OSTIS-2011
(Open Semantic Technologies for Intelligent Systems)

УДК 004.82:004.55

НАПРАВЛЕНИЯ, МЕТОДЫ И СРЕДСТВА ПРИМЕНЕНИЯ СЕМАНТИЧЕСКИХ СЕТЕЙ В INTERNET-ТЕХНОЛОГИЯХ

Д.Г. Колб (*kolb@bsuir.by*)
Белорусский государственный университет информатики и радиоэлектроники, г.Минск, Республика Беларусь

Приводятся принципы построения интеллектуальных систем для сети Internet, в основе которых лежат однородные семантические сети. Демонстрируется один из возможных способов перехода от традиционных гипертекстовых систем к системам семантически-структурированных гипертекстов и на основе последних к интеллектуальным системам, в основе которых лежат семантические сети.

Ключевые слова: семантически структурированный гипертекст, модель пользовательского интерфейса, языки описания мультимедиа данных, семантическая сеть, интеллектуальная система.

Введение

Объемы информации, которые накоплены в сети Internet за последние годы, и неэффективность методов её обработки указывают на необходимость использования более эффективных средств представления и обработки информации, чем традиционные. Одними из перспективных моделей, позволяющих решить указанную проблему, является модели представления знаний в основе которых лежат семантические сети. Наиболее перспективными для сети Internet и динамично развивающимися в настоящее время направлениями, связанными с семантическими сетями можно назвать:

- языки представления знаний, в основе которых лежат семантические сети;
- языки обработки знаний и языки запросов к базам знаний, представленным с помощью семантических сетей;
- хранилища баз знаний;
- инструментарии для реализации интеллектуальных систем, в основе которых лежат семантические сети.

Языки представления знаний, в основе которых лежат семантические сети. Первыми результатами в данной области можно назвать реализации для сети Internet формализованной модели гипертекста, основу такой реализации составлял язык HTML [W3C, 2010]. Новой ветвью развития языков представления знаний для сети Internet стало направление Semantic web [W3C, 2010], однако до появления этого направления уже существовали работы, которые были направлены на оптимизацию текущего представления знаний в виде HTML для сети Internet [Гаврилова, 2000].

Semantic web предполагает представление знаний в виде семантической сети с помощью онтологий. Основу технологий предлагаемых Semantic Web составляет семейство стандартов на языки описания, включающее XML, XML Schema, RDF, RDF Schema, OWL [Хорошевский, 2008].

Первые результаты работ в рамках направления Semantic web были использованы разработчиками пользовательских интерфейсов web-приложений для семантического

размещения типовых сущностей пользовательского интерфейса или его структуры. Основу таких подходов составлял язык XML. В настоящее время, разработкой языков описания пользовательского интерфейса и технологиями, поддерживающими такие языки, занимаются такие ведущие разработчики инструментальных для разработки программного обеспечения как Microsoft (XAML, MRML), Adobe (MXML, OpenLaszlo), Oracle (CookSwing, SwiXML, SwixNG, Thinlet, Ultrid, Vexi, XALXAL, XSWT, ZUML), Mozilla (XUL).

Однако только семантическая разметка структуры web-приложения не давала возможность реализовать эффективные способы поиска информации в рамках web-приложения. Одним из наиболее доступных по реализации подходов для решения данной проблемы стал подход, основанный на использовании микроформатов (microformats, μF или uF) – способа семантически размечать сведения о разнообразных сущностях на веб-страницах, используя стандартные элементы языка HTML (или XHTML) [Вики о микроформатах, 2010]. Наиболее известными микроформатами являются:

- hCard – для публикации контактной информации людей, компаний, организаций и мест;
- hCalendar – микроформат для представления семантической информации о событиях в формате календаря;
- hAtom – ленты новостей;
- XFN – социальные взаимоотношения.

В настоящее время микроформаты используются как для унифицированного представления однотипных сущностей пользовательского интерфейса веб-приложения, так и для настройки и адаптации пользовательского интерфейса web-браузеров к пользователю [Веб-фрагмент, 2010].

Несмотря на простоту использования микроформатного подхода, они не позволяли решить все задачи, которые ставило направление Semantic Web, поэтому одновременно с микрформатным подходом развивались подходы, в основе которых лежит использование RDF. Ярким примером реализации подхода на базе RDF является проект FOAF("Friend of a Friend") [FOAF,2010], который позволяет описывать отношение знакомства с помощью RDF. Пользовательский интерфейс, при таком подходе, генерируется на основании RDF-описаний с помощью специальных клиентов социальных сетей (FoaF Explorer, Origo), каждый сущности, описанной с помощью RDF, соответствует сущность, её визуализирующая, в пользовательском интерфейсе.

С ростом количества web-ресурсов поддерживающих стандарты Semantic web появилась проблема унифицированного оформления таких ресурсов. Решением такой проблемы является использование системы метаданных для представления знаний web-ресурсов. В настоящее время существует несколько десятков проектов, связанных с разработкой систем метаданных. Одним из популярных проектов, направленных на решения проблемы унификации представления знаний в виде семантических сетей, стал проект «Дублинское ядро»[DCMI, 2010]. Целью проекта стала разработка стандартов метаданных, которые были бы независимы от платформ и подходили бы для широкого спектра задач. Основными результатами проекта являются словари метаданных общего назначения, стандартизирующих описания ресурсов в формате RDF.

Одной из серьёзных проблем при разработке интеллектуальных систем для сети Internet является проблема глубинного web. Не смотря на то, что проблема выявлена в 2000 г. сейчас, по мнению разных авторов, к видимому web относится только 20-30% содержимого web-пространства. Существует несколько типов ресурсов глубинного web, основную массу таких ресурсов составляют ресурсы, представляющие собой мультимедиа информацию, так как в данное время еще не существует удовлетворительных алгоритмов поиска не текстовой информации [Ландэ, 2009].

Появление проблемы глубинного web частично связано с решением еще одной проблемы характерной для web-приложений – использования различных визуальных моделей в рамках одного web-приложения. Визуальные модели обладают особой познавательной силой, фактически предоставляя средства когнитивной графики для структурирования информации. На практике используется более сотни методов визуального структурирования [Касьянов,

2003], [Гаврилова, 2008]. Использование разных визуальных моделей предполагает совмещение в рамках одной сущности интерфейса различных синтаксически разнородных форм представления информации, обеспечивая тем самым мультимедийность пользовательского интерфейса. Однако при использовании данных технологий ни одна современная поисковая система не позволяет производить поиск по мультимедийным фрагментам, которые использует такое web-приложение, так как подход не позволяет проиндексировать мультимедиа.

Языки обработки знаний и языки запросов к базам знаний, представленным с помощью семантических сетей. Наиболее интересные практические наработки в этой области тесно связаны с направлением Semantic Web. В рамках направления можно выделить две группы языков запросов к базам знаний:

- SQL-подобные языки запросов к RDF/OWL-данным. Наиболее известным представителем таких языков язык запросов SPARQL (SPARQL Protocol and RDF Query Language)[SPARQL, 2010]. В настоящее этот язык поддерживается большинством хранилищ баз знаний, разработанных с использованием RDF.
- не SQL-подобные языков запросов к RDF/OWL-данным. Одним из представителей группы является язык запросов Versa [VERSA, 2010]. Это компактный функциональный язык программирования, синтаксис которого напоминает Lisp. К сожалению поддержка языка VERSA разработана только для Python, поэтому он не получил широкого распространения.

Наибольший интерес с точки зрения обработки знаний, хранящихся в виде семантических сетей, представляет проект Neno/Fhat [NENO/FHAT, 2010]. Neno – это язык программирования для семантических сетей, а Fhat – это виртуальная машина, для исполнения кода написанного на Neno. Идея авторов Neno заключается в том, чтобы позволить пользователям ассоциировать исполняемый код (методы) с OWL классами, тогда RDF ресурсы превращаются в полноценные объекты. Исходные тексты на Neno транслируются в промежуточный код, который представляет собой набор RDF-троек. Таким образом, исполняемый код может храниться в RDF-хранилище вместе с данными относящимися к ресурсу. Виртуальная машина Fhat может читать и исполнять этот код, что позволяет на базе такого подхода реализовывать распределенные вычислительные системы.

Хранилища баз знаний. В начале развития направления Semantic Web первые средства для работы с RDF-данными ориентировались в первую очередь на RDF-ресурсы, которые целиком помещались в оперативной памяти. Такой подход полностью устраивал разработчиков при создании небольших приложений. С развитием Semantic Web и накоплением больших массивов знаний в формате RDF, необходимо было решить проблему масштабирования баз знаний. Наиболее простым и относительно дешевым решением этой проблемы стало использование существующих решений для реляционных баз данных.

В средах Jena [Jena, 2010], Sesame[Sesame, 2010], которые позволяют эффективно использовать реляционные СУБД, предполагается, что обработка данных и логический вывод выполняются в основной памяти, что означает слабую связанность с базой данных. Однако практически установлено, что лучшая эффективность и масштабируемость выполнения запросов достигаются при использовании средств, опирающихся на возможности СУБД для логического вывода новых фактов [Hawk, 2010]. В текущих реализациях хранилищ баз знаний разработчикам, как правило, приходится находить компромисс между выразительностью поддерживаемых диалектов OWL и эффективностью, масштабируемостью хранения и доступа к RDF-данным. Обычно средства, слабо связанные с базой данных, поддерживают больше конструкций OWL, но для более ограниченных наборов данных, чем более тесно связанные или интегрированные решения [Левшин, 2009]. Однако заметим, что выбор хранилища наиболее целесообразно проводить в зависимости от текущей задачи.

