

# ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.42

**Евгений Рафаилович Пантелейев**

ФГБОУ ВО «Ивановский государственный энергетический университет имени В.И. Ленина», доктор технических наук, профессор кафедры программного обеспечения компьютерных систем, Россия, Иваново, телефон (4932) 26-98-60, e-mail: panteleev@gapps.ispu.ru

## Синтез вопросов контроля результатов обучения методом инверсии аксиомы модели предметных знаний

### Авторское резюме

**Состояние вопроса.** Разработка тестов контроля результатов обучения требует создания контрольных вопросов. Количество этих вопросов должно быть достаточным для достоверной оценки знаний, а их содержание – целостно отражать знания в предметной области контроля. В настоящее время используются методы автоматической генерации вопросов, основанные на использовании моделей машинного обучения и рациональных моделей представления знаний. Их применение полностью снимает количественную составляющую проблемы. Однако модели машинного обучения не гарантируют целостного представления знаний, а рациональные модели, хотя и обеспечивают целостность, не решают проблему генерации вопросов, контролирующих понимание простейших правил, которые необходимы для получения заданного результата. Цель исследования заключается в решении этой проблемы.

**Материалы и методы.** Рациональная модель предметных знаний представлена в стандарте языка онтологий web. Данный выбор обоснован глобальной распространностью стандарта, наличием доступной документации и поддержкой свободно распространяемыми приложениями, в частности редактором онтологий Protégé и средой программирования SWI Prolog, декларативная модель представления знаний которой совместима с моделью онтологий web.

**Результаты.** Предложен метод синтеза вопросов контроля по онтологической модели предметных знаний, представляющий собой инверсию аксиомы вида «если свойства сущности удовлетворяют ограничениям OWL-класса, она принадлежит этому классу», результатом которой является вопрос вида «каким ограничениям должны удовлетворять свойства сущности, если она принадлежит OWL-классу?». Инверсия является особенностью метода, который, в отличие от известных, генерирующих вопросы на понимание следствий известных причин, выполняет синтез вопросов на понимание причин, вызвавших заданное следствие. Достоверность результатов подтверждена их сопоставлением с парами «аксиома–следствие» онтологической модели.

**Выводы.** Метод обеспечивает контроль понимания правил, необходимых для получения заданного решения с помощью аксиом OWL-классов. Он был использован для разработки тестов по дисциплинам инженерной подготовки. Использование стандартного формата представления знаний позволяет применять метод для разработки вопросов контроля в других предметных областях.

**Ключевые слова:** компьютерный контроль знаний, автоматическая генерация вопросов, онтология предметной области, аксиомы вывода, инверсия

**Evgeny Raphailovich Panteleev**

Ivanovo State Power Engineering University, Doctor of Engineering Sciences (Post-doctoral degree), Professor of Computer Systems Software Department, Russia, Ivanovo, telephone (4932) 26-98-60, e-mail: panteleev@gapps.ispu.ru

## **Synthesis of control questions on learning outcomes using the axiom inversion method of subject knowledge model**

### **Abstract**

**Background.** Development of control tests of learning outcomes requires creation of test question. The number of these questions should be sufficient for reliable knowledge assessment, and their content should comprehensively reflect knowledge in the subject area of control. Currently, methods of automatic question generation based on machine learning models and rational models of knowledge representation are used. Their application completely removes the quantitative component of the problem. However, machine learning models do not guarantee a holistic representation of knowledge, and rational ones, although they provide integrity, do not solve the problem of generating questions that control understanding of the simple rules necessary to obtain a given result. The purpose of the study is to solve this problem.

**Materials and methods.** The rational model of subject knowledge is presented according to the web ontology language standard. This choice is justified due to wide use of the standard, the availability of accessible documentation and freely distributed applications, in particular, the Protégé ontology editor and the SWI Prolog programming environment, which declarative knowledge representation model is compatible with the web ontology model.

**Results.** A method for synthesizing control questions based on a model of subject knowledge has been developed. Synthesis is performed by inversion of axioms of the following form: "if the properties of an entity satisfy the constraints of an OWL class, it belongs to this class." The result of the inversion is a question of the following form: "what constraints should the properties of an entity satisfy if it belongs to an OWL class?" Inversion is a feature of the method, which, unlike the known ones that generate questions for understanding the consequences of known causes, synthesizes questions for understanding the causes of a given consequence. The validity of results is confirmed by their comparison with the "axiom-consequence" pairs of the ontological model.

