

СРЕДСТВА УНИФИКАЦИИ ПРОЦЕССА ВЕРИФИКАЦИИ РАЗЛИЧНЫХ ВИДОВ ЗНАНИЙ

М. С. Веренич, Р. М. Зубель, П. Л. Титова

Кафедра интеллектуальных информационных технологий, Белорусский государственный университет
информатики и радиоэлектроники
Минск, Республика Беларусь

E-mail: {maxim25101996, polina.ti.19}@gmail.com, zubelruslan@mail.ru

В данной статье рассматриваются многоагентный подход верификации различных видов знаний в рамках технологии OSTIS, примеры sc-агентов верификации баз знаний.

ВВЕДЕНИЕ

База знаний является неотъемлемой частью любой интеллектуальной системы. Ее разработка является сложным и трудоемким процессом [3]. Как правило, в современных системах поддержки проектирования баз знаний вопросам верификации базы знаний уделяется относительно мало внимания, при этом алгоритмы верификации ориентированы на конкретный вид знаний. Обеспечение такой возможности позволило бы существенно увеличить продолжительность жизненного цикла системы, а автоматизация процесса верификации — существенно уменьшить количество ошибок, связанных с человеческим фактором.

I. КОЛЛЕКТИВ АГЕНТОВ ВЕРИФИКАЦИИ БАЗ ЗНАНИЙ

Машина обработки знаний любой системы, построенной по Технологии OSTIS, рассматривается как коллектив агентов, взаимодействующих через общую память. [2] В данной работе будет рассмотрен подход к унификации процесса верификации различных видов знаний, используемый в рамках подсистемы поддержки коллективного проектирования баз знаний по Технологии OSTIS. [1][3]

На данный момент в рамках СПБЗ реализованы следующие агенты верификации:

- Абстрактный sc-агент проверки выполнения условия единственности существования некоторой конструкции
- Абстрактный sc-агент проверки отсутствия циклов в рамках заданного отношения порядка
- Абстрактный sc-агент проверки соответствия мощности связок заданного отношения его типу
- Абстрактный sc-агент проверки элементов связок заданного отношения на соответствие доменам указанного отношения
- Абстрактный sc-агент проверки корректности составления множества используемых констант для заданного определения
- Абстрактный sc-агент проверки наличия определения или пояснения для понятий,

принадлежащих заданной предметной области

- Абстрактный sc-агент проверки наличия основных идентификаторов у всех понятий, имеющих системный идентификатор
- Абстрактный sc-агент проверки на наличие непустого пересечения у элементов одного разбиения
- Абстрактный sc-агент проверки наличия ключевого sc-элемента у указанной семантической окрестности
- Абстрактный sc-агент проверки наличия основных идентификаторов сущности для всех внешних языков
- Абстрактный sc-агент проверки наличия максимального класса объектов исследования у предметной области
- Абстрактный sc-агент проверки наличия доменов у отношения
- Абстрактный sc-агент проверки соотношения всех понятий с предметной областью
- Абстрактный sc-агент проверки наличия единицы измерения или шкалы у параметра
- Абстрактный sc-агент верификации заданной структуры

II. УНИФИКАЦИЯ ПРОЦЕССА ВЕРИФИКАЦИИ

Рассмотрим несколько агентов верификации более детально:

- sc-агент проверки отсутствия циклов в рамках заданного отношения порядка;
- sc-агент проверки наличия основных идентификаторов у всех понятий, имеющих системный идентификатор.

В связи с возможностью появления в базе знаний системы таких логических неточностей, как циклы, появилась необходимость в агенте поиска циклов для любого отношения порядка. Данный агент инициируется при появлении в sc-памяти представленной ниже конструкции:

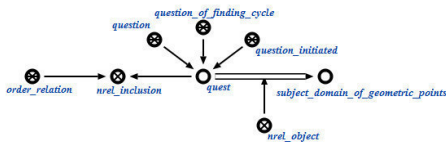


Рис. 1 – Конструкция иницирования sc-агента проверки отсутствия циклов в рамках заданного отношения порядка

Результатом работы этого агента является структура, содержащая в себе цикл, найденный в проверяемой предметной области.