Большие объемы данных и несовершенство методов их обработки заставляют разработчиков и бизнес приглядываться к альтернативам реляционных баз данных, используемым вот уже более тридцати лет. В совокупности все эти технологии известны как «NoSQL базы данных» Термин NoSQL появился в 1998 году - так назывался проект Карло Строцци, представлявший собой легковесную реляционную базу данных с открытым исходным кодом, которая была лишена

SQL интерфейса. В начале 2009 года этот термин был возрождён Эриком Эвансом, чтобы обозначить быстро растущее семейство не реляционных, распределённых хранилищ данных[NoSQL, 2010].

Начиная с 2009 года появилось большое количество работ по проектам в рамках направления NoSQL. NoSQL–технологии развиваются применительно к трем проблемным областям:

- горизонтальное масштабирование при больших объемах данных, например для таких проектов как Facebook (50 терабайт для поиска по входящим сообщениям) или eBay (2 петабайта в целом);
- производительность каждого отдельного сервера;
- не гибкий дизайн логической структуры.

Наиболее интересными из всех групп проектов NoSQL с точки зрения обработки семантических сетей является группа, так называемых СУБД на основе графов. Долгое время лидером в этом направлении был проект Neo4j [Neo4j,2010], однако на сегодняшний момент появилась целое семейство серьёзных конкурентов для Neo4j – Sones, InfoGrid, HyperGraphDB, AllegroGraph, Bigdata, DEX, Infinite, Graph, OpenLink, Virtuoso, VertexDB. Особенностью данных проектов является то, что СУБД полностью ориентирована на эффективную распределенную обработку сетевых структур данных. Кроме того ряд СУБД семейства поддерживает языки запросов, используемые в Semantic Web (SPARQL), или дают возможность выдавать запрос в виде RDF-графа.

Инструментарии для реализации интеллектуальных систем, в основе которых лежат семантические сети. В рамках данного направления можно выделить подходы, которые основаны на технологиях направления Semantic Web и альтернативные данному направлению технологии. К первой категории можно отнести инструментарии, которые позволяют сделать интеллектуальную систему “под ключ”. Основной характеристикой таких средств является в первую очередь реализация технологии проецирования объектов предметной области, представленных в RDF, на какой либо из объектно-ориентированных языков. Среди таких средств можно выделить три группы инструментариев:

- инструментарии, которые ориентированы на разные фреймворки для разработки интерфейсов web-приложений. К таким инструментариям можно отнести проекты Jena[Jena, 2010], Arc [Arc, 2010]. Arc – это проект для языка PHP, который ориентирован на работу с RDF-хранилищами на базе MySQL и позволяет легко себя интегрировать в существующие фреймворки для разработки web-приложений на языке PHP.
- инструментарии, которые ориентированы как на интеграцию с разными RDF-хранилищами, так и на интеграцию с фреймворками для разработки интерфейса web-приложений. Наиболее интересными в этом плане является проект Topaz [Topaz, 2010] на сегодняшний день он поддерживает работу с такими хранилищами как Sesame [Sesame, 2010], Mulgara [Mulgara, 2010], основной системой публикации web-контента в проекте Topaz является система Ambra, хотя сам проект ориентирован на поддержку любой системы публикации web-контента.
- инструментарии, в которые включены все элементы для разработки полноценной приложения для сети Internet. Одним из таких инструментариев является проект CubicWeb [CubicWeb, 2010], который развивается французской компанией LogiLab.

Вторую категорию инструментариев составляют проекты, которые являются альтернативой по идеологии для направления Semantic Web. Особенностью таких проектов является использование своих языковых и программных средств, возможность экспортирования или импортирования баз знаний для дальнейшей обработки средствами Semantic Web.

В настоящее время широко получил распространение ряд проектов, которые в литературных источниках известны под общим названием «Semantic wiki». Одним из представителей таких проектов является проект Semantic Mediawiki [Semantic Mediawiki, 2010]. Semantic MediaWiki – это дополнение проекта MediaWiki [MediaWiki, 2010]. В то время как традиционные вики содержат только текст, который компьютер не может ни понять, ни вычислить, Semantic MediaWiki добавляет семантические комментарии, что позволяет вики функционировать как

общей базе знаний. Semantic MediaWiki включает простой для использования язык запросов, который дает пользователям доступ к данным Вики.

Система Cyc[OpenCyc,2010] – является в настоящее время одной из старейших систем построения онтологий. Представляет собой программную систему, библиотека онтологий которой созд, которые человек использует в своей повседневной деятельности [Лапшин, 2010].

Еще одним представителем второй категории инструментариев можно назвать проект ЭЗОП [ЭЗОП, 2010]. Целью проекта является разработка системы Элементов Задач и ОПределений (ЭЗОП), которая должна представлять собой Web-сервер коллективного конструирования библиотек онтологий, разрабатываемый в стиле Web 2.0. В настоящее время доступна реализация проекта на платформе Drupal.

Не смотря на обилие проектов и технологий для сети Internet, в основе которых лежат семантические сети необходимо отметить ряд проблем, которые остались до сих пор не решенными:

- отсутствие единых стандартов для унификации представления знаний в сети Internet. Помимо проекта «Дублинское ядро» в сети Internet широко используются ряд других систем метаданных, таких как GILS, MARC, ONIX, LOM, UDDI [Башмаков и др., 2005].
- отсутствие унифицированных подходов к представлению мультимедиа данных для решения проблем глубинного web. Несмотря на то, что в данное время существуют подходы для поиска в сети Internet в документах, которые представлены в форматах таких PDF, DOC, RTF, для большинства мультимедийных форматов данная проблема до сих пор не решена.
- отсутствие эффективных реализаций систем управления базами знаний. Промышленность и клиенты нуждаются в системах управления знаниями. Производители реляционных СУБД и финансируемые ими группы ученых озабоченны в первую очередь сохранением многомиллионных инвестиций. Большинство разработчиков любой ценой стараются сохранить эволюционный путь развития СУБД. Поэтому, в исследованиях, посвященных «постреляционным моделям», нет ни слова об искусственном интеллекте как средстве моделирования бизнес процессов. Специалистам, работающим в области искусственного интеллекта, известно множество моделей представления знаний, обладающих не меньшей, а возможно, большей гибкостью и универсальностью, по сравнению с реляционной моделью данных [Шуклин, 2005].
- отсутствие единого подхода к проектированию интеллектуальных систем, в основе которых лежат семантические сети. Обилие подходов, которые существует в настоящее время, не дает возможность разрабатывать интеллектуальные системы, которые обладают легкой интегрируемостью, ориентированы на распределенную обработку знаний.

Для решения проблем указанных выше необходим пересмотр подходов к разработке интеллектуальных систем для сети Internet, которые применяются в настоящее время.

Переход от гипертекстов к семантически структурированным гипертекстам, в основе которых лежит SC-код

В качестве основы предлагаемого подхода будем использовать семантические сети с базовой теоретико-множественной интерпретацией. Основным способом кодирования информации для таких сетей является SC (Semantic Code)-код [Голенков и др, 2001]. Интеллектуальные системы, построенные с использованием SC-кода, будем называть sc-системами.

Сущности web-страницы будем представлять как элементы базы знаний (БЗ). Такой подход позволяет рассматривать пользовательский интерфейс web-приложения как специализированную интеллектуальную систему, решающую задачу организации диалога человека и интеллектуальной системы, и обеспечивающую решение основных задач предметной интеллектуальной системы. Основными классами задач такой специализированной интеллектуальной системы являются:

- просмотр текстов внешних языков, удобных и понятных для пользователя интеллектуальной системы;
- редактирование текстов внешних языков;

• трансляция текстов внешних языков в тексты SC-кода;
• трансляция текстов SC-кода в тексты внешних языков.

Для решения задач каждого класса используется отдельный класс интерфейсных компонент. Каждый компонент трактуется как специализированная интеллектуальная система, имеющая свою БЗ и машину обработки знаний (МОЗ). Пользовательский интерфейс в целом является результатом интеграции всех его интерфейсных компонент.

В основу предметной интеллектуальной системы положено представление знаний предметной области с помощью SC-кода. Машину обработки знаний такой интеллектуальной системы составляет множество согласованных предметно-независимых операций[OSTISa, 2010].

На основе приведенного выше анализа и анализа ряда других работ можно сформулируем шаги, которые позволят осуществить переход от традиционных web-ресурсов к семантическим web-ресурсам:

1) Переход от гипертекстов к семантически структурированным гипертекстам, в основе которых лежит SC-код.
2) Построение web-сайтов на основе семантически структурированных гипертекстов.
3) Разработка средств интеллектуальной навигации по семантически структурированному web-сайту.
4) Использование специализированного языка для более четкой семантической спецификации мультимедиа-документов, которые входят в состав web-ресурса.
5) Построение web-сайта как интеллектуальной справочной системы, в основе которой лежат семантические сети.
6) Общий поиск по семантическому пространству web-сайтов, организованных как интеллектуальная справочная система.

Семантически структурированные гипертексты – это гипертексты, информация в которых будет отображаться помощью SCn-кода (способа псевдо-естественного кодирования семантических сетей, представленных в SC-коде, Semantic Code natural). Разметка таких гипертекстов производится с помощью SCnML (SCn Markup Language) – языка разметки текстов SCn-кода[OSTISb, 2010].