**Conclusions.** The method ensures control of understanding of the rules necessary to obtain a given solution using the axioms of OWL classes. It has been used to develop tests for engineering training courses. The use of a standard format for knowledge representation allows the method to be used for developing control questions in other subject domains.

**Key words:** computerized knowledge control, automatic question generation, subject domains ontology, inference axioms, inversion

**DOI:** 10.17588/2072-2672.2025.6.086-093

**Введение.** Эффективная организация процесса обучения невозможна без использования процедур контроля результатов, которые играют роль обратной связи, используемой для перераспределения ресурсов обучения в функции отклонения измеренных результатов от целевых (этапных). Этот факт определяет широкое разнообразие форм контроля, которые должны использоваться согласованно и в соответствии с принципом методической целесообразности. Он определяет для каждой из форм нишу предпочтительного применения (фазу процесса обучения, таксономический уровень результатов), в пределах которой эффект от использования данной формы контроля превосходит связанные с ее применением издержки. В соответствии с этим принципом, компьютерные формы контроля результатов обучения целесообразно использовать в качестве инструментов входного или промежуточного контроля знаний (способности воспроизводить и классифицировать материал) и умений (способности применять простейшие правила для получения результата). В этой нише преимущества использования компьютерного контроля (объективность результатов, возможность встраивания в корпоративные системы управления, глобальная доступность) существенно превосходят затраты на раз-

работку заданий контроля. Автоматизация разработки заданий преследует цель дальнейшего уменьшения затратной составляющей. Одновременно она может способствовать повышению объективности результатов в том случае, когда «прозрачная» процедура генерации задания применяется к целостной и непротиворечивой модели предметных знаний. Ценность автоматизации тем выше, чем больше уровней таксономии результатов обучения (таксономии Блума) покрывают синтезированные задания. Таким образом, при анализе состояния вопроса в области автоматизации разработки заданий контроля целесообразно позиционировать известные разработки в пространстве критериев «тематическая полнота заданий – целостность и непротиворечивость модели знаний – количество уровней оцениваемых результатов».

Методам автоматизации разработки контрольных заданий и моделям предметных знаний, на которых они базируются, посвящено большое количество опубликованных работ, что свидетельствует об актуальности данного направления исследований. В этих публикациях можно выделить обзорные исследования [1–3], тематические разработки [4–10] и исследования смежных вопросов, например аддитивной генерации вопросов для целевых групп пользователе-

лей [11]. Чтобы сформировать общее представление о возможностях различных групп методов, начнем с анализа обзорных статей. При этом будем исходить из того, что все множество используемых этими методами моделей представления знаний можно подразделить на подмножество рациональных моделей, имеющих формальную структуру и определенные на ней аксиомы вывода, и подмножество моделей машинного обучения.

В [1] содержится анализ методов генерации контрольных заданий (вопросов) на базе онтологий, представляющих подкласс рациональных моделей представления знаний. Согласно [12], онтология — это формальная спецификация разделяемой концептуальной модели представления знаний. Здесь онтология определена как формальная спецификация концептуальной модели, т. е. явное и однозначное описание системы существенных понятий предметной области. Атрибут «разделяемая» характеризует эту модель как имеющую согласованное понимание внутри профессионального сообщества. Как отмечают авторы [1], такое позиционирование онтологий позволяет предположить, что именно эта модель явного представления знаний является предпочтительной базой для генерации объективных заданий контроля. Ее использование создает предпосылки для построения заданий, удовлетворяющих критериям тематической полноты и целостности. Что касается третьего критерия, то использование онтологий в принципе позволяет строить задания как на «воспроизведение», так и на «понимание», однако приведенные в [1] примеры ограничены интерпретацией отношений «элемент – класс», «класс – подкласс», «свойство – (под)свойство» и не предполагают использования более мощных правил, определяющих новый класс через уже существующие.

В отличие от [1], в [2, 3] обсуждается построение заданий контроля с использованием методов NLP (Natural Language Processing) и моделей машинного обучения на базе наборов априори слабо структурированных текстовых и мультимедийных данных. Необходимость реконструкции формальной структуры по документу, полнота, целостность и непротиворечивость которого могут вызывать сомнения, ставит эту группу методов в неконкурентную позицию по отношению к методам, построенным на использовании онтологий. Кроме того, вопросы, которые генерируют подобные модели, способны оценивать только результаты низшего таксономического уровня – знания.

Оценим с позиций этих предварительных выводов результаты конкретных исследований и разработок в области синтеза заданий компьютерного контроля.

