

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Направление подготовки: 09.04.04 – Программная инженерия
Магистерская программа: Разработка программно-информационных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
МОДЕЛЬ ВОПРОСНО-ОТВЕТНОЙ СИСТЕМЫ НА ОСНОВЕ
ОНТОЛОГИЧЕСКОГО ПОДХОДА

Обучающийся 2 курса
группы 11-922

Бирюков А.М.

Научный руководитель

к. ф.-м. н., доцент,
доцент кафедры интеллектуальных технологий поиска

Липачёв Е.К.

Директор ИТИС КФУ

канд. техн. наук

Абрамский М.М.

Казань – 2021

Оглавление

Аннотация	4
Введение.....	6
Глава 1. Основные определения и обозначения	11
Глава 2 Постановка задачи.....	15
Глава 3. Назначение вопросно-ответных систем.....	17
Глава 4. Модель вопросно-ответной системы.....	21
4.1. Архитектура и протекающие процессы предлагаемой модели	21
4.2. Представление данных онтологии по модели RDF.....	24
4.3. Структура онтологии, применяемой вопросно-ответной системой.....	26
4.4. Взаимодействие с RDF-графами	28
4.5. Типы вопросительных предложений	29
4.6. Алгоритм обработки вопроса пользователя.....	32
Глава 5. Реализация прототипа модели вопросно-ответной системы, основанной на онтологическом подходе	36
5.1. Программная реализация	36
5.2. Элементы адаптивности	37
5.2.1. Карта преобразований	37
5.2.2. Конфигурируемость.....	40
5.3. Тестирование	42
5.4. Наполнение карты преобразований	44
5.5. Примеры работы вопросно-ответной системы.....	48
Заключение	54
Список использованных источников	56

Приложение 1. Карта преобразований

Аннотация

Наиболее частый сценарий применения вопросно-ответных систем состоит в следующем: пользователь системы в произвольной форме на естественном языке подаёт на вход вопрос (например, «что входит в состав класса вектор?») и получает ответ, содержащий набор определений и, желательно, список веб-страниц (элементов аннотирования концептов онтологии), подтверждающих корректность ответа.

Для того чтобы обеспечить взаимодействие пользователя с программным комплексом, и сокрытой внутри вопросно-ответной системы онтологией в частности, на естественном языке, необходимо проработать систему преобразования запросов пользователя во внутреннее представление системы, а также разработать способ обратного преобразования – из ответа системы в ответ на естественном языке.

В работе решаются следующие основные задачи:

- Разработать систему преобразования вопроса пользователя в запрос к онтологии.
- Выстроить адаптивный процесс взаимодействия вопросно-ответной системы с онто-семантическими сетями.
- Разработать систему преобразования результатов поиска по онтологии в ответ на естественном языке.
- Разработать рабочий прототип модели вопросно-ответной системы на основе онтологического подхода.

Результатом работы является модель вопросно-ответной системы на основе онтологического подхода, отработанная на онтологии профессионального математического знания OntoMathPro, с возможностью дальнейшего её совершенствования как в рамках расширения и развития

базы знаний OntoMathPro, так и использования с любой другой базой знаний, построенной по онтологическому подходу.

В рамках выполнения данной магистерской диссертации в прототип модели вопросно-ответной системы были заложены как возможность отвечать на вопросы пользователя, так и посредством естественно-языкового запроса иметь возможность получать сведения о самой используемой онтологии. Данная функциональная возможность представляется, например, с помощью вопроса к системе «Сколько всего концептов в онтологии», из ответа на который можно узнать количественное наполнение используемой онтологии.

Введение

В настоящее время мы столкнулись с проблемой колоссального роста объёма информации, с которой необходимо работать и, без привлечения интеллектуальных сервисов, обработать эту информацию не представляется возможным, что подтверждено многочисленными данными, например в ряде исследований [1, 2], проведённых Университетом Южной Калифорнии Экономической комиссии ООН для Латинской Америки и Карибского бассейна.

