экз. Заказ 56.

Полиграфическое исполнение
УП «Бестпринт». Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий №1/160 от 27.01.2014.
Ул. Филатова, д. 9, к. 1, 220026, г. Минск



OSTIS-
2024

14th International Scientific and Technical Conference

«Open Semantic Technologies for Intelligent Systems»

Open Semantic Technologies for Intelligent Systems

April 18-20, 2024 Minsk, Republic of Belarus

CALL FOR PAPERS

We invite you to take part in XIV International Scientific and Technical Conference “Open Semantic Technologies for Intelligent Systems” (OSTIS-2024), which will focus on fields of use of semantic technologies.

The conference will take place from *April, 18th to April, 20th, 2024* at the Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus.

The language of the research papers collection: English

Working languages of the conference: Russian, Belarusian, English

MAIN ORGANIZERS OF THE CONFERENCE

- Ministry of Education of the Republic of Belarus
- Ministry of Communications and Informatization of the Republic of Belarus
- Belarusian State University of Informatics and Radioelectronics (BSUIR)
- State Scientific Institution «The United Institute of Informatics Problems of the National Academy of Sciences of Belarus» (UIIP NASB)
- State Institution “Administration of High Technologies Park” (Republic of Belarus)
- Educational-scientific association in the direction of “Artificial Intelligence” (ESA-AI)
- Belarusian State University (BSU)
- Brest State Technical University (BrSTU)
- Minsk State Linguistic University (MSLU)
- Polotsk State University named after Euphrosyne of Polotsk
- Grodno State University named after Yanka Kupala
- Belarusian State Medical University
- Russian Association of Artificial Intelligence (RAAI)
- Belarusian Public Association of Artificial Intelligence Specialists (BPA of Artificial Intelligence Specialists)

CONFERENCE TOPICS:

- *Principles underlying the semantics-based knowledge representation and their unification. Knowledge types and features of the semantics-based representation of various knowledge and metaknowledge types. Relations between knowledge; relations, that are defined on the knowledge. Semantic structure of a global knowledge base, that integrates various accumulated knowledge.*
- *Parallel-oriented programming languages for processing of the semantics-based representation of knowledge bases.*
- *Problem-solving models, that are based on knowledge processing, which occurs directly at the semantics-based level for representing knowledge being processed. Semantic models of information retrieval, knowledge integration, correctness and quality analysis of knowledge bases, garbage collection, knowledge base optimization, deductive and inductive inference in knowledge bases, plausible reasoning, pattern recognition, intelligent control. Integration of various problem-solving models.*
- *Semantic models of environment information perception and its translation into the knowledge base.*
- *Semantic models of multimodal user interfaces of intelligent systems, based on the semantic representation of knowledge used by them and unification of such models.*
- *Semantic models of natural language user interfaces of intelligent systems. The structure of semantic representation of linguistic knowledge bases, which describe natural languages and facilitate solution of natural language text and speech interpretation problems, as well as natural language texts and speech messages synthesis, that are semantically equal to certain knowledge base fragments.*
- *Integrated logical-semantic models of intelligent systems, based on semantic knowledge representation, and their unification*
- *Various technical platforms and implementation variants of unified logical-semantic models of intelligent systems, based on semantic knowledge representation.*
- *Models and means, that are based on the semantic knowledge representation and that are oriented on the design of various typical components of intelligent systems (knowledge bases, programs, problem solvers, user interfaces).*

- *Models and means, that are based on semantic knowledge representation and that are oriented on the complex design of various classes of intelligent systems (intelligent reference systems, intelligent learning systems, intelligent control systems, intelligent robotics systems, intelligent systems for design support, etc.)*
- *Applied intelligent systems, that are based on the semantic representation of knowledge used by them*

CONFERENCE GOALS AND FORMAT

The goal of the conference is to discuss the problems of creating an **Open Comprehensive Semantic Technology for Hybrid Intelligent Systems Design**. This determines the Conference format, which involves wide discussion of various issues of creating such a technology and poster sessions.