Изложенный в [4] подход к генерации контрольных вопросов базируется на использовании моделей машинного обучения, следова-

тельно, не гарантирует выполнения критериев полноты, целостности и непротиворечивости представления знаний. Кроме того, синтезированные вопросы предполагают оценку знаний на уровне известных фактов.

Подход, описанный в [5], базируется на онтологической модели знаний, что создает предпосылки для выполнения критериев полноты, целостности и непротиворечивости. Данный подход позволяет интерпретировать правила для построения вопросов, которые, следовательно, могут оценивать умение применять эти правила. Например, если правило определяет инженера как специалиста, который имеет диплом, выданный инженерным вузом, то построенный в результате интерпретации этого правила вопрос может выглядеть так: «Кем является специалист с дипломом инженерного вуза?». Таким образом, вопрос оценивает умение применять правила в прямом направлении – от заданного набора причин к их следствиям. Однако в рамках данного подхода невозможно построение вопросов на понимание причин, вызвавших то или иное следствие, т. е. на интерпретацию правил в обратном направлении, например: «Какие предусловия необходимы для получения статуса инженера?». Следовательно, покрытие таксономического уровня умений путем интерпретации правил онтологии в рамках данного подхода нельзя считать полным.

В [6] также используется онтологическая модель представления знаний. Предложенный подход использует ограниченное подмножество аксиом вывода «класс–подкласс», «свойство–(под)свойство» для проверки умений применять эти аксиомы. Аксиомы, определяющие новые классы в терминах существующих через ограничения на значения свойств элементов, в рамках данного подхода не интерпретируются.

Рассматриваемый в [7] метод генерации тестовых заданий базируется на продукционной модели. Продукционные модели, как и онтологии, – это подкласс рациональных моделей представления знаний, однако, в отличие от онтологий, концептуальная целостность которых обеспечивается методологией построения «от общего к частному», продукционные модели строятся в противоположном направлении, вследствие чего подвержены трудно выявляемым ошибкам. Теоретически продукционные модели универсальны, что в рамках обсуждаемой работы подтверждается генерацией вопросов контроля фактических знаний и умений применять правила. Однако применение правил здесь контролируется на уровне результата, а не способа его получения.

В [8] в качестве информационной основы метода генерации контрольных вопросов рассматривается реляционная модель, которая не обладает семантикой, т. е. не является моделью представления знаний, поэтому полнота и

целостность такой модели не гарантированы. Кроме того, возможности генерации ограничены порождением вопросов контроля фактических знаний.

В [9] генератор вопросов контроля построен на онтологической модели OSTIS (Open Semantic Technology for Intelligent Systems: Открытые семантические технологии для проектирования интеллектуальных систем), разработанной инженерами и учеными из Беларуси. Однако представленные возможности генератора ограничены уровнем контроля фактических знаний.

В [10] представлен автоматический генератор тестов на базе онтологической модели предметных знаний консорциума web, однако авторы представляют возможности этого генератора только на уровне построения вопросов контроля фактических знаний.

Подводя итоги анализа существующей литературы, можно сделать следующие выводы:

1) преимущество методов генерации контрольных вопросов, основанных на использовании рациональных моделей представления знаний, перед используемыми для этой же цели методами машинного обучения состоит в том, что они создают предпосылки для обеспечения полноты, целостности и непротиворечивости корпуса контрольных вопросов;

2) среди рациональных моделей предпочтительной является модель, представленная на языке онтологий web, который считается глобально признанным стандартом, исчерпывающе документирован и имеет свободно распространяемые программные средства его интерпретации;

3) синтезированные вопросы охватывают два нижних уровня таксономии результатов обучения – уровень знаний и уровень способностей применять правила, причем в известных реализациях речь идет о контроле умений применять правила только в прямом направлении – от причин к следствиям.

В связи с этим целью настоящего исследования является решение проблемы генерации вопросов, контролирующих понимание правил, которые необходимы для получения заданного результата, путем разработки метода инверсии аксиом OWL-классов.