Исходя из вышесказанного возникает логичная задача: обеспечить пользователю удобный и быстрый способ поиска интересующих его фактов в бесконечно растущем объёме информации. Для решения этой задачи разрабатываются специальные программные комплексы, так называемые вопросно-ответные системы, которые призваны упрощать задачу поиска информации на больших объёмах текста. Однако не каждый человек имеет достаточную квалификацию, чтобы самостоятельно строить запросы к подобным системам в «понятном» им формате, поэтому современные вопросно-ответные системы развиваются в направлении большей человеко-ориентированности, чтобы предоставлять пользователю возможность задавать вопрос на естественном языке, конечно же, с определёнными оговорками на формализацию вопросного предложения, и получать ответ от системы в такой же естественно-языковой форме. Актуальность проблемы разработки качественных вопросно-ответных систем напрямую подтверждается быстро растущим числом научных трудов, посвящённых формированию синтаксических и общезыковых анализаторов для вопросно-ответных систем [3–7], новых онтологий конкретных областей знаний и их расширения посредством связывания с уже существующими онтологиями общих знаний [8–12], а также работ, предлагающих непосредственно модели

и программные реализации вопросно-ответных систем с применением семантических графов или основанных на онтологическом подходе [13–17].

Под вопросно-ответной системой (Question-answering system) понимается информационная система, способная принимать вопросы и отвечать на них на естественном языке. В общем случае такая система представляет собой программный комплекс, состоящий из большого объема справочных материалов и интеллектуальных подсистем, использующих естественно-языковой интерфейс. На вход вопросно-ответная система ожидает получить вопрос на естественном языке, обрабатывает его и, преобразовав релевантные наборы данных в конечный результата запроса, выдаёт пользователю естественно-языковой ответ. Источниками данных системы могут выступать как локальные хранилища в различных формах и форматах представления, так и другие системы или сама сеть Интернет.

Особый интерес, в свете развития программных и аппаратных средств, вызывают вопросно-ответные системы, в основе своей использующие онтологии, онто-семантические графы.

Под онтологией в срезе развития именно вопросно-ответных систем, понимается попытка всеобъемлющей и подробной формализации некоторой области знаний с помощью концептуальной схемы. Обычно такая схема состоит из структуры данных, содержащей все релевантные классы объектов, их связи и правила (теоремы, ограничения), принятые в этой области. Этот термин в информатике является производным от древнего философского понятия «онтология».

Применение онтологий позволяет предоставлять достаточную полноту сведений по предметной области, при сохранении всех связей характеристик концептов, применяемых в ней.

В качестве подготовки к началу работ было принято решение ближе ознакомиться с библиометрическими и наукометрическими измерениями

ценности научных статей, принципами их написания, не писаными правилами их отбора для использования в качестве библиографии.

Наукометрия – дисциплина, изучающая эволюцию науки через многочисленные измерения и статистическую обработку научной информации (смотри подробнее [18]). Применительно к гуманитарным и общественным наукам – всё, что можно измерить – количество публикаций и их цитирование. Такие показатели называются «библиометрическими».

Основа наукометрии – библиографические ссылки и публикации. В книге [18] отмечено: «Библиографические ссылки отражают ту интеллектуальную атмосферу, в которой проходило создание публикации», «Будем называть наукометрией количественные методы изучения развития науки как информационного процесса» (смотри подробнее [19]).

Для наиболее быстрого и при этом продуктивного ознакомления с принципами и подходами, предлагаемыми наукометрией, была прослушана группа курсов Clarivate [18, 20], посвящённых Наукометрии, наукометрическим показателям, принципам оформления аннотаций и списков литературы, поиска в них.

Работа посвящена разработке вопросно-ответной системы в области математического знания. Её задача – в он-лайн режиме, используя онтологический ресурс, отвечать на вопросы, поставленные пользователем вопросно-ответной системы, основанной на онтологии математических знаний. В основе предлагаемой вопросно-ответной системы применена онтология профессионального математического знания OntoMathPro. Отметим, что эта онтология, созданная в Казанском (Приволжском) федеральном университете, предоставлена в открытый доступ (<https://github.com/CLLKazan/OntoMathPro>) и постоянно расширяется (смотри, например, [8, 21]).

Создана вопросно-ответная система, которая способна обрабатывать пользовательские вопросы в естественно-языковой форме, искать ответы в онтологии профессионального математического знания OntoMathPro с помощью SPARQL запросов и возвращать результат в понятной человеку форме.