During the **poster sessions**, every participant of the conference will have an opportunity to demonstrate their results. The conference format assumes exact start time of each report and exact time of its exhibition presentation.

One of the major objectives of the conference is to attract not only scientists and postgraduate students but also students who are interested in Artificial intelligence, as well as commercial organizations willing to collaborate with research groups working on the development of modern technologies for intelligent systems design.

PARTICIPATION TERMS AND CONDITIONS

All those interested in Artificial intelligence problems, as well as commercial organizations willing to collaborate with research groups working on the development of modern technologies for intelligent systems design, are invited to take part in the Conference.

To participate in the OSTIS-2024 conference, it is necessary to register in the [CMT](#) system **before March 18, 2024**, find a conference page, and from there:

- submit a **participation form** for the OSTIS conference. Each participation form field is required to be filled in, including indication of the reporter. By filling in the registration form, you agree that your personal data will be processed by the Organizing Committee of the Conference and the paper and information about the authors will be published in printed and electronic format. A participation form should contain information on all of the authors. If author(s) are participating with a report, a participation form should contain their **color photo(s)** attached (they are required for the Conference Program);

- upload an **article** for publication in the OSTIS Research Papers Collection. Papers should be formatted according to the provided template (see <https://proc.ostis.net/for-authors/>). Four full pages is a minimum size of a paper.
- send the signed **scan of the letter of consent**.

If a report is submitted to participate in one of the contests, this intention should be clearly indicated in the participation form.

The selection of papers for publication in the Research Papers Collection and participation in the Conference is performed by a number of reviewers from among the members of the Conference Program Committee.

Incompliant applications and papers will be rejected.

Conference participation does not require any fees.

PAPERS SUBMISSION PROCEDURE

Papers (only on topics mentioned above) should be submitted ready for publication (<http://proc.ostis.net> -> For authors). The text should be logically complete and contain new scientific and practical results. Each author is allowed to submit two reports maximum.

After receiving the article, it is sent for review. The authors can get acquainted with the results of the review in [CMT](#), if necessary, correct the comments of the reviewers, and send them for re-review.

The Organizing Committee reserves the right to reject any paper, if it does not meet the formatting requirements and the Conference topics, as well as if there was no participation form submitted for the paper.

YOUNG SCIENTIST REPORTS CONTEST

The co-authors of the report submitted to the competition of reports of young scientists can be scientists with degrees and titles, but authors under the age of 35 who do not have degrees and titles must directly submit the report.

To take part in the young scientist report contest, it is necessary to:

- 1) fill in the participation form, where your participation in the contest is clearly indicated;
- 2) write an article and upload it to the [CMT](#) website;
- 3) fill in, sign, scan, and send a letter of consent via the email;
- 4) make a report at the conference (in person).

YOUNG SCIENTIST PROJECTS CONTEST

Projects of applied intelligent systems and systems aimed at supporting the design of intelligent systems are allowed to take part in the contest; they have to be presented by a scientist without a degree and under the age of 35.

To take part in the young scientist projects contest, it is necessary to:

- 1) fill in the participation form, where your participation in the contest is clearly indicated;
- 2) write an article and upload it to the [CMT](#) website;
- 3) make a report at the conference (in person);
- 4) make an exhibition presentation of the software.

STUDENT INTELLIGENT SYSTEM PROJECTS CONTEST

To participate in the contest, a project must meet the following criteria: (a) to be developed by students and/or undergraduates of the higher education institutions and (b) project consultants and advisors must hold a scientific degree and title. To participate in this contest, it is necessary to:

- 1) familiarize yourself with contest's terms and conditions (<http://conf.ostis.net>);
- 2) fill in the participation form for the contest (<http://conf.ostis.net>);
- 3) prepare a summary of the project (<http://conf.ostis.net>);
- 4) submit the participation form and project summary to the student projects' email address: ostis.stud@gmail.com.