**Методы исследования.** Обоснованный выше выбор модели представления знаний для решения задачи синтеза контрольных вопросов требует описания существенных для решения этой задачи свойств модели. Онтологическая модель консорциума W3C построена с использованием стека стандартов, определяющих единый синтаксис и совместно используемую семантику языка описания моделей, благодаря использованию общих словарей. Стандарты поддерживают

реляционный синтаксис записи отношений в виде троек RDF «субъект–предикат (свойство)–объект» и семантику, которая обеспечивает уникальность элементов модели по принципу «один элемент–один глобально уникальный адрес» (ссылочная семантика) и стандартные правила вывода следствий из аксиом модели (семантика вывода). Именно семантика вывода является существенным с точки зрения решаемой задачи качеством модели. Она базируется на использовании понятия класса как абстрактного множества сущностей и отношений, в которых фигурирует это понятие. Одно из таких отношений – это отношение «подкласс»: subClassOf. Связанная с ним аксиома вывода гласит: сущность, принадлежащая подклассу, принадлежит также и родительскому классу. Второй тип отношения – это отношение эквивалентности определяемого класса анонимному классу, для которого заданы ограничения на значения свойств принадлежащих ему сущностей. Например, по классификации МАГАТЭ<sup>1</sup>, класс реакторов GCR (gas-cooled reactor) может быть определен как множество реакторов, теплоносителем в которых является газообразный агент, а замедлителем – графит. Имплементация данного определения в редакторе онтологий Protégé и построенные в результате его интерпретации в контексте заданных значений свойств сущностей класса «Реакторы» следствия показаны на рис. 1. Построенный на основе этого определения «прямой» вопрос может выглядеть так: «Какие реакторы относятся к классу GCR?», а «инверсный» вопрос – «Почему реактор Magnox относится к классу GCR?»

Редактор Protégé сохраняет онтологию (по желанию пользователя – вместе с результатами вывода) в одном из стандартных текстовых форматов (XML, TTL и т.п.). Фрагмент формата TTL, содержащий определение сущности Magnox с учетом заданных (asserted) и выведенных (inferred) значений свойств, имеет следующий вид:

```
:Magnox rdf:type
owl:NamedIndividual,
:GCR,
:IAEA_classes,
:reactor;
:heat_carrier :CO2;
:moderator :C;
rdfs:label "Magnox".
```

К заданным относятся, например, свойства, определяющие принадлежность сущности к классам «Реакторы» и «Классификация МАГАТЭ», а также тип теплоносителя (углекислый газ) и замедлителя (графит). К выведенным свойствам относится принадлежность к классу GCR.

<sup>1</sup> <https://pris.iaea.org/PRIS/Glossary.aspx> Глоссарий терминов, используемых в базе данных PRIS. Дата обращения: 13 ноября 2016.