Структура настоящей работы состоит из аннотации, введения, 5 глав, списка литературы и одного Приложения.

В первой главе представлены основные определения и обозначения, используемые в работе. В следующей главе определяются объект и предмет исследования, и цель настоящей работы. В главе 3 рассматривается назначение вопросно-ответных систем, причины возникновения задачи по их разработке и совершенствованию, обуславливается ценность и значимость такого рода разработок. Четвёртая глава включает в себе описание представленной в текущей работе модели вопросно-ответной системы. Последняя пятая глава посвящена реализации прототипа модели вопросно-ответной системы, основанной на онтологическом подходе.

Список литературы состоит из 33 наименований, включая 19 из интервала 2016-2021 гг. Список литературы сформирован в соответствии с ГОСТ Р 7.0.100-2018 [32].

Приложение 1 содержит использованную при отладке модели вопросно-ответной системы на онтологии OntoMathPro карту преобразований вопросов и ответов пользователей.

Блок-схемы, приведённые в магистерской диссертации, сформированы в соответствии с ГОСТ 19.701-90 (ИСО 5807-85) [33]. Диаграммы сформированы в соответствии с нотацией UML.

Проектирование диаграмм и блок-схем проводилось посредством пользования Draw.io – Diagrams.net (<https://app.diagrams.net/>) – он-лайн инструмента для создания диаграмм, графиков и блок-схем.

Разработка программного решения выполнялась в интегрированной среде разработки Visual Studio 2019 с использованием объектно-ориентированного языка программирования C#. Версия используемого фреймворка – ASP.NET Core 5.0.

В качестве инструмента для логирования применён NuGet пакет Serilog. В качестве контейнера инверсии управления (IoC-контейнера) выбран DryIoc – это быстрый, небольшой, полнофункциональный IoC-контейнер для .NET. Для работы с локальными данными используется текстовый формат данных JSON.

Запросы к точке доступа онтологии производятся на языке запросов к данным SPARQL с применением технологии dotNetRDF. Взаимодействие с пользователем производится через Swagger.

В качестве системы контроля версий была выбрана git, исходный код разработанной модели вопросно-ответной системы доступен в открытом доступе на GitHub (<https://github.com/EgoPingvina/Question-answering-system>). Также в процессе написания кода использовался GitKraken — GUI git клиент. Разработка проводилась по методологии TDD.

Апробация работы проводилась в рамках выступлений:

- на научных семинарах (научный руководитель Е.К. Липачёв);
- на конференции в рамках форума IFME`2021 25 марта.

Результаты магистерской диссертации частично опубликованы в работе [15]. Подготовлена к печати статья по результатам выполненных в рамках магистерской диссертации работ.

Глава 1. Основные определения и обозначения

В этой главе приводятся основные определения и обозначения, касающиеся исследования, проведённого в настоящей работе. Использовались следующие научные источники [3–31].

ВОПРОСНО-ОТВЕТНАЯ СИСТЕМА (QA-СИСТЕМА; от англ. QA – англ. QUESTION-ANSWERING SYSTEM) – информационная система, способная принимать вопросы и отвечать на них на естественном языке, другими словами, это система с естественно-языковым интерфейсом.

ГИПЕРОНИМ – слово с более широким значением, выражающее общее, родовое понятие, название класса (множества) предметов (свойств, признаков).

ГИПОНИМ – понятие, выражающее частную сущность по отношению к другому, более общему понятию.

ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ (англ. Sequence Diagram) — UML-диаграмма, на которой для некоторого набора объектов на единой

временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актёров (действующих лиц) информационной системы в рамках прецедента.

Для того чтобы обеспечить взаимодействие пользователя с программным комплексом, и сокрытой внутри вопросно-ответной системы онтологией в частности, на естественном языке, необходимо проработать систему преобразования запросов пользователя во внутреннее представление системы, а также разработать способ обратного преобразования – из ответа системы в ответ на естественном языке. Для решения поставленной задачи был разработан способ хранения данных, названный КАРТОЙ ПРЕОБРАЗОВАНИЙ.

НАУКОМЕТРИЯ – дисциплина, изучающая эволюцию науки через многочисленные измерения и статистическую обработку научной информации. Применительно к гуманитарным и общественным наукам – всё, что можно измерить – количество публикаций и их цитирование. Такие показатели называются «библиометрическими».