В семантической технологии проектирования баз знаний sc-систем SCn-код используется, в качестве одного из основных способов представления знаний. Использование для записи текстов базы знаний языка близкого к естественному существенно повышает качество восприятия текстов баз знаний как инженерами по знаниям, так и пользователями баз знаний, поэтому SCn-код – может использоваться в качестве внешнего, понятного пользователю, языка sc-системы.

SCn-код задается множеством всех sc.n-статей, каждая из которых описывает семантическую окрестность некоторого понятия предметной области. Каждая статья в свою очередь состоит из из идентификатора sc-элемента, описываемого в этой sc.n-статье, и, возможно, одного или нескольких последующих sc.n-полей. При описании sc-элемента в sc.n-статье sc.n-поля описывают как, какими ролями и связками каких отношений, связан описываемый sc-элемент с другими sc-элементами. Ряд sc.n-полей может содержать мультимедиа или тексты логических утверждений. Мультимедиа может включать любые информационные конструкции, обозначаемые как внешние по отношению к SCn, в том числе и sc.n-тексты.

Каждому sc.n-полю в языке SCnML соответствует отдельный тег, такой подход позволяет однозначно определить однозначное соответствие между отображаемым sc.n-полем и тегом SCnML, с помощью которого это поле кодируется.

Построение web-сайтов на основе семантически структурированных гипертекстов

Семантический размеченный гипертекст, в основе которого лежит SCn, позволяет перейти от формализованной модели гипертекста, в которой ссылки между web-страницами не имели явно заданных отношений, к условно-типовой модели гипертекста, в которой переходы между web-страницами производятся по ссылке с явно выделенным типом отношения. Таким образом, при

использовании такого подхода можно говорить о навигации по гипермедийной семантической сети.

В качестве платформы для реализации web-сайтов на основе семантически структурированных гипертекстов используется платформа MediaWiki. В основе подхода, используемого в проекте MediaWiki, лежит понятие статьи. Статья MediaWiki – это семантическая окрестность некоторого понятия, записанная на естественном языке. Использование SCn-кода при разметке статей MediaWiki, позволяет описать семантическую окрестность данного понятия формально. Таким образом, если переход по ссылке в обычной Mediawiki-системе дает возможность ответить на вопрос "что это такое X", то переход по ссылке в Mediawiki-системе, использующей семантически структурированный гипертекст, позволяет указать контекст данного перехода – "что такое X, связанное с данным понятием отношением Y".

Средства интеллектуальной навигации по семантически структурированному web-сайту.

Использование семантически размеченного гипертекста позволяет не только говорить о навигации и переходах между элементами семантической статьи, но и навигации с помощью специального языка запросов. Такой язык запросов позволяет искать на основе шаблона поиска, который задается в виде графа изоморфного искомому фрагменту семантической сети. Подобные языки запросов используются для поиска на web-сайтах, разработанных с использованием платформы Semantic MediaWiki [Бениаминов, 2008]. Однако, в отличие от предлагаемого подхода, в качестве области поиска в Semantic MediaWiki используется не семантическая сеть, а семантически аннотированный гипертекст.

Средства Mediawiki при использовании SCn-кода позволяют осуществлять поиск не только по понятиям, которым соответствует некоторая sc.n-статья, но и по отношениям, связкам и ролям, которые являются частью описания семантической окрестности искомого понятия. Ввиду того, что в основе sc.n-статьи лежит унифицированный способ представления гипертекста и более широкий набор типовых отношений, языки запросов для таких систем позволяют более точно и полно сформулировать запрос к системе.

Расширение возможностей MediaWiki в предлагаемом подходе предполагается осуществлять за счет четкой спецификации страниц, соответствующих метаописаниям представления SCnML-тегов для платформы Mediawiki. Использование таких унифицированных элементов страницы совместно с простейшим языком запросов платформы Semantic Mediawiki позволяет формировать метаописания страницы семантически структурированного web-сайта.

Специализированный язык для более четкой семантической спецификации мультимедиа-документов, которые входят в состав web-ресурса.

Для более четкой спецификации мультимедийных документов, которые используются в рамках семантического web-сайта необходим специальный язык. В качестве такого языка предлагается использовать язык гипермедийных структур (ЯГС) [Колб, 2009], который позволяет обеспечить семантически точную спецификацию фрагмента web-ресурса любой синтаксической структуры за счет введения специальных классов отношений на множестве фрагментов web-ресурса. В основе ЯГС лежит представление знаний с помощью SC-кода. ЯГС позволяет описывать сущности пользовательского интерфейса на двух уровнях представления информации синтаксическом и семантическом.

Основным понятием, с которым работает ЯГС является понятие sc-файла (sc-ссылки) – sc-элемента, содержимым которого является инородный по отношению к SC-коду файл произвольной структуры. ЯГС предусматривает следующие базовые классы мультимедийных документов, которые можно использовать в качестве содержимого sc-файла:

- Изображение
- Абстрактный текст
  - Линейная конфигурация графем
    - Строка символов
      - Символьное представление числа

•• Нелинейная конфигурация графем
••• sc.g-конструкция
• Аудио-информация
• Динамическая графика
•• Видеоинформация
•• Анимация

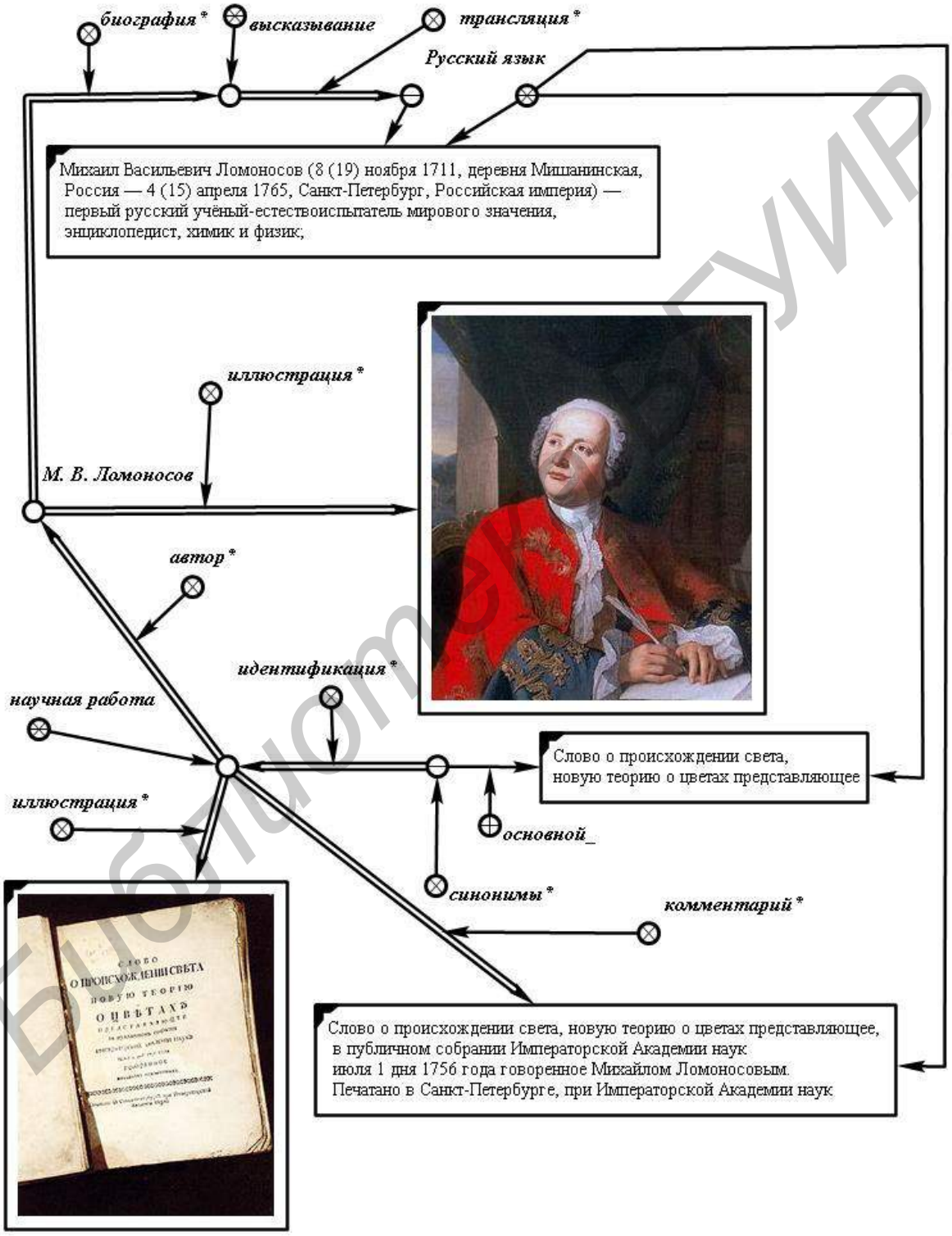


Рисунок 1– Фрагмент базы знаний sc-систем

Для каждого из данных типов содержимого sc-файла вводится понятия отношения, которое указывает роль, которую играет содержимое данного sc-файла, в рамках конкретной предметной области. Приведем пример, допустим, мы создали в базе знаний sc-файл, содержимым которого является изображение М.В Ломоносова (Рисунок 1). Для предметной области физическая химии будет введено отношение ***иллюстрация****, которое будет связывать sc-узел, обозначающий М.В. Ломоносова с sc-узлом обозначающим изображение М.В. Ломоносова. Введем отношение ***автор****, которое будет связывать sc-элемент, обозначающий автора М.В. Ломоносова, с одной из его научных работ. Используя отношение ***иллюстрация****, мы свяжем sc-элемент, обозначающий научную работу, с изображением, на котором отображена первая страница данной научной работы. Отношение ***комментарий**** на рисунке 1 связывает обозначение конкретной научной работы М.В. Ломоносова с sc-файлом, содержащим комментарий на естественном языке к данной работе.