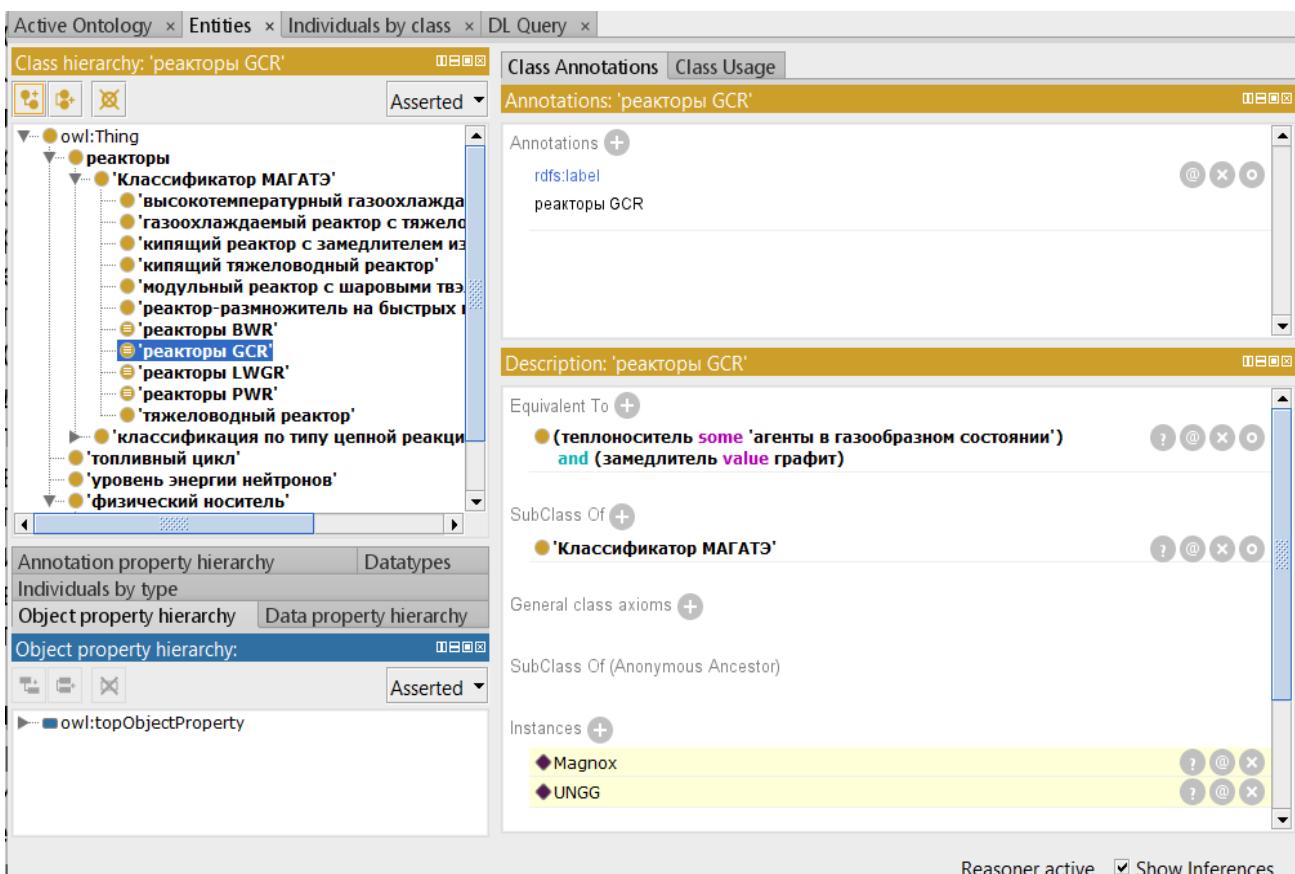


Рис. 1. Запись и интерпретация определений ограниченного класса в редакторе Protégé

Программной средой реализации обсуждаемого ниже метода инверсии выбран язык Пролог, точнее, его версия SWI Prolog<sup>2</sup>. В качестве обоснования выбора можно указать на следующие факторы. Во-первых, Пролог – это декларативный язык отношений, т. е. реляционный язык, и сопряжение двух реляционных моделей будет более естественным и лаконичным, чем, например, интерпретация онтологической модели процедурным языком Python. Например, в результате импорта онтологии в программу на Прологе тройки «субъект–предикат–объект» становятся частью этой программы – обычными утверждениями Пролога в формате фактов – и как таковые могут непосредственно использоваться при построении утверждений формата правил и при выполнении запросов. Во-вторых, SWI Prolog располагает набором библиотек для работы с онтологиями<sup>3</sup> и создания интернет-приложений, скрывающих от разработчика низкоуровневые детали.

**Результаты исследования.** Прежде чем излагать суть метода инверсии, определим базовые отношения для записи owl-классов, используя следующие соглашения. Для обозначения именованных классов owl будем использовать курсив прописных букв латинского алфавита,

например *A*, *C*, *D*, кроме зарезервированных *B*, *P* и *V*. Сущности именованных классов будем обозначать строчным курсивом соответствующей прописной буквы, например *a* ∈ *A*. Для обозначения анонимных классов, с помощью которых определяются ограничения на значения свойств owl-классов, будем использовать курсив прописной буквы *B*, а для обозначения всего множества анонимных классов – жирный курсив **B**. Свойства сущности обозначим курсивом прописной буквы *P*. Для обозначения области значений свойства *P* будем использовать прописной курсив *V*.

Классы owl определяются через отношения эквивалентности с анонимными классами, которые и содержат описание ограничений на значения свойств. Например, класс водо-водяных реакторов ВВР определяется как эквивалент пересечения множеств сущностей двух анонимных классов, один из которых устанавливает соответствующее ограничение на значение агента-охладителя, а второй – на значение агента-замедлителя.

В общем случае определение класса owl может быть записано следующим образом:

$$C = \{ \begin{matrix} B \\ B_1 \cap B_2 | B_1 \cup B_2 | B_1 \setminus B_2 \end{matrix} \} \quad (1)$$

<sup>2</sup> <https://www.swi-prolog.org/>

<sup>3</sup> Библиотекой Owlready2 того же назначения располагает и Python.

Каждый из эквивалентных классов в (1) может быть определен через вложенные ограничения или быть терминальным. В последнем случае он непосредственно определяет множество сущностей, удовлетворяющих конкретному типу ограничения:

$$B_i^T = \{b | P_i(b)\}. \quad (2)$$

В методе инверсии используются следующие типы ограничений.