ОНТО-СЕМАНТИЧЕСКИЙ ГРАФ – ориентированный граф, вершинами которого являются синтаксемы, выделенные в анализируемом тексте, в совокупности с соответствующей им информацией из онтологии, а именованные дуги графа определяют названия семантических отношений, связывающих эти синтаксемы. Направление дуг онто-семантического графа определяет последовательность аргументов таких зависимостей.

ОНТОЛОГИЯ – это попытка всеобъемлющей и подробной формализации некоторой области знаний с помощью концептуальной схемы. Обычно такая схема состоит из структуры данных, содержащей все релевантные классы объектов, их связи и правила (теоремы, ограничения), принятые в этой области. Этот термин в информатике является производным от древнего философского понятия «онтология».

СЕМАНТИКА – раздел языкознания, изучающий значение единиц языка, прежде всего его слов и словосочетаний. В более общем смысле, семантика определяет смысл знаков (образов, обозначений) и их сочетаний.

СЕМАНТИЧЕСКАЯ СЕТЬ (СМЫСЛОВАЯ СЕТЬ) – модель предметной области, представленная в виде графа, вершинами которого являются понятия, а дуги (ребра) – отношения между ними.

СЕМАНТИЧЕСКИЙ АНАЛИЗ – этап в последовательности действий алгоритма автоматического понимания текстов, заключающийся в выделении семантических отношений, формировании семантического представления текстов.

СИНОНИМЫ – слова, звучание и написание которых различно, но при этом у них похожее значение (например, полином – многочлен). Чаще всего они принадлежат к одной и той же части речи.

ТОЧКА ДОСТУПА SPARQL – это точка присутствия в сети HTTP, которая способна принимать и обрабатывать запросы протокола SPARQL. Она идентифицируется URL-адресом, который обычно называют URL-адресом конечной точки SPARQL.

Ресурсом в RDF может быть любая сущность — как информационная (например, изображение или веб-ресурс), так и неинформационная (например, теорема, человек или книга). Утверждение, высказываемое о ресурсе, имеет вид «субъект — предикат — объект» и называется ТРИПЛЕТОМ.

ЯЗЫК ЗАПРОСОВ SPARQL – это декларативный язык запросов (как SQL) для исполнения операций манипуляции и определения данных на информации, представленной коллекцией утверждений (триплетов), определённых по RDF.

RESOURCE DESCRIPTION FRAMEWORK (RDF, «СРЕДА ОПИСАНИЯ РЕСУРСА») — это разработанная консорциумом Всемирной паутины модель для представления данных, в особенности — метаданных. RDF представляет сведения о концептах в виде, пригодном для обработки программными средствами. RDF является частью концепции семантической паутины.

Глава 2 Постановка задачи

Объектом исследования является моделирование вопросно-ответных систем в области математического знания.

Предмет исследования – применение онтологического подхода в моделях вопросно-ответных систем.

Целью работы является разработка модели вопросно-ответной системы, основанной на имеющихся онтологиях математического знания, и реализация предложенной в работе модели с использованием онтологии математического знания OntoMathPro.

Под вопросно-ответной системой понимаем информационную систему, способную принимать вопросы пользователя и отвечать на них на естественном языке.

Выделим основные задачи:

- Разработать систему преобразования вопроса пользователя в запрос к онтологии.
- Выстроить адаптивный процесс взаимодействия вопросно-ответной системы с онто-семантическими сетями.
- Разработать систему преобразования результатов поиска по онтологии в ответ на естественном языке.
- Разработать прототип модели вопросно-ответной системы на основе онтологического подхода.

Результатом работы является модель вопросно-ответной системы на основе онтологического подхода, отработанная на онтологии математического знания OntoMathPro [21], с возможностью дальнейшего её совершенствования как в рамках расширения и развития базы знаний OntoMathPro, так и использования с любой другой базой знаний, построенной по онтологическому подходу.