CONFERENCE PROCEEDINGS PUBLICATION

The Conference Organizing Committee plans to publish the papers, selected by the Program Committee based on the results of their review, in the Conference Proceedings on the official Conference website <http://conf.ostis.net> and on the Conference Proceedings website <http://proc.ostis.net>.

Upon successful review an author sends a letter of consent to the Organizational Committee. The author therefore agrees that their paper can be made freely available in electronic form at other resources at the Editorial Board's discretion.

Since 2020, the OSTIS Research Papers Collection has been included in the List of Scientific Publications of the Republic of Belarus for publishing the results of dissertation research (List of the Higher Attestation Commission of the Republic of Belarus) in the technical field of sciences (informatics, computer technology, and management).

In addition, following the results of the conference, it is planned to publish the Selected Papers Collection of the OSTIS conference in the series "Communications in Computer and Information Science" (CCIS) published by Springer. Detailed information about this can be found on the conference website (<http://conf.ostis.net>).

KEY DATES OF THE CONFERENCE

February 18, 2024	The beginning of the submission of materials for participation in the conference
March 18, 2024	A deadline for receiving materials for participation in the conference by the

	Organizing Committee
<i>March 28, 2024</i>	A deadline for submitting reviews of articles
<i>April 08, 2024</i>	A final decision on paper publication; sending out invitations and notifications on inclusion of a paper in the OSTIS Research Papers Collection
<i>April 15, 2024</i>	Draft Conference Program publication on the conference website: http://conf.ostis.net
<i>April 17, 2024</i>	Research Papers Collection publication on the conference website: http://proc.ostis.net
<i>April 18, 2024</i>	Participant registration and the opening of the OSTIS-2024 Conference
<i>April 18-20, 2024</i>	OSTIS-2024 Conference
<i>May 2, 2024</i>	Photo-report and conference report publication on the conference website: http://conf.ostis.net
<i>May 18, 2024</i>	Research Papers Collection uploading to the Russian Science Citation Index database

CONFERENCE PROGRAM FORMATION

Conference program is formed by the Program Committee according to the paper review results; author(s)' confirmation of participation is required as well.

CONTACTS

All the necessary information about the forthcoming and previous OSTIS Conferences can be found on the conference website <http://conf.ostis.net> and <http://proc.ostis.net>.

For questions regarding conference participation and dispute resolution, please contact: ostisconf@gmail.com.

Methodological and advisory support to the conference participants shall be provided through the conference e-mail only.

The conference venue is the 5th academic building of the Belarusian State University of Informatics and Radioelectronics (Platonov Str., 39, Minsk, Republic of Belarus).



OSTIS-
2024

**XIV международная научно-техническая конференция
«Открытые семантические технологии
проектирования интеллектуальных систем»**

Open Semantic Technologies for Intelligent Systems

18 – 20 апреля 2024 г. Минск. Республика Беларусь

И Н Ф О Р М А Ц И О Н Н О Е П И С Ь М О

Приглашаем принять участие в XIV Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2024), которая будет посвящена вопросам области применения семантических технологий.

Конференция пройдет в период с *18 по 20 апреля 2024* года в Белорусском государственном университете информатики и радиоэлектроники, г. Минск, Республика Беларусь.

Язык статей сборника научных трудов: английский

Рабочие языки конференции: русский, белорусский, английский.

ОСНОВНЫЕ ОРГАНИЗАТОРЫ КОНФЕРЕНЦИИ

- Министерство образования Республики Беларусь
- Министерство связи и информатизации Республики Беларусь
- Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»
- Государственное научное учреждение «Объединенный институт проблем информатики Национальной академии наук Беларуси»
- Государственное учреждение «Администрация Парка высоких технологий» (Республика Беларусь)
- Учебно-научное объединение по направлению «Искусственный интеллект» (УНО-ИИ)
- Белорусский государственный университет
- Учреждение образования «Брестский государственный технический университет»
- Минский государственный лингвистический университет
- Полоцкий государственный университет имени Евфросинии Полоцкой
- Гродненский государственный университет имени Янки Купалы
- Учреждение образования «Белорусский государственный медицинский университет»
- Российская ассоциация искусственного интеллекта (РАИИ)
- Белорусское общественное объединение специалистов в области искусственного интеллекта (БОИИ)