Сущность  $v$  принадлежит классу  $C$ , если хотя бы одно значение  $v$  ее свойства  $P$  принадлежит классу  $V$ :

$$\exists v \in V: v = P_i(c) \rightarrow c \in C. \quad (3)$$

Сущность  $v$  принадлежит классу  $C$ , если все значения  $v$  ее свойства  $P$  принадлежат классу  $V$ :

$$\forall v \in V: v = P_i(c) \rightarrow c \in C. \quad (4)$$

Сущность  $v$  принадлежит классу  $C$ , если значение  $v$  ее свойства  $P$  равно константе  $v_c$  из определения ограничения:

$$\exists v \in V: v = P_i(c), v = v_c \rightarrow c \in C. \quad (5)$$

Принадлежность элементов owl-классу устанавливается путем рекуррентной интерпретации (1) с учетом (2)–(5). Эту работу в редакторе Protégé выполняет компонент Reasoner. Результаты такого вывода для owl-класса CGR представлены в виде списка сущностей класса в нижней части рис. 1 и могут быть экспортированы в стандартном формате в составе онтологии для использования приложением, выполняющим синтез контрольных вопросов.

Таким образом, метод инверсного синтеза получает на входе аксиомы owl-классов (ограничения, определяющие условия принадлежности сущности классу) и следствия из этих аксиом, констатирующие принадлежность конкретных сущностей конкретным классам. Эта информация традиционно используется для построения вопросов типа «Какие сущности принадлежат owl-классу?», ориентированных от причин к следствию. Предполагается, что правильный ответ может быть получен в результате интерпретации аксиом класса в контексте значений свойств сущностей и квалифицирован как умение применять аксиомы.

Предлагаемый метод ориентирован на оценку умения реконструировать причины по заданному следствию, т. е. на синтез вопросов типа: «Почему конкретная сущность принадлежит owl-классу? Какие аксиомы класса устанавливают факт принадлежности?» или «Какие аксиомы входят в определение owl-класса?».

Компьютерный контроль, помимо построения вопроса, предполагает формальную регламентацию ответа. Открытый тип вопросов предусматривает ручной ввод ответа, тогда как закрытый – только выбор из предопределенного множества вариантов. Использование вопросов

открытого типа усложнено необходимостью регламентации синтаксиса ответа, поэтому в обсуждаемом методе принят закрытый формат одиночного/множественного выбора.

Таким образом, в рамках упомянутых выше шаблонов контрольных вопросов основная задача инверсного синтеза сводится к построению множества предопределенных вариантов ответа  $Answer$ , представляющего собой объединение непересекающихся подмножеств правильных ответов  $True$  и неправильных, но правдоподобных  $False$  (дистракторов):

$$Answer = True \cup False, True \cap False = \emptyset.$$

Подмножество правильных ответов  $True$  может быть построено на базе определения ограничений  $B_k$  того класса  $C_k$ , аксиомы которого запрашиваются в контексте вопроса об обосновании условий принадлежности сущности к  $C_k$ , или в контексте вопроса о реконструкции определения этого класса.

В качестве полного множества  $Answer$  метод использует объединение аксиом  $\cup B_i$  для множества ограниченных классов  $C$  ( $C_k \in C$ ), являющихся подклассами  $S$ :

$$\forall C_i \in C: C \text{ sub } S,$$

где  $sub$  – отношение  $subClassOf$ .

Тогда подмножество дистракторов  $False$  определяется как разность  $Answer$  и  $True$ :

$$False = Answer \setminus True.$$

С учетом сказанного построение множества контрольных вопросов методом инверсии аксиом owl-классов может быть реализовано следующей последовательностью операций:

1. Построить множество owl-классов  $C$  для импортированной онтологии.
2. Для каждого  $C_i \in C$ :
  - 2.1. Построить множества  $Answer_i, True_i, False_i = Answer_i \setminus True_i$
  - 2.2. Заполнить текстовый шаблон варианта ответа «свойство  $P$  – значение  $v \in V$ »
  - 2.3. Для каждого  $c_{ik} \in C_i$ :
    - 2.3.1. Заполнить текстовый шаблон вопроса « $c_{ik} \in C_i$ , так как:»
    - 2.3.2. Экспортировать контрольный вопрос и варианты ответа на него в онтологию заданий контроля

Приведенный псевдокод генерирует вопросы на обоснование принадлежности сущности owl-классу. Псевдокод для генерации вопросов на реконструкцию свойств owl-класса выглядят как упрощение вышеприведенного заменой внутреннего цикла по сущностям класса инструкцией генерации текста вопроса в шаблоне: «Принадлежность к  $C_i$  предполагает:». Второстепенные детали процесса, такие как построение текста конкатенацией значений свойств `rdfs:label` для соответствующих элементов онтологии, в коде не рассматриваются.