Глава 3. Назначение вопросно-ответных систем

По мнению ведущих экспертов в области больших данных и машинного обучения по состоянию на 2021 год, точка не возврата в вопросе коммуникации человека и различных информационных систем пройдена (смотри, например, [22]). Прошедшие годы ознаменовали стремительное развитие таких направлений как интернет вещей, беспилотные транспортные средства, принципиально новые системы обучения, технологии распознавания образов на базе машинного обучения. А несколько лет назад широкое распространение получили программы сопровождения пользователя в текстовых и голосовых чатах.

Вместе с тем все эти направления в совокупности открывают потенциал новой (а лучше сказать, хорошо забытой) технологии — экспертные системы данных. Как правило, с точки зрения обывателей, данный пласт решений не столь заметен, так как их работа, как правило, применяется на стороне серверных приложений. Но, так или иначе, вопрос

организации связанности и навигации данных между собой становится всё более острым и востребованным. Абсолютно очевидно, что традиционные методы коммуникации в системах типа человек-машина (клавиатура, мышка), машина-машина (традиционные протоколы и форматы передачи и обмена информацией) утрачивают актуальность с каждой минутой.

В самом ближайшем будущем возникнет необходимость в совершенно новых подходах организации и обмена информацией. Эта необходимость продиктована, прежде всего, необходимостью сокращения расходов на разработку вышеописанных систем. В настоящее время подготовка обучающих наборов данных и машинное обучение чрезмерно трудозатратны и объективно превышают бюджеты. Другая проблема заключается в сложности повторного использования и развития уже готовых и натренированных систем. Другими словами, создав рабочую модель, способную распознавать дорожные знаки, разработчики вынуждены производить повторный процесс переобучения каждый раз при изменении или расширении среды, в которой данная система должна функционировать. В текущих реалиях, любая попытка выстроить новую систему распознавания на базе коммуникации с уже существующими программными комплексами потребует практически полного цикла проектирования и разработки.

Для решения всех выше изложенных проблем, в рамках реализации магистерской диссертации было произведено исследование современных возможностей вопросно-ответных систем и предложена модель вопросно-ответной системы, основанной на онтологическом подходе.

Вопросно-ответные системы (англ. question-answering systems, QA) являются основополагающими компонентами функционала всех виртуальных помощников. Наиболее частый сценарий применения QA систем состоит в следующем: пользователь системы в произвольной форме на естественном языке подаёт на вход вопрос (например, «что входит в состав класса вектор?») и получает ответ, содержащий набор определений и,

желательно, список веб-страниц (элементов аннотирования концептов онтологии), подтверждающих корректность ответа.

Согласно существующей классификации [23], можно выделить некоторые классы вопросно-ответных систем. Вопросно-ответные системы можно разграничить по двум принципам: первый – это тип ожидаемого ответа, а второй – используемые для поиска ответа методы.

Современные вопросно-ответные системы умеют работать со следующими группами вопросов:

- Фактологические, то есть такие, которые требуют ответ в виде некоторого установленного факта. У таких вопросов есть отличительная черта – они почти всегда формируются с использованием вопросительных слов (что, когда, кем, где и т.д.). К этому классу вопросов не относятся вопросы, требующие обоснования причинно-следственных связей и использующие конструкции, начинающиеся со слов «почему» и «зачем».
- Оценочные и сравнительные вопросы (например, «какой метод лучше применять при нахождении корней квадратного уравнения»).
- Бинарные вопросы, на которые возможны ровно два варианта ответа: либо «да», либо «нет» (например: «содержит ли онтология концепт аффинная система координат?»).
- Вопросы на понимание здравого смысла (например: «вейвлет – это функция, множество или геометрический объект?»).
- Экзаменационные вопросы. Тяжело вообразить такую ситуацию, когда пользователь голосового помощника обратился бы к системе с вопросами экзаменационного характера, однако, такое вполне возможно и для тестирования некоторых QA моделей действительно применяются вопросы, позаимствованные из экзаменационных тестов.
- Вопросы, заданные к данному тексту (иначе, вопросы машинного чтения, machine reading comprehension). Хотя постановка задачи MRC отличается от традиционной постановки задачи QA, эти задачи часто

изучают в совокупности. Задача MRC заключается в следующем: дан текст и несколько вопросов, возможно, с множественным выбором ответа, ответ на которые гарантировано есть в тексте. Модель MRC должна «прочитать» и «понять» текст, и ответить на заданные вопросы.