НАПРАВЛЕНИЯ РАБОТЫ КОНФЕРЕНЦИИ:

- *Принципы, лежащие в основе семантического представления знаний, и их унификация. Типология знаний и особенности семантического представления различного вида знаний и метазнаний. Связи между знаниями и отношения, заданные на множестве знаний. Семантическая структура глобальной базы знаний, интегрирующей различные накапливаемые знания*
- *Языки программирования, ориентированные на параллельную обработку семантического представления баз знаний*
- *Модели решения задач, в основе которых лежит обработка знаний, осуществляемая непосредственно на уровне семантического представления обрабатываемых знаний. Семантические модели информационного поиска, интеграции знаний, анализа корректности и качества баз знаний, сборки информационного мусора, оптимизации баз знаний, дедуктивного и индуктивного вывода в базах знаний, правдоподобных рассуждений, распознавания образов, интеллектуального управления. Интеграция различных моделей решения задач*
- *Семантические модели восприятия информации о внешней среде и отображения этой информации в базу знаний*
- *Семантические модели мультимодальных пользовательских интерфейсов интеллектуальных систем, в основе которых лежит семантическое представление используемых ими знаний, и унификация этих моделей*
- *Семантические модели естественно-языковых пользовательских интерфейсов интеллектуальных систем. Структура семантического представления лингвистических баз знаний, описывающих естественные языки и обеспечивающих решение задач понимания естественно-языковых текстов и речевых сообщений, а также задач синтеза естественно-языковых текстов и речевых сообщений, семантически эквивалентных*

заданным фрагментам баз знаний

- *Интегрированные комплексные логико-семантические модели интеллектуальных систем, основанные на семантическом представлении знаний, и их унификация*
- *Различные технические платформы и варианты реализации интерпретаторов унифицированных логико-семантических моделей интеллектуальных систем, основанных на семантическом представлении знаний*
- *Средства и методы, основанные на семантическом представлении знаний и ориентированные на проектирование различных типовых компонентов интеллектуальных систем (баз знаний, программ, решателей задач, интерфейсов)*
- *Средства и методы, основанные на семантическом представлении знаний и ориентированные на комплексное проектирование различных классов интеллектуальных систем (интеллектуальных справочных систем, интеллектуальных обучающих систем, интеллектуальных систем управления, интеллектуальных робототехнических систем, интеллектуальных систем поддержки проектирования и др.)*
- *Прикладные интеллектуальные системы, основанные на семантическом представлении используемых ими знаний*

ЦЕЛЬ И ФОРМАТ ПРОВЕДЕНИЯ КОНФЕРЕНЦИИ

Целью конференции является обсуждение проблем создания **открытой комплексной семантической технологии компонентного проектирования семантически совместимых гибридных интеллектуальных систем**. Этим определяется и формат её проведения, предполагающий широкое обсуждение различных вопросов создания указанной технологии и выставочные презентации докладов.

Выставочная презентация докладов даёт возможность каждому докладчику продемонстрировать результаты своей разработки на выставке. Формат проведения конференции предполагает точное время начала каждого доклада и точное время его выставочной презентации.

Важнейшей задачей конференции является привлечение к её работе не только учёных и аспирантов, но и студенческой молодежи, интересующейся проблемами искусственного интеллекта, а также коммерческих организаций, готовых сотрудничать с научными коллективами, работающими над интеллектуальными системами и созданием современных технологий и их проектированием.

УСЛОВИЯ УЧАСТИЯ В КОНФЕРЕНЦИИ

В конференции имеют право участвовать все те, кто интересуется проблемами искусственного интеллекта, а также коммерческие организации, готовые сотрудничать с научными коллективами, работающими над созданием современных технологий проектирования интеллектуальных систем.