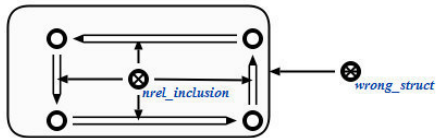


Рис. 2 – Результат работы агента проверки отсутствия циклов в рамках заданного отношения порядка

Агент проверки наличия основных идентификаторов у всех понятий, имеющих системный идентификатор предназначен для проверки сущностей, на предмет наличия у них основных идентификаторов на всех внешних языках. Единственным аргументом (param) запроса является знак структуры, элементы которой будут проверяться и иницируется при появлении в сс-памяти представленной ниже конструкции:

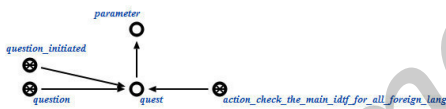


Рис. 3 – Конструкция иницирования sc-агента проверки наличия основных идентификаторов у всех понятий, имеющих системный идентификатор

Результатом выполнения команды является занесение всех сущностей, у которых есть идентификатор хотя бы на одном языке и нет идентификаторов на каких-либо других языках, известных системе, в специальное множество.

Каждый из рассмотренных агентов имеет собственное уникальное условие иницирования, что приводит к необходимости использования разработчиком отдельных команд верификации для каждого вида некорректности в базе знаний [4]. Для решения данной проблемы предлагается унифицировать условия иницирования всех sc-агентов верификации баз знаний для того, чтобы обеспечить возможность верификации указанного пользователем фрагмента базы знаний на все виды некорректностей, известных системе на данный момент. При таком подходе от пользователя не требуется знание всей типологии некорректностей, известных системе.

Рассмотрим унификацию процесса верификации на примере двух приведённых выше агентов. В результате унификации иницирование

двух агентов будет происходить в результате появления в сс-памяти только одной иницирующей конструкции.

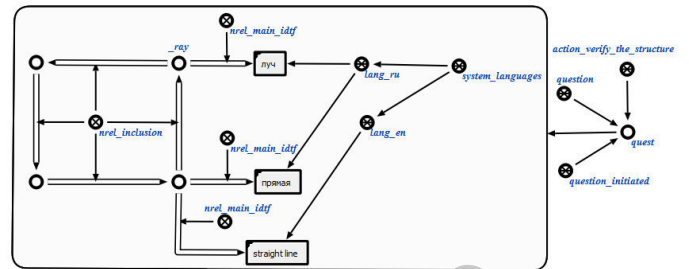


Рис. 4 – Конструкция иницирования двух агентов при использовании средств унификации

В данном случае мы получим две структуры, одна из которых является результатом работы агента проверки отсутствия циклов в рамках заданного отношения порядка, а вторая результатом работы агента проверки наличия основных идентификаторов у всех понятий, имеющих системный идентификатор.

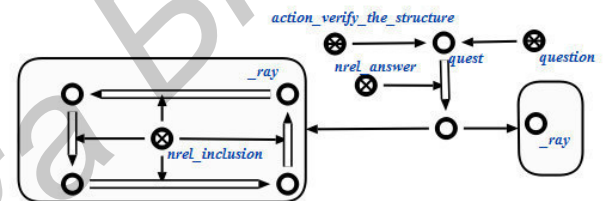


Рис. 5 – Результат работы двух агентов верификации при использовании средств унификации

Выше было наглядно продемонстрировано, что при использовании средств унификации процесса верификации различных видов знаний мы можем выявить все виды некорректностей посредством запуска одного компонента системы, который иницирует работу всех агентов верификации. Таким образом затраты временных и человеческих ресурсов на верификацию базы знаний значительно снижаются, что влечёт за собой повышение производительности, а в некоторых случаях и качества обработки знаний.

СПИСОК ЛИТЕРАТУРЫ

1. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования / В. В. Голенков, Н. А. Гулякина. – Минск БГУИР 2012, 2012. – 561 с.
2. Машина обработки знаний интеллектуальной мета-системы поддержки проектирования интеллектуальных систем / Д. В. Шункевич. – Минск БГУИР 2014, 2014. – 576 с.
3. Семантическая модель коллективного проектирования баз знаний / И. Т. Давыденко. – Минск БГУИР 2016, 2016. – 596 с.
4. Метасистема IMS.OSTIS [Электронный ресурс]. Минск, 2016. – Режим доступа: <http://ims.ostis.net>. – Дата доступа: 15.09.2016.