- Открытые вопросы (open domain QA, ODQA) – вопросы, ответ на которые надо найти в большом корпусе текстов на разные темы, например, в Википедии, некотором семантическом графе или узко-профильной онтологии.

Глава 4. Модель вопросно-ответной системы

В этой главе предложена модель вопросно-ответной системы в области математического знания, основанной на онтологическом подходе. Предлагаемая модель опирается на ресурсы онтологии профессионального математического знания OntoMathPro.

4.1. Архитектура и протекающие процессы предлагаемой модели

Для понимания архитектуры модели вопросно-ответной системы, то есть того, как она устроена, из каких частей состоит и по каким принципам работает, стоит сначала рассмотреть её высокоуровневую архитектуру.

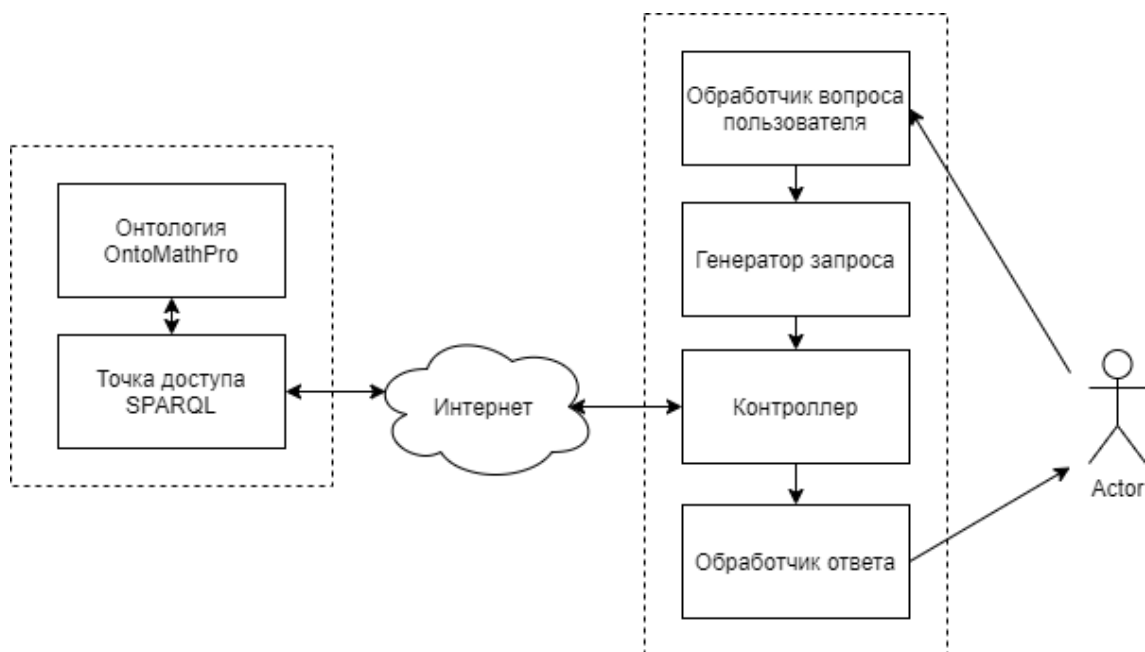


Рисунок 1. Высокоуровневое описание архитектуры (построено при помощи инструмента для создания диаграмм и блок-схем *Diagrams.net*, <https://app.diagrams.net>)

Как видно из рисунка 1, модель вопросно-ответной системы на основе онтологического подхода глобально состоит из двух основных частей:

- непосредственно онтологии, с развёрнутой на ней SPARQL точкой доступа;
- программного комплекса обработки вопросов и выдачи результатов на естественной языке.

В действительности, онтология может быть развёрнута локально на одном сервере с программой обработки и взаимодействия, но, так как одной из преследуемых целей было построить адаптивную гибкую модель вопросно-ответной системы, не имеющую искусственных ограничений, так называемых контрактов, ни на форматы вопросов, ни на местоположение развёрнутой онтологии, было принято решение заранее предусмотреть возможность подключения к любому RDF-графу в любом местоположении. Также это имеет ценность с той точки зрения, что действительно объёмные онтологии, такие как, например, WikiData или DBPedia, стоит разворачивать именно на выделенном сервере, чтобы не получать ощутимых потерь

производительности всего программного комплекса. Далее разработанная модель будет рассмотрена подробнее.