Для участия в конференции OSTIS-2024 необходимо **до 18 марта 2024** года зарегистрироваться в системе [СМТ](#), найти страницу конференции и на ней:

- подать **заявку** на конференцию OSTIS. Каждое поле заявки обязательно для заполнения, в том числе указание того автора, кто будет представлять доклад. Заполняя регистрационную форму, Вы подтверждаете согласие на обработку Оргкомитетом конференции персональных данных, публикацию статей и информации об авторах в печатном и электронном виде. В заявке должна содержаться информация по каждому автору;
- загрузить **статью** для публикации в Сборнике научных трудов конференции OSTIS. Статья должна быть оформлена в соответствии с правилами оформления публикуемых материалов и занимать не менее 6 полностью заполненных страниц;
- загрузить **сканированный вариант письма о согласии** на публикацию и размещения передаваемых материалов в сети Интернет;
- загрузить **цветные фотографии** всех авторов статьи (это необходимо для оформления Программы конференции)

Если доклад представляется на конкурс докладов молодых учёных или на конкурс программных продуктов молодых учёных, это должно быть явно указано в заявке статьи (в [СМТ](#)).

Отбор статей для публикации в Сборнике и участия в работе конференции осуществляется рецензентами и редакционной коллегией сборника.

Заявки и статьи, оформленные без соблюдения предъявляемых требований, не

рассматриваются.

Участие в конференции не предполагает организационного взноса.

ПОРЯДОК ПРЕДСТАВЛЕНИЯ НАУЧНЫХ СТАТЕЙ

Статьи (только по перечисленным выше направлениям) представляются в готовом для публикации виде (<http://proc.ostis.net> -> Авторам). Текст статьи должен быть логически законченным и содержать новые научные и практические результаты. От одного автора допускается не более двух статей.

После получения статьи, она отправляется на рецензирование. С результатами рецензирования авторы могут ознакомиться в [СМТ](#), при необходимости устранить замечания рецензентов и отправить для повторного рецензирования.

Оргкомитет оставляет за собой право отказать в приеме статьи в случае, если статья не будет соответствовать требованиям оформления и тематике конференции, а также, если будет отсутствовать заявка доклада, соответствующая этой статье.

КОНКУРС ДОКЛАДОВ МОЛОДЫХ УЧЁНЫХ

Соавторами доклада, представляемого на конкурс докладов молодых учёных, могут быть учёные со степенями и званиями, но непосредственно представлять доклад должны авторы в возрасте до 35 лет, не имеющие степеней и званий.

Для того, чтобы принять участие в конкурсе научных докладов молодых учёных, необходимо:

- 1) заполнить заявку на участие в конференции, в которой чётко указать своё желание принять участие в данном конкурсе;
- 2) написать статью для публикации в Сборнике научных трудов и загрузить на сайте [СМТ](#);
- 3) заполнить, подписать, отсканировать и отправить по почте письмо о согласии;
- 4) лично представить доклад на конференции.

КОНКУРС ПРОЕКТОВ МОЛОДЫХ УЧЁНЫХ

Принимать участие в конкурсе проектов молодых учёных могут проекты прикладных интеллектуальных систем и систем, ориентированных на поддержку проектирования интеллектуальных систем, при этом представлять проект на конкурсе должен молодой учёный в возрасте до 35 лет, не имеющий учёной степени.

Для того, чтобы принять участие в конкурсе программных продуктов молодых учёных, необходимо:

- 1) заполнить заявку на участие в конференции), в которой чётко указать своё желание принять участие в данном конкурсе;
- 2) написать статью для публикации в Сборнике научных трудов и загрузить на сайте [СМТ](#);
- 3) лично представить доклад на конференции;
- 4) провести выставочную презентацию, разработанного программного продукта.