Пример спецификации мультимедийной информации на рисунке 1 демонстрирует, как можно описать мультимедийные фрагменты, используя понятийный аппарат предметной области. Однако такой способ спецификации мультимедийной информации не всегда удобен для машинной обработки. Для более четкой спецификации мультимедиа информации в ЯГС используется понятие формата. Формат – это однозначное описание структур данных, записанных в sc-файле. Будем назвать sc-файлами одного типа sc-файлы, содержимое которых соответствует одному формату. Понятие формата позволяет уточнить синтаксическую структуру содержимого sc-файла и определяет способ хранения информационной конструкции (например: растровый, векторный), форму хранения информационной конструкции (например: используемый алгоритм сжатия) и способ кодирования информационной конструкции (например: SC-код, бинарный код, XML). Пример спецификации мультимедиа содержимого sc-файла приведен на рисунке 2.

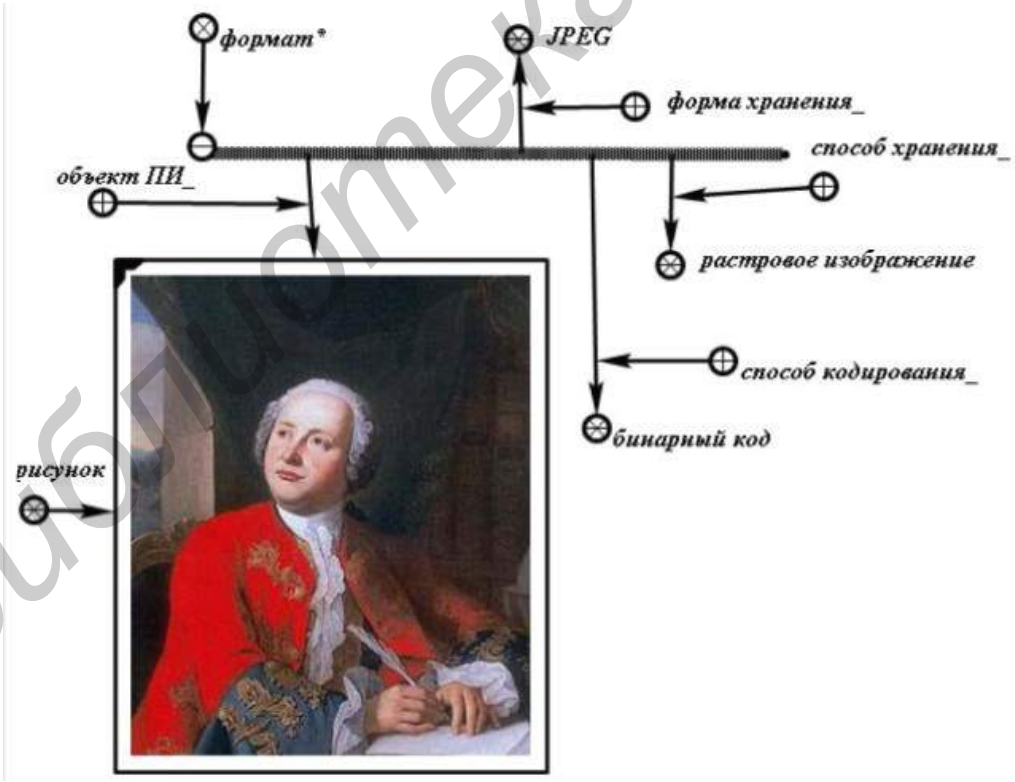


Рисунок 2– Спецификация мультимедиа информации, уточняющая ее синтаксическую структуру.

Заметим, что sc-файл, обозначающий некоторую информационную конструкцию, вовсе не обязан явно “хранить” эту информационную конструкцию в качестве своего содержимого. Эта информационная конструкция может быть неизвестна (не сформирована) и разбита на

фрагменты, каждый из которых представлен явно, и, следовательно, нет никакой необходимости явно представлять и хранить всю исходную информационную конструкцию.

Часть информации, которая описывается с помощью ЯГС, может быть не востребована пользователем. Как правило, при описании базы знаний sc-системы пользователя интересует лишь роль, выполняемая информационной конструкцией, хранящейся в содержимом sc-ссылки, в рамках конкретной предметной области. Однако, в некоторых случаях, необходимо получить более подробную информацию о какой-либо информационной конструкции, чем роли, которые задаются отношениями над узлами с содержимыми в рамках предметной области. С этой целью при формировании БЗ в неё добавляется дополнительная информация.

Одним из достоинств SC-кода является возможность представления инородных информационных конструкций с разной степенью подробности. На Рисунке 3 представлен один из возможных способов кодирования растрового изображения с помощью средств SC-кода. Такой фрагмент sc-конструкции передает лишь закономерности расположения точек растрового изображения. С целью получения полной информации об изображении в базу знаний можно добавить sc-конструкцию описывающую цвет пиксела по шаблону, приведенному на рисунке 4.

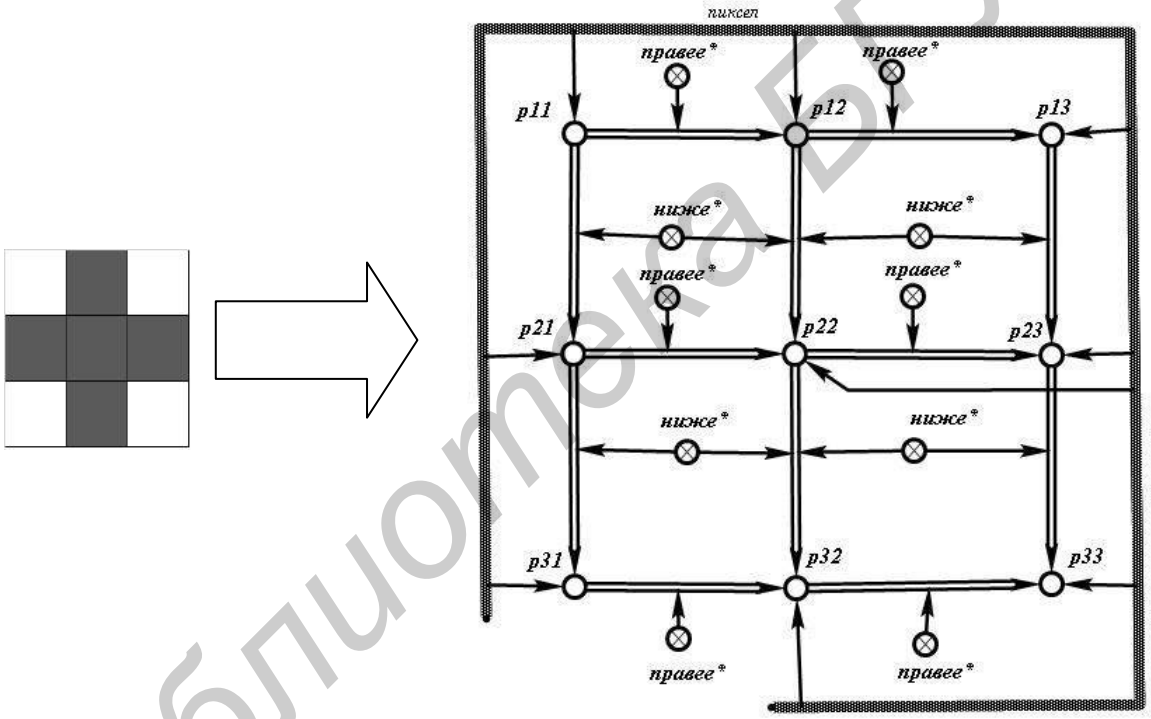


Рисунок 3– Один из возможных способов представления растрового изображения с помощью средств SC-кода

Согласно рисунку 4 для каждого пиксела белого цвета необходимо построить sc-конструкцию, описывающую цвет пиксела, путем замены элемента *Pmk*, а для каждого пиксела черного цвета необходимо построить sc-конструкцию, описывающую цвет пиксела, путем замены элемента *Pij*.

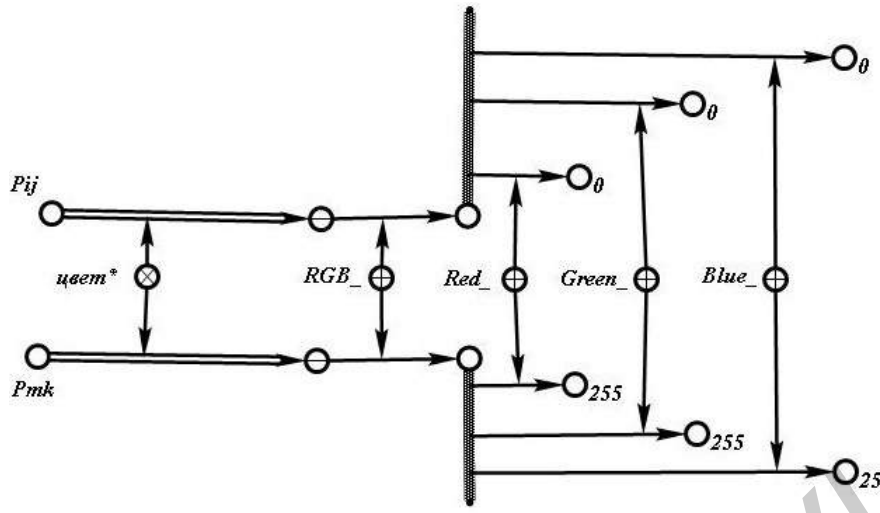


Рисунок 4 – Один из возможных способов представления растрового изображения с помощью средств SC-кода

Необходимо заметить, что подобное кодирования информационных конструкций средствами SC-кода необходимо только в тех случаях, когда БЗ интеллектуальной системы разрабатывается для интеллектуального анализа самой инородной информационной конструкции. К задачам, при решении которых эта возможность SC-кода может понадобиться, можно отнести задачи распознавания образов, задачи распознавания голоса и ряд других задач интеллектуального анализа данных.