Прежде чем переходить к более подробному рассмотрению отдельных частей модели, не лишним будет посмотреть на порядок действий и протекающих процессов внутри модели, детальнее представленных на диаграмме последовательности (sequence diagram) (рисунок 2), выполненной в нотации UML.

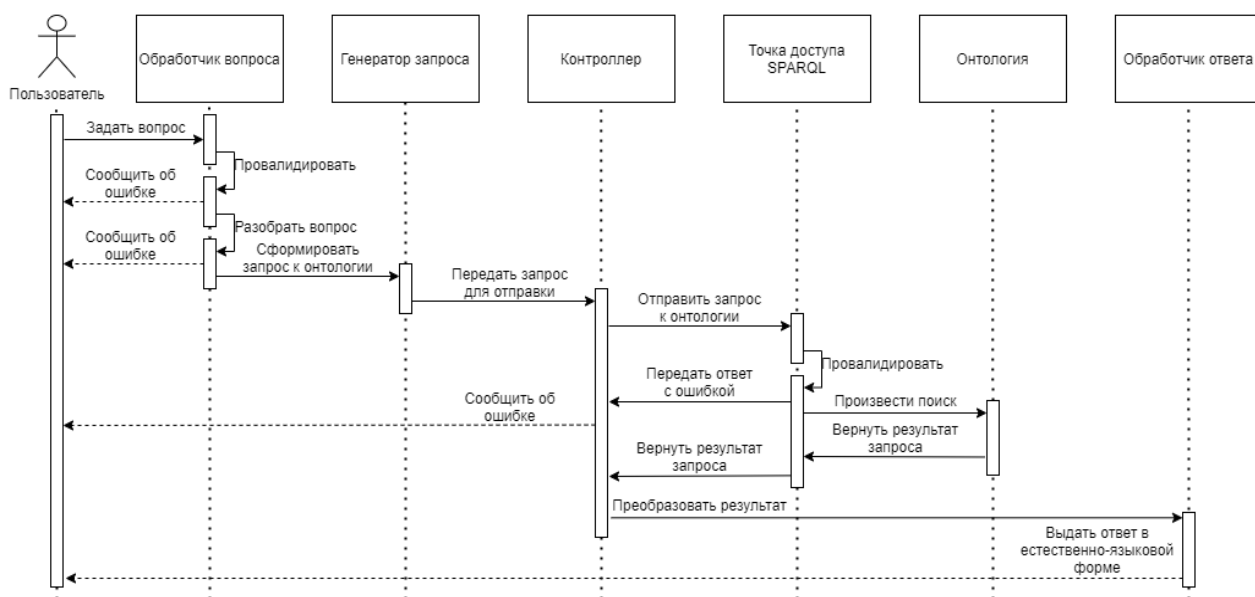


Рисунок 2. Диаграмма последовательности. Отображены порядок действий и протекающих процессов в модели вопросно-ответной системы, основанной на онтологическом подходе, предлагаемой в данной работе (*построено при помощи инструмента для создания диаграмм и блок-схем Diagrams.net, <https://app.diagrams.net>*)

Как видно из рисунка 2, процесс получения ответа на вопрос пользователя является комплексной задачей, состоящей из нескольких важных этапов:

1. Валидация вопросительного предложения на возможность его обработки системой.
2. Разбор вопроса пользователя по заложенным паттернам, извлечение из него всей необходимой информации (параметров будущего запроса, шаблона вопросительного предложения).

3. Формирование SPARQL запроса для поиска ответа на заданный вопрос.
4. Передача по каналу связи сформированного запроса к онтологии на развёрнутую SPARQL точку доступа.
5. Валидация запроса на возможность исполнения на RDF графе.
6. Исполнение запроса и поиск вариантов ответа.
7. Преобразование выданных онтологией результатов в качестве ответа в естественно-языковой форме.
8. Выдача результатов исполнения пользователю.

Подробнее обо всех аспектах работы реализованной модели вопросно-ответной системы будет сказано в последующих разделах.