КОНКУРС СТУДЕНЧЕСКИХ ПРОЕКТОВ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

В конкурсе студенческих проектов могут принимать участие проекты, разработчиками которых являются студенты и магистранты высших учебных заведений, консультантами и руководителями проекта могут быть лица, имеющие научную степень и звание. Для того, чтобы принять участие в данном конкурсе, необходимо:

- 1) ознакомиться с положением о конкурсе студенческих проектов (<http://conf.ostis.net>);
- 2) заполнить заявку на участие в конкурсе студенческих проектов (<http://conf.ostis.net>);
- 3) подготовить описание проекта (<http://conf.ostis.net>).
- 4) выслать заявку на участие в конкурсе и описание проекта по электронному адресу конкурса студенческих проектов: ostis.stud@gmail.com.

ПУБЛИКАЦИЯ МАТЕРИАЛОВ КОНФЕРЕНЦИИ

Оргкомитет конференции предполагает публикацию статей, отобранных Программным комитетом по результатам их рецензирования, в Сборнике научных трудов OSTIS в печатном

виде и на официальном сайте сборника <http://proc.ostis.net> в электронном виде.

По результатам рецензирования автор отправляет оргкомитету письмо о согласии, которое предусматривает дальнейшую возможность размещения статей, вошедших в сборник научных трудов, в открытом электронном доступе на иных ресурсах по усмотрению редакции сборника.

С 2020 года Сборник научных трудов OSTIS включен в Перечень научных изданий Республики Беларусь для опубликования результатов диссертационных исследований (Перечень ВАК РБ) по технической отрасли наук (информатика, вычислительная техника и управление).

Кроме того, по итогам конференции планируется издание Сборника научных трудов OSTIS в серии «Communications in Computer and Information Science» (CCIS) издательства Springer. Подробная информация об этом приведена на сайте конференции (<http://conf.ostis.net>).

КЛЮЧЕВЫЕ ДАТЫ КОНФЕРЕНЦИИ

<i>18 февраля 2024 г.</i>	начало подачи материалов для участия в конференции
<i>18 марта 2024 г.</i>	срок получения материалов для участия в конференции Оргкомитетом
<i>28 марта 2024 г.</i>	срок предоставления рецензий на статьи
<i>08 апреля 2024 г.</i>	срок принятия решения о публикации присланных материалов и рассылки приглашений для участия в конференции и сообщение о включении статьи в Сборник научных трудов OSTIS
<i>15 апреля 2024 г.</i>	размещение на сайте конференции http://conf.ostis.net проекта Программы конференции OSTIS-2023
<i>17 апреля 2024 г.</i>	размещение на сайте конференции http://proc.ostis.net Сборника научных трудов OSTIS
<i>18 апреля 2024 г.</i>	регистрация участников и открытие конференции OSTIS-2024
<i>18-20 апреля 2024 г.</i>	работа конференции OSTIS-2024
<i>2 мая 2024 г.</i>	публикация фоторепортажа и отчёта о проведённой конференции на сайте конференции: http://conf.ostis.net
<i>18 мая 2024 г.</i>	загрузка материалов сборника конференции в РИНЦ

ФОРМИРОВАНИЕ ПРОГРАММЫ КОНФЕРЕНЦИИ

Программа конференции формируется Программным комитетом по результатам рецензирования, представленных статей, а также на основании подтверждения автора(-ов) статьи о прибытии на конференцию.

КОНТАКТНЫЕ ДАННЫЕ ОРГАНИЗАТОРОВ КОНФЕРЕНЦИИ OSTIS

Вся необходимая информация по предстоящей и предыдущих конференциях OSTIS находится на сайте конференции <http://conf.ostis.net>, а также на сайте материалов конференции <http://proc.ostis.net>.

По вопросам участия в конференции и решения спорных вопросов обращайтесь: ostisconf@gmail.com.

Методическая и консультативная помощь участникам конференции осуществляется только через электронную почту конференции.

Конференция проходит в Республике Беларусь, г. Минск.

Оргкомитет конференции находится на кафедре интеллектуальных информационных технологий Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники (БГУИР) — г. Минск, ул. Платонова, 39, 5-ый учебный корпус БГУИР.