Однако при решении целого ряда задач нет необходимости детализировать инородные информационные конструкции до машинного представления. В таких случаях можно ограничиться лишь необходимым уровнем декомпозиции информационной конструкции.

Зачастую, при отображении разнообразных информационных конструкций, возникает задача обратная задаче декомпозиции. В таких случаях sc-система осуществляет поиск sc-конструкции, на основании которой можно синтезировать запрашиваемые данные.

Построение web-сайта как интеллектуальной справочной системы, в основе которой лежат семантические сети.

Неотъемлемой частью любой мультимедийной системы является компонент трансляции действий пользователя с некоторого внешнего языка на язык понятный системе. Важность таких компонентов возрастает, когда мы говорим об интеллектуальных системах, так как такая система не только должна реагировать на команды, которые отправляет ей пользователь, но и “понимать” эти команды.

Рассмотрим устройство транслятора sc-системы. Основными операциями транслятора sc-системы является:

- трансляция информационных конструкций, представленных с помощью некоторого внешнего языка, понятного пользователю sc-системы, в SC-код;
- трансляция информационных конструкций, представленных с помощью SC-кода в некоторый внешний язык понятный пользователю sc-системы.

Рассмотрим устройство транслятора текстов SCn-кода в тексты SC-кода, как типовой пример устройства компонента транслятора sc-системы.

В зависимости от предметной области компонент sc.n-транслятора будет задавать целую иерархию трансляторов. На рисунке 5 приведен пример далеко не полной иерархии трансляторов, которые могут входить в состав sc.n-транслятора.

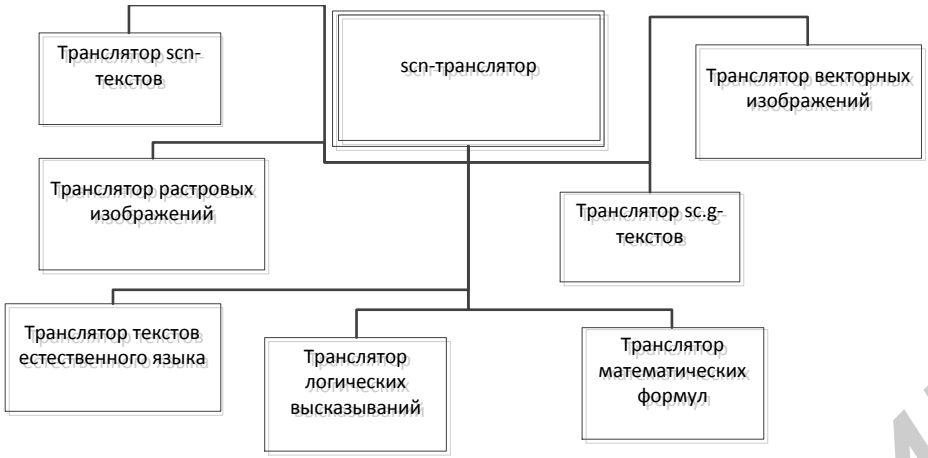


Рисунок 5 – Трансляторы, входящие в состав scn-транслятора

Как уже говорилось ранее, SC-код позволяет представлять знания с различным уровнем детализации, поэтому в зависимости от предметной области, поэтому к схеме, показанной на рисунке 5 можно как добавить новые трансляторы, так и убрать их.

Заметим, что при реализации sc.n-транслятора не всегда целесообразно преобразовывать напрямую из SCn в SC-код, для удобной отладки возможна трансляция и в формы, которые более близки к представлению знаний с помощью SC-кода, а именно SCg-код [OSTISc, 2010] – графическое двухмерное представление SC-кода и SCs-код [Голенков и др, 2001] – линейное представление SC-кода. При этом возможны следующие цепочки преобразования текстов SCn:
1.scn-тексты –> тексты SC-кода –> синтаксическое описание sc.g-конструкций –> sc.g-тексты;
2.scn-тексты –> тексты SC-кода –> синтаксическое описание sc.s-конструкций –> sc.s-тексты;
3.scn-тексты –> синтаксическое описание sc.g-конструкций –> sc.g-тексты;
4.scn-тексты –> синтаксическое описание sc.s-конструкций –> sc.s-тексты;
5.scn-тексты –> синтаксическое описание sc.g-конструкций –> sc.g-тексты –> тексты SC-кода;
6.scn-тексты –> синтаксическое описание sc.s-конструкций –> sc.s-тексты –> тексты SC-кода.

Наиболее универсальными из приведенных цепочек являются первая и вторая цепочки так как позволяют транслировать в общий для всех sc-систем способ кодирования – SC-код. Такой подход позволяет строить трансляторы из любого внешнего языка sc-системы, для которого построен транслятор его текстов в тексты SC-кода, в любой внешний язык, для которого построен транслятор из SC-кода в тексты этого внешнего языка. То есть если мы имеем транслятор scn-текстов в тексты SC-кода и транслятор из текстов SC-кода в тексты SCg-кода, то у нас есть всё необходимое для того чтобы построить транслятор из scn-текстов в тексты SCg-кода.

Любой внешний по отношению к sc-системе язык можно транслировать с разной степенью полноты, в зависимости от потребностей, которые испытывает пользователь sc- системы. Пример неполной трансляции продемонстрирован на рисунке 4. Таким образом, для каждого внешнего по отношению к sc-системе языка можно построить семейство трансляторов транслирующих с различной степенью семантической полноты. Ниже приведены схемы трансляции, которые возможны при реализации трансляторов для sc-систем:
1. scn-тексты –>семантически эквивалентные тексты SC-кода с полнотой 1;
scn-тексты –>семантически эквивалентные тексты SC-кода с полнотой 2;
…
scn-тексты –>полностью семантически эквивалентные тексты SC-кода.
2. scn-тексты –>семантически эквивалентные тексты SC-кода с полнотой –> семантически эквивалентные тексты SC-кода с полнотой 2 –> … семантически

эквивалентные тексты SC-кода с полнотой n –> ... –> полностью семантически эквивалентные тексты SC-кода.

Обе приведенные схемы возможны при реализации трансляторов sc-систем. Согласно первой схеме для указанной sc-системы реализуется один из семейства возможных трансляторов транслирующий тексты некоторого внешнего языка с заданной степенью полноты. Согласно второй схеме необходимо реализовывать семейство трансляторов, каждый из которых, используя результаты предыдущего транслятора, достраивает фрагмент базы знаний до некоторого уровня полноты. Если первая схема удобна для sc-систем, в которых необходимо производить трансляции только с заданным уровнем полноты, то вторая схема будет полезна для sc-систем, которые работают с разными уровнями детализации информации.

В качестве основной схемы для sc.n-транслятора мы выберем первую схему с построением полностью семантически эквивалентных текстов sc-кода.

В зависимости задач, которые ставятся перед пользовательским интерфейсом sc-системы можно выделить еще два класса трансляторов транслятор статических информационных конструкций и транслятор интерактивных информационных конструкций.

Первый класс трансляторов используется для распределенных sc-систем, в которых тяжело обеспечить быструю трансляцию текстов внешних языков из-за распределенного характера системы. Основными характеристиками для пользовательских интерфейсов таких sc-систем является наличие языка семантической разметки текстов внешнего языка недоступного или неизвестного пользователю такой sc-системы и языка для общения с пользователем, который транслируется в семантический язык разметки. Такие трансляторы характерны для web-sc-систем. Для таких трансляторов не важна скорость трансляции текстов, а важна семантическая мощность языка разметки.

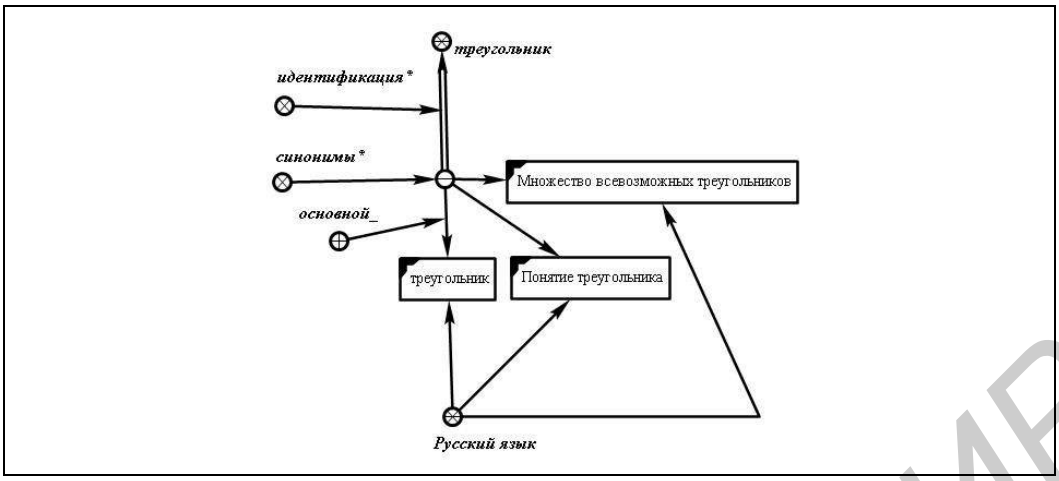
Второй класс трансляторов может быть использован для sc-систем, которые работают локально. Такие трансляторы осуществляют трансляцию текстов внешнего языка напрямую в SC-код.