Описанный метод реализован на языке SWI Prolog и обеспечивает сохранение процедуры контроля в виде множества тематически сгруппированных вопросов (признаком группировки является принадлежность вопроса к оценке конкретного owl-класса) в файле онтологии стандартного формата. Это открывает возможности для редактирования процедуры контроля любым приложением, работающим со стандартными форматами онтологий, например Protégé, а также для использования синтезированной онтологии контрольных заданий веб-сервером, который в рамках данного исследования также был реализован средствами SWI-Prolog (рис. 2).

**Тема: реакторы GCR**

Текущий результат: 1 вопросов, 1 правильных ответов

**Вопрос: Magnox - это реакторы GCR, так как:**

- теплоноситель: углекислый газ
- теплоноситель: легкая вода под давлением
- спектр нейтронов: быстрые нейтроны
- спектр нейтронов: тепловые нейтроны
- замедлитель: кипящая легкая вода
- замедлитель: легкая вода под давлением
- замедлитель: графит
- теплоноситель: кипящая легкая вода

Отправить

Рис. 2. Отображение контрольного вопроса в окне браузера

**Выводы.** Поставленная цель исследования, состоящая в обеспечении возможности контроля понимания правил, необходимых для получения заданного решения с помощью аксиом owl-классов, достигнута. Достоверность полученных результатов подтверждена разработкой тестов «Ядерные реакторы», «Алгоритмы сжатия данных» по дисциплинам инженерной подготовки ИГЭУ. Практическое использование результатов предполагает наличие навыков работы с редактором онтологий Protégé для разработки онтологий предметных знаний. Встроенные в редактор инструменты обеспечивают контроль целостности и непротиворечивости представленных в онтологии знаний.

Безусловно, освоение навыков разработки онтологий в среде редактора Protégé несколько увеличивает трудозатраты на разработку тестов. Однако следует заметить, что, во-

первых, применение «ручной» технологии предполагает, что модель предметных знаний, пусть и в неявном виде, присутствует в голове разработчика, так что речь идет лишь о затратах на формализацию этой модели. Во-вторых, эти затраты многократно перекрываются за счет автоматической генерации вопросов, которые требуют объяснения фактов, вытекающего из правил. Количество фактов, соответствующих существенным, может на порядок превышать количество классов, которые определяют правила и которым в силу этих правил сущности принадлежат. Кроме того, одни и те же сущности можно классифицировать по нескольким основаниям. Например, помимо классификатора МАГАТЭ, ядерные реакторы можно классифицировать по виду топлива, по конструкции и по другим признакам. Это еще больше увеличивает количество автоматически генерируемых вопросов. Таким образом, затраты на формализацию ограниченного набора правил с избытком компенсируются автоматической генерацией вопросов, которые требуют объяснения многочисленных фактов, вытекающих из этих правил.

Алгоритмы, реализующие предложенный метод синтеза, обеспечивают генерацию исчерпывающего множества вопросов одиночного или множественного выбора. Стандартный формат представления результатов синтеза позволяет использовать их как в режиме ручной корректировки в редакторе Protégé, так и для их интерпретации веб-сервером, который был разработан для экспериментального подтверждения достоверности результатов данного научного исследования.

#### Список литературы

1. Papasalouros A., Chatzigiannakou M. Semantic Web and Question Generation: An Overview of the State of the Art // International Association for Development of the Information Society, 2018.
2. Rakangor S., Ghodasara Y.R. Literature review of automatic question generation systems // International journal of scientific and research publications. – 2015. – Vol. 5, No. 1. – P. 1–5.
3. Automatic question generation and answer assessment: a survey / B. Das, M. Majumder, S. Phadikar, A.A. Sekn // Research and Practice in Technology Enhanced Learning. – 2021. – Vol. 16, No. 1. – P. 5. DOI: <http://dx.doi.org/10.1186/s41039-021-00151-1>
4. Blšták M., Rozinajová V. Automatic question generation based on sentence structure analysis using machine learning approach // Natural Language Engineering. – 2022. – Vol. 28, No. 4. – P. 487–517. DOI: <http://dx.doi.org/10.1017/S1351324921000139>
5. Zoumpatianos K., Papasalouros A., Kotis K. Automated Transformation of SWRL Rules into Multiple-Choice Questions // FLAIRS. – 2011.
6. Papasalouros A., Kanaris K., Kotis K. Automatic Generation Of Multiple Choice Questions From Domain Ontologies // e-Learning. – 2008. – Vol. 1. – P. 427–434.