Важно понимать, что то, как работает и хранит в себе данные RDF граф, и то, как с ними работают модули вопросно-ответной системы, не являются человеко-читаемым форматом представления данных. Далее рассматривается представление онтологии по модели RDF.

4.2. Представление данных онтологии по модели RDF

Онто-семантическим графом назовём ориентированный граф, вершинами которого являются концепты, выделенные в рамках предметной области, в совокупности с соответствующей им информацией, аннотированием из онтологий и графов знаний, а именованные дуги графа определяют название семантических отношений, связывающих эти концепты. Направление дуг онто-семантического графа определяет последовательность аргументов таких зависимостей.

Для представления баз знаний, представляемых как онто-семантические графы, применяется модель представления данных RDF.

Resource Description Framework (RDF, «среда описания ресурса» — это разработанная консорциумом Всемирной паутины модель для представления данных, в особенности — метаданных (смотри, например, [24, 25]). RDF представляет сведения о концептах в виде, пригодном для обработки программными средствами. RDF является частью концепции семантической паутины. Ресурсом в RDF может быть любая сущность — как информационная (например, изображение или веб-ресурс), так и неинформационная (например, теорема, человек или книга). Утверждение, высказываемое о ресурсе, имеет вид «субъект — предикат — объект» и называется триплетом (смотри подробнее [24]).

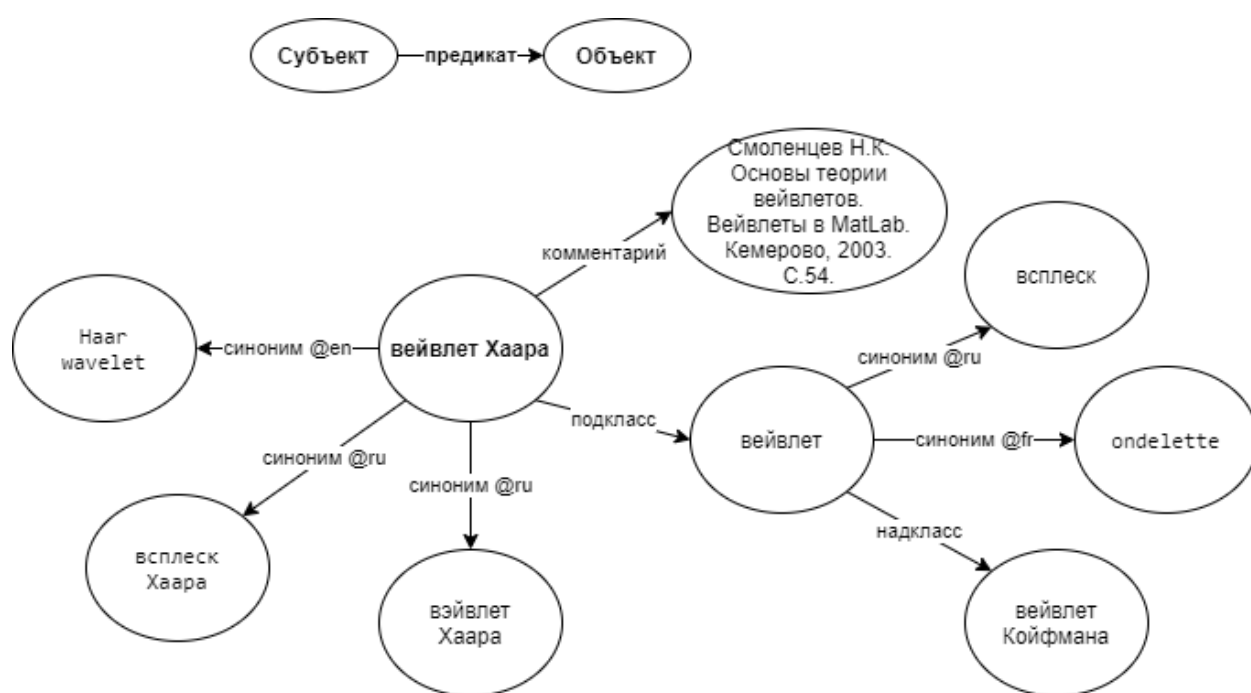


Рисунок 3. Фрагмент