Исходя из приведенных выкладок приведем общие этапы разработки трансляторов для sc-систем:
1. Определение конструкций внешнего языка;
2. Выявление однородных с точки зрения синтаксического представления информационных конструкций (подъязыков) в рамках внешнего языка;
3. Выбор схемы трансляции для транслятора каждого подъязыка;
4. Разработка правил трансляции для каждого из подъязыков;
5. Отладка и тестирование транслятора.

Приведем пример правила трансляции текстов SCn-кода в тексты SC-кода.

Таблица 1 – Правило трансляции текстов SCn-кода в тексты SC-кода.

Текст SCn-кода
треугольник = *Понятие треугольника* = *Множество всевозможных треугольников*
Разметка SCn-кода
{{SCnFieldConcept|треугольник}} *{{SCnFieldSpecConSyn|1|Класс треугольников}}* *{{SCnFieldSpecConSyn|1|Понятие треугольника}}* *{{SCnFieldSpecConSyn|1|Множество всевозможных треугольников}}*
Семантически эквивалентная SCn-разметке конструкция в SC-коде



Рассмотрим еще один компонент sc-системы, который тесно связан с компонентом трансляции, речь идет о компоненте просмотрщик. Основными операциями просмотрщика в рамках sc-системы являются:

- операции отображение текстов внешних по отношению к sc- системе языков;
- операции формирования команд поиска информационных конструкций, на основе которых можно синтезировать информационные конструкции, которые может отобразить просмотрщик. Такой класс операций необходим для того, чтобы просмотрщик мог отображать элементы алфавита внешнего языка, которые хранятся в базе знаний sc-системы, но декомпозированы на информационные конструкции, находящиеся на более низком семантическом уровне, чем те, которые может отображать просмотрщик.

Выделим два класса просмотрщиков – просмотрщики, которые отображают тексты внешних языков, содержащих синтаксически однородные информационные конструкции и просмотрщики, которые отображают тексты внешних языков, содержащих синтаксически разнородные информационные конструкции. К первому классу можно отнести просмотрщики изображений в формате BMP, просмотрщики изображений в формате JPG и другие подобные. Второй класс просмотрщиков наиболее востребован и более широко распространен, к этому классу можно отнести просмотрщики web-страниц, просмотрщики документов PDF и многие другие. В данном контексте тип просмотрщика определяется множеством всевозможных мультимедиа-конструкций, которые могут входить в состав внешнего языка sc-системы.

Как компонент транслятор, компонент просмотрщика декомпозируется на просмотрщики информационных конструкций более низкого семантического уровня. При этом во множество просмотрщиков, полученное при декомпозиции, могут входить просмотрщики обоих рассмотренных классов.

Полноценный диалог пользователя и системы не возможен без редактора языка, с помощью которого общается пользователь и система. К редакторам внешних языков интеллектуальных систем и в частности sc-систем для обеспечения эффективного диалога предъявляются более высокие требования, чем редакторам внешних языков традиционных систем. Рассмотрим устройство редакторов внешних языков интеллектуальных систем на примере редакторов внешних языков sc-систем.

Редактор sc-системы представляет собой многокомпонентную sc-систему, в состав которой могут входить разнообразные просмотрщики, трансляторы и другие редакторы. В зависимости от внешнего языка можно выделить два класса редакторов – редакторы внешних языков, тексты которых синтаксически однородны, и редакторы внешних языков, тексты которых имеют синтаксически разнородную структуру. Первый класс редакторов декомпозируется на компоненты просмотрщиков и трансляторов, а второй класс декомпозируется на редакторы подъязыков внешнего языка sc-системы.

Кроме указанных классов выделяются еще два класса редакторов sc-систем. Класс редакторов, редактирования в которых осуществляется по принципу WYSIWYG (What You See Is What You

Get, «что видишь, то и получишь»). В таких редакторах измененные тексты внешнего языка сразу поступают компоненту просмотрщик для дальнейшего отображения. Второй класс редакторов меняет тексты некоторого языка семантической разметки, который затем транслируется для отображения пользователю. К первому классу редакторов можно отнести редакторы текстов SCg-кода, SCs-кода, ко второму относятся редакторы web-страниц.

К основным операциям редактора sc-системы относят:
- операции редактирования информационных конструкций внешнего языка;
- операции формирования команд для трансляции текстов внешнего языка;
- операции формирования команд для просмотра текстов внешнего языка;
- операции формирования команд для редакторов, входящих в состав данного.

При работе с пользовательским интерфейсом любой системы пользователь обязан поставить перед собой некоторую цель, которой он собирается достигнуть. И затем сформулировать пользовательскому интерфейсу системы задачи, которые он обязан решить для того, чтобы достигнуть цели пользователя. Для того чтобы уточнить понятие цели и задачи будем считать, что цель дает ответ на вопрос «Чего нужно достигнуть пользователю?», а задача – на вопрос «Какими действиями этого может достигнуть пользователь, используя пользовательский интерфейс системы?».

Назовем интерфейсной целью изменение текущего состояния пользовательского интерфейса sc-системы в сторону улучшения данного состояния. Назовем интерфейсной задачей упорядоченное множество интерфейсных действий, которые необходимо совершить для решения данной интерфейсной задачи. Тогда формально интерфейсную цель определим как множество интерфейсных задач. Такое множество будет упорядоченным в том случае, если результаты решения каждой последующей интерфейсной задачи зависят от решения предыдущей, и не упорядоченным в том случае, если решения интерфейсных задач не зависят друг от друга. Очевидно, что интерфейсная цель будет является конечным множеством интерфейсных задач в том и только в том случае, если она достижима на множестве интерфейсных задач, которые можно решить с помощью пользовательского интерфейса sc-системы. Будем считать, что интерфейсная цель не является мультимножеством, то есть для того, чтобы достигнуть некоторой цели, которая подразумевает решение одной интерфейсной задачи n раз, необходимо сформулировать n интерфейсных целей.

В общем случае для достижения каждой конкретной интерфейсной цели можно решить некоторое конечное множество интерфейсных задач. Каждое такое множество задач назовем способом достижения интерфейсной цели sc-системы. Для конкретной интерфейсной цели такие способы могут:
- быть уже сформированы и хранится в базе знаний пользовательского интерфейса sc-системы;
- формироваться пользователем в процессе работы с пользовательским интерфейсом sc-системы;
- быть синтезированы на основе знаний о пользователе из предметной sc-системы.

Для решения некоторой интерфейсной задачи необходимо совершить конечное множество интерфейсных действий. Каждое такое интерфейсное действие должно быть инициировано специальной интерфейсной командой. Интерфейсная команда – это связка, в состав которой входят аргументы интерфейсной команды однозначно специфицирующие некоторое интерфейсное действие sc-системы. Формирование интерфейсной команды в базе знаний пользовательского интерфейса sc-системы (инициирование интерфейсного действия) приводит к выполнению интерфейсного действия, которое специфицируется данной интерфейсной командой.

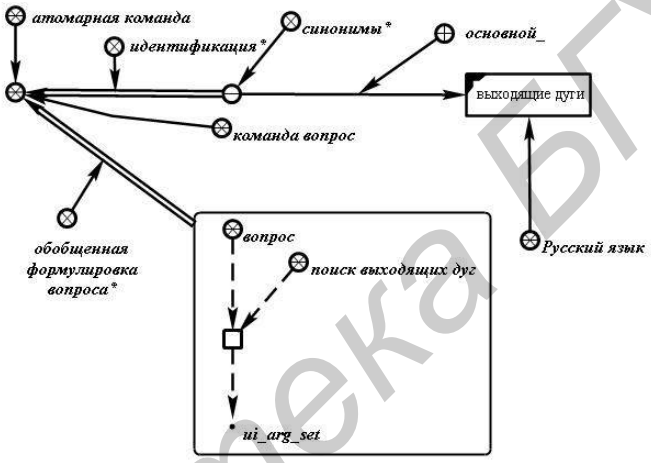
Приведем классификацию команд пользовательского интерфейса:
- по признаку атомарности выделяют следующие классы интерфейсных команд:
  - атомарные классы интерфейсных команд – элементами атомарного класса интерфейсных команд являются знаки однотипных интерфейсных команд, формируемых в разное время. Каждому классу интерфейсных команд сопоставлен некоторый ключевой элемент,

являющийся sc-элементом и обозначающий этот класс. Элементами класса интерфейсных команд являются обозначения команд (знаки команд);

- не атомарные классы интерфейсных команд. Элементами неатомарного класса интерфейсных команд являются классы однотипных классов атомарных интерфейсных команд и однотипных классов неатомарных интерфейсных команд;

- по признаку автора формирования интерфейсной команды:
  - интерфейсные команды, сформированные пользователем sc-системы;
  - интерфейсные команды, сформированные sc-системой;
- по признаку адресата интерфейсной команды:
  - интерфейсные команды, инициирующие интерфейсные действия формирования запросов к предметной sc-системе (см. пример на рис.6 а));
  - интерфейсные команды, инициирующие интерфейсные действия редакторов и просмотрщиков (см. пример на рис.6 б)).