7. Сергушичева А.П., Швецов А.Н. Метод построения компьютерных тестирующих систем для обучения в техническом вузе // Алгоритмы, методы и системы обработки данных. – 2005. – № 10. – С. 168–180.
8. Кручинин В.В., Морозова Ю.В. Модели генераторов вопросов для компьютерного контроля знаний // Открытое и дистанционное образование. – 2004. – № 2. – С. 52–62.
9. Ли В. Онтологический подход к автоматической генерации вопросов в интеллектуальных обучающих системах // Доклады Белорусского государственного университета информатики и радиоэлектроники. – 2020. – Т. 18, № 5. – С. 44–52. DOI: <http://dx.doi.org/10.35596/1729-7648-2020-18-5-44-52>
10. Vinu E.V., Kumar P.S. Automated generation of assessment tests from domain ontologies // Semantic Web. – 2015. – Vol. 8, No. 6. – P. 1023–1047.
11. Le N.T. Using Semantic Web for Generating Questions: Do Different Populations Perceive Questions Differently? // Transactions on Computational Collective Intelligence XVIII. – Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. – P. 1–19.
12. Studer R., Benjamins R., Fensel D. Knowledge Engineering: Principles and methods // Data and knowledge engineering. – 1998. – Vol. 25. – P. 161–197.
- References**
1. Papasalouros, A., Chatzigiannakou, M. Semantic Web and Question Generation: An Overview of the State of the Art. International Association for Development of the Information Society, 2018.
  2. Rakangor, S., Ghodasara, Y.R. Literature review of automatic question generation systems. *International journal of scientific and research publications*, 2015, vol. 5, no. 1, pp. 1–5.
  3. Das, B., Majumder, M., Phadikar, S., Sekn, A.A. Automatic question generation and answer assessment: a survey. *Research and Practice in Technology Enhanced Learning*, 2021, vol. 16, no. 1, p. 5. DOI: <http://dx.doi.org/10.1186/s41039-021-00151-1>
  4. Blšták, M., Rozinajová, V. Automatic question generation based on sentence structure analysis using machine learning approach. *Natural Language Engineering*, 2022, vol. 28, no. 4, pp. 487–517. DOI: <http://dx.doi.org/10.1017/S1351324921000139>
  5. Zoumpatianos, K., Papasalouros, A., Kotis, K. Automated Transformation of SWRL Rules into Multiple-Choice Questions. *FLAIRS*, 2011.
  6. Papasalouros, A., Kanaris, K., Kotis, K. Automatic Generation Of Multiple Choice Questions From Domain Ontologies. *e-Learning*, 2008, vol. 1, pp. 427–434.
  7. Sergushicheva, A.P., Shvetsov, A.N. Metod postroeniya kompyuternykh testiruyushchikh sistem dlya obucheniya v tekhnicheskem vuze [Method of constructing computer testing systems for training in a technical university]. *Algoritmy, metody i sistemy obrabotki dannykh*, 2005, no. 10, pp. 168–180.
  8. Kruchinin, V.V., Morozova, Yu.V. Modeli generatorov voprosov dlya kompyuternogo kontrolya znanii [Models of question generators for computer-based knowledge control]. *Otkrytoe i distantsionnoe obrazovanie*, 2004, no. 2, pp. 52–62.
  9. Li, V. Ontologicheskiy podkhod k avtomaticheskoy generatsii voprosov v intellektual'nykh obuchayushchikh sistemakh [Ontological approach to automatic question generation in intelligent learning systems]. *Doklady Belorusskogo gosudarstvennogo universiteta informatiki i radioelektroniki*, 2020, vol. 18, no. 5, pp. 44–52. DOI: <http://dx.doi.org/10.35596/1729-7648-2020-18-5-44-52>
  10. Vinu, E.V., Kumar, P.S. Automated generation of assessment tests from domain ontologies. *Semantic Web*, 2015, vol. 8, no. 6, pp. 1023–1047.
  11. Le, N.T. Using Semantic Web for Generating Questions: Do Different Populations Perceive Questions Differently? Transactions on Computational Collective Intelligence XVIII. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 1–19.
  12. Studer, R., Benjamins, R., Fensel, D. Knowledge Engineering: Principles and methods. *Data and knowledge engineering*, 1998, vol. 25, pp. 161–197.