а). Пример оформления класса интерфейсных команд вопросов к предметной sc-системе



б). Пример оформления класса интерфейсных команд

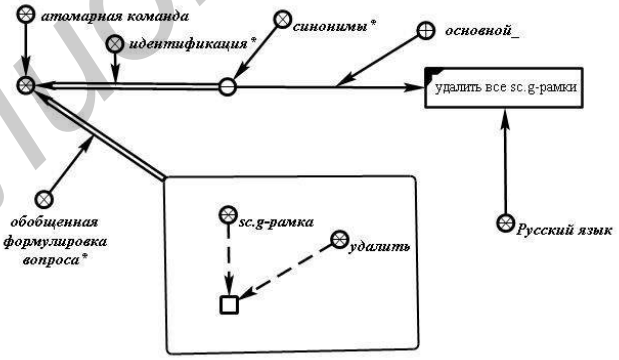


Рисунок 6 – Пример оформления класса команд пользовательского интерфейса sc-системы

В процессе взаимодействия пользователя с пользовательским интерфейсом sc-системы конкретная интерфейсная задача может быть решена с помощью некоторого конечного количества способов решения данной интерфейсной задачи. Такие способы решения могут отличаться количеством используемых интерфейсных команд или порядком их следования. Назовем протоколом решения конкретной интерфейсной задачи множество интерфейсных команд, формирование которых в базе знаний пользовательского интерфейса sc-системы позволяет решить указанную интерфейсную задачу. Множество интерфейсных команд в

протоколе решения интерфейсной задачи будет упорядоченным в том случае, когда результат выполнения каждого предыдущего интерфейсного действия является необходимым аргументом интерфейсной команды инициирующей следующее интерфейсное действие. Множество интерфейсных команд в протоколе решения интерфейсной задачи будет неупорядоченным в том случае, когда интерфейсные действия необходимые для решения интерфейсной задачи можно инициировались в произвольном порядке. На основании протокола решения конкретной интерфейсно задачи очевидно однозначно можно определить способ решения класса интерфейсных задач, к которому принадлежит данная интерфейсная задача. Для класса интерфейсных задач такие способы её решения могут:

- быть уже сформированы и хранится в базе знаний пользовательского интерфейса sc-системы;
- формироваться пользователем в процессе работы с пользовательским интерфейсом sc-системы;
- быть синтезированы на основе знаний пользовательского интерфейса sc-системе о способе решения других задач.

В процессе взаимодействия с пользователем может возникнуть ситуация когда в базе знаний пользовательского интерфейса sc-системы хранится несколько различных способов решения некоторой интерфейсной задачи. В таком случае, если пользователь специально не указывает какой из способов решения интерфейсной задачи необходимо использовать, система выбирает наиболее эффективный из существующих, например, с наименьшим количеством используемых интерфейсных команд.

На основании сказанного ранее дадим определение понятию протокола достижения конкретной интерфейсной цели. Протоколом достижения некоторой конкретной интерфейсной цели sc-системы будем называть множество протоколов решения конкретных интерфейсных задач, которые были решены для достижения данной конкретной интерфейсной цели.

Общий поиск по семантическому пространству web-сайтов, организованных как интеллектуальная справочная система.

Для разработки sc-систем, ориентированных на использование в сети Internet, необходима разработка специализированной системы, которая бы обеспечивала удаленный доступ к базе знаний sc-системы. Основными задачами такой системы будут являться:

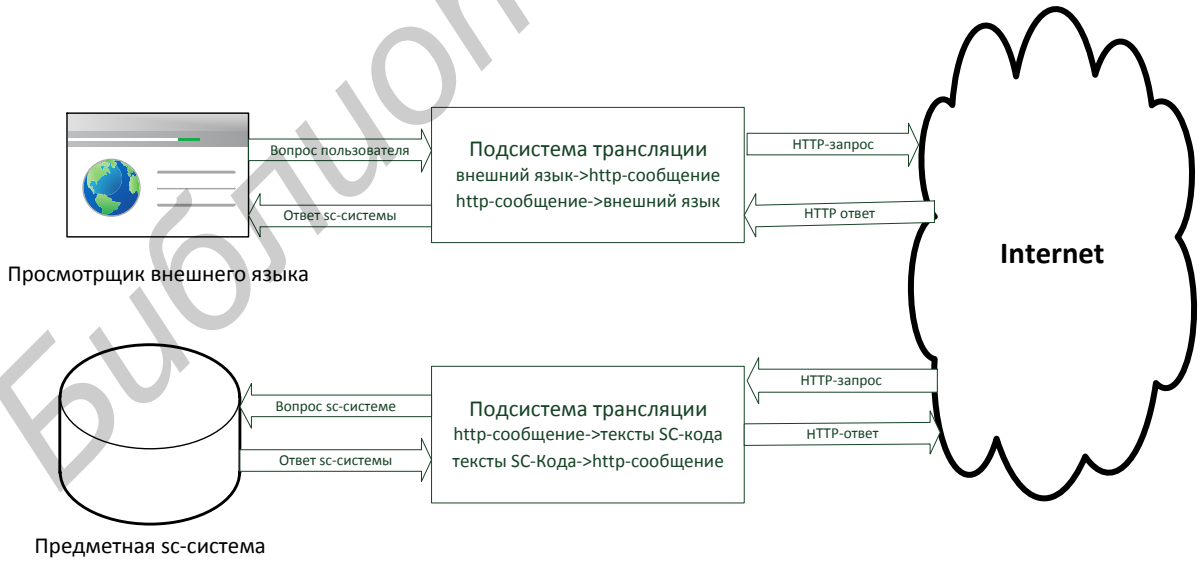


Рисунок 7 – Крупноблочная схема работы сервера баз знаний для sc-систем

- задача обеспечения удаленного выполнения запросов к sc-системе;
- задача обеспечения удаленного выдачи ответов на запросы пользователя sc-системы;
- задача обеспечения многопользовательской работы с базой знаний;
- задача обеспечения трансляции действий пользователя в базу знаний sc-системы по запросу.

Назовем систему, решающую подобные задачи сервером баз знаний для sc-систем. Крупноблочная схема работы сервера баз знаний sc-систем представлена на рисунке 7.

Реализация схемы, приведенной на рисунке, предполагается на базе одной из известных реализаций сервисно-ориентированных архитектур – архитектура REST [Roy Fielding, 2000] или на архитектура на основе RPC [W3C, 2010]. Такая организация взаимодействия пользовательского интерфейса sc-системы и предметной sc-системы с позволяет говорить о возможности работы с семантическими web-сервисами [W3C, 2010], реализованными на базе SC-кода.

С точки зрения практики наиболее интересна такая реализация сервера баз знаний sc-систем, которая позволяет при наличии множество серверов обеспечить их взаимодействие друг с другом, таким образом, образуя семантической пространство для взаимодействия с пользователями sc-систем. В этом случае сервер баз знаний sc-систем можно рассматривать как специализированный компонент распределенной sc-системы. А взаимодействие серверов как взаимодействие однотипных компонент распределенной sc-системы.

Специализированный компонент распределенной sc-системы для сети Internet должен удовлетворять следующим требованиям:
- иметь функцию самостоятельной работы в качестве отдельного сервера баз знаний sc- систем;
- иметь функции обработки сообщений от других компонентов распределенной sc-системы;
- иметь функции рассылки сообщений для других компонентов распределенной sc-системы.

Требования которые выдвинуты к компоненту распределенной sc-системы определяются характером сети Internet, в которой данный компонент будет функционировать.

В настоящее время существует большое количество подходов к организации взаимодействия в распределенных интеллектуальных системах [Тарасов, 2002], однако, когда дело касается практической реализации распределенной системы для сети Internet, возникают значительные технические трудности. Трудности связаны с тем, что с ростом количества взаимодействующих узлов распределённой системы существенно возрастают вычислительные мощности и размер дискового пространства, необходимые для обработки взаимодействия между узлами системы. Эта проблема не позволила до настоящего времени создать распределенную систему для глобальных сетей уровня сети Internet.

Одним из интересных и перспективных решений данной проблемы являет проект Netsukuku [Netsukuku, 2010]. Проект разрабатывается для социальных сетей, работающих с использованием технологии Wi-Fi. Структуру сети авторы предлагают представлять как фрактал – математическую структуру с дробной размерностью, которая обладает свойством рекурсивности: каждая её часть является уменьшенной копией целого. Поэтому возможно большое сжатие структуры, которая может безгранично расширяться. Структуру маршрутной карты такой сети можно определить как высококластеризованный граф узлов. Для расчёта всех необходимых путей связи узла со всеми остальными узлами протокол использует особый алгоритм, который работает по законам распространения волны в физике.

Учитывая подход, предложенный в проекте Netsukuku, для организации взаимодействия между компонетами в распределенной sc-системе, приведем основные классы операций компонента распределенной sc-системы:
- операции обработки сообщений от других компонент распределенной sc-системы;
- операции ретрансляции сообщений от других компонентов распределенной sc-системы;
- операции синтеза фрагмента маршрутной карты распределенной sc-системы;
- операции формирования волны сообщений другим компонентам распределенной sc-системы;
- операция выполнения запросов к sc-системе;
- операция выдачи ответов на запросы пользователя sc-системы;
- операция трансляции действий пользователя в базу знаний sc-системы по запросу.

Используя такой подход для организации сети серверов баз знаний sc-систем, мы получаем новое качество инструментария для реализации идей Semantic Web.

Заключение

В статье предложен подход к разработке интеллектуальных систем в основе, которых лежат семантические сети, для сети Internet. Выделено два класса систем: системы на базе семантически структурированного гипертекста и системы, в основе которых лежат гипермедийные семантические сети. К основным достоинствам предлагаемого подхода можно отнести:

- ориентация подхода на коллективную разработку баз знаний для интеллектуальных систем за счет использования платформы Mediawiki и семантически структурированного гипертекста;
- использование для представления знаний способа псевдоестественного кодирования знаний;
- простой и прозрачный переход от семантически структурированного гипертекста к семантической сети;
- унификация способов организации диалога пользователя с интеллектуальной системой за счет разработки языковых средств ведения диалога и, как следствие этого, снижение начальной требований предъявляемых к подготовке конечных пользователей;
- возможность интеграции базы знаний и машины обработки знаний пользовательского интерфейса интеллектуальных систем с другими базами знаний и машинами обработки знаний, построенными с использованием SC-кода;
- возможность поиска и анализа необходимой информации в базе знаний пользовательского интерфейса, в том числе с учетом поиска в различных мультимедийных фрагментах, за счет семантической спецификации мультимедиа информации на различных уровнях представления

Работа выполнена при поддержке гранта БРФФИ Ф10М-085 и гранта БРФФИ-РФФИ Ф10Р-175.

Библиографический список

[Башмаков и др., 2005] Башмаков, И.А. Интеллектуальные информационные технологии: учебное пособие/ И.А. Башмаков, А.И. Башмаков// Москва: МГТУ им. Н.Э. Баумана, 2005 г., 304 стр.

[Бениаминов, 2008] Бениаминов, Е.М. О построении Web-сервера в стиле Semantic Wiki с открытым контекстным языком представления и запросов/Е. М. Бениаминов// КИИ-2008. Труды конференции. Т 2, с. 15-21

[Веб-фрагмент, 2010] Спецификация формата веб-фрагмента – версия 0.9 [Электронный ресурс]. – 2010. - Режим доступа: http://msdn.microsoft.com/ru-ru/library/cc304073%28VS.85%29.aspx . – Дата доступа: 27.11.2010

[Вики о микроформатах, 2010] Вики о микроформатах [Электронный ресурс]. – 2010. - Режим доступа: http://microformats.org/wiki/Main_Page-ru . – Дата доступа: 27.11.2010

[Гаврилова, 2000] Гаврилова, Т.А. Базы знаний интеллектуальных систем/ Т.А. Гаврилова, В.Ф. Хорошевский //СПб – Питер, 2000 г., 384 стр.

[Гаврилова, 2008] Гаврилова, Т.А. Визуальные методы работы со знаниями: попытка обзора/Т. А. Гаврилова, Н.А. Гулякина // Искусственный интеллект и принятие решений, 2008, № 1, с. 15-21

[Голенков и др, 2001] Представление и обработка знаний в графодинамических ассоциативных машинах /В. В. Голенков, [и др]; – Мн. : БГУИР, 2001.

[Касьянов, 2003] Касьянов, В.Н. Графы в программировании: обработка, визуализация и применение/ В. Н. Касьянов, В. А. Евстигнеев // BHV–Санкт-Петербург, 2003.–1104 с.

[Колб, 2009] Колб, Д.Г. Средства просмотра баз знаний интеллектуальных систем / Д. Г. Колб // Вестник БрГТУ. - 2009. - № 5. - С.58-62.

[Ландэ, 2009] Ландэ, Д. В. Глубинный веб информационная среда для бизнес аналитика /Д. В. Ландэ // Информационные технологии для менеджмента, 2009, № 9, с. 28-32

[Лапшин, 2010] Лапшин, В.А. Система Сус и её библиотека онтологий /В. А. Лапшин // Искусственный интеллект и принятие решений, 2010, № 2, с.42-53

[Левшин, 2009] Левшин, Д. Базы данных в семантической паутине [Электронный ресурс]. – 2009. - Режим доступа: http://www.osp.ru/os/2009/07/10464695/. – Дата доступа: 27.11.2010

[Тарасов, 2002] Тарасов, В. Б**.** От многоагентных систем к интеллектуальным организациям: философия, психология, информатика/ В. Б. Тарасов // Эдиториал УРСС–Москва, 2002.–352 с.

[Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В. Ф. Хорошевский // Искусственный интеллект и принятие решений. - 2008. - № 1. - С.80-97.

[Шуклин, 2005] Шуклин, Д. Е. О перспективах применения систем объектно-ориентированных баз данных и знаний в системах искусственного интеллекта. / Д.Е.Шуклин// [Электронный ресурс]. – 2005. - Режим доступа: http://www.shuklin.com/ai/ht/ru/ai05001f.aspx – Дата доступа: 27.11.2010

[ЭЗОП, 2010] ezop-project.ru [Электронный ресурс]. – 2010. - Режим доступа: http://www.ezop-project.ru. – Дата доступа: 27.11.2010
[Ark, 2010] Easy RDF and SPARQL for LAMP systems [Электронный ресурс]. – 2010. - Режим доступа: http://arc.semsol.org. – Дата доступа: 27.11.2010
[CubicWeb, 2010] CubicWeb - The Semantic Web is a construction game! [Электронный ресурс]. – 2010. - Режим доступа: http://www.cubicweb.org. – Дата доступа: 27.11.2010
[DCMI, 2010] The Dublin Core Metadata Initiative [Электронный ресурс]. – 2010. - Режим доступа: http://dublincore.org/. – Дата доступа: 27.11.2010
[FOAF, 2010] The Friend of a Friend (FOAF) project [Электронный ресурс]. – 2010. - Режим доступа: http://www.foaf-project.org. – Дата доступа: 27.11.2010
[Hawk, 2010] The Semantic Web and Agent Technologies Lab [Электронный ресурс]. – 2010. - Режим доступа: http://swat.cse.lehigh.edu/index.html. – Дата доступа: 27.11.2010
[Jena, 2010] Jena – A Semantic Web Framework for Java [Электронный ресурс]. – 2010. - Режим доступа: http://openjena.org . – Дата доступа: 27.11.2010
[Mediawiki, 2010] MediaWiki [Электронный ресурс]. – 2010. - Режим доступа: http://mediawiki.org . – Дата доступа: 27.11.2010
[Mulgara 2010] Mulgara-semantic store [Электронный ресурс]. – 2010. - Режим доступа: http://www.mulgara.org. – Дата доступа: 27.11.2010
[Neno/Fhat, 2010] NENO a semantic network programming language [Электронный ресурс]. – 2010. - Режим доступа: http://neno.lanl.gov . – Дата доступа: 27.11.2010
[Neo4j, 2010] Neo4j the graph database. [Электронный ресурс]. – 2010. - Режим доступа: http://neo4j.org. – Дата доступа: 27.11.2010
[NoSQL, 2010] NoSQL. [Электронный ресурс]. – 2010. - Режим доступа: http://nosql-database.org. – Дата доступа: 27.11.2010
[Netsukuku, 2010] Официальный сайт проекта Netsukuku. [Электронный ресурс]. – 2010. - Режим доступа: http://netsukuku.freaknet.org. – Дата доступа: 27.11.2010
[OpenCyc, 2010] OpenCyc.org. – 2010. - Режим доступа: http://www.opencyc.org. – Дата доступа: 27.11.2010
[OSTISa, 2010] Унифицированные семантические модели обработки знаний [Электронный ресурс]. – 2010. - Режим доступа: http://www.ostis.net/wiki/Проект_14. – Дата доступа: 27.11.2010
[OSTISb, 2010] Правила записи исходных текстов баз знаний на псевдоестественном языке [Электронный ресурс]. – 2010. - Режим доступа: http://www.ostis.net/wiki/Прав_записи_исх_текста_БЗ_на_scn. – Дата доступа: 27.11.2010
[OSTISc, 2010] Унифицированный способ визуализации семантических сетей [Электронный ресурс]. – 2010. - Режим доступа: http://www.ostis.net/wiki/Проект_11. – Дата доступа: 27.11.2010
[Roy Fielding, 2000] Architectural Styles and the Design of Network-based Software Architectures /Roy Thomas Fielding [Электронный ресурс]. – 2000. - Режим доступа: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm. – Дата доступа: 27.11.2010
[Semantic Mediawiki, 2010] Semantic MediaWiki is a free, open-source extension to MediaWiki [Электронный ресурс]. – 2010. - Режим доступа: http://semantic-mediawiki.org . – Дата доступа: 27.11.2010
[Sesame, 2010] OpenRDF.org, a community site to support the development of Sesame. [Электронный ресурс]. – 2010. - Режим доступа: http://www.openrdf.org. – Дата доступа: 27.11.2010
[SPARQL, 2010] SPARQL Query Language for RDF W3C Recommendation 15 January 2008 [Электронный ресурс]. – 2010. - Режим доступа: http://www.w3.org/TR/rdf-sparql-query/ . – Дата доступа: 27.11.2010
[Topaz, 2010] Topaz. [Электронный ресурс]. – 2010. - Режим доступа: http://www.topazproject.org. – Дата доступа: 27.11.2010
[VERSA, 2010] Versa 2.0 Specification [Электронный ресурс]. – 2010. - Режим доступа: http://wiki.xml3k.org/Versa/Specification. – Дата доступа: 27.11.2010
[Web Services, 2010] Web Services Resource Access Working Group [Электронный ресурс]. – 2010. - Режим доступа: http://www.w3.org/2002/ws/ra. – Дата доступа: 27.11.2010
[W3C, 2010] World Wide Web Consortium [Электронный ресурс]. – 2010. - Режим доступа: http://www.w3.org. – Дата доступа: 27.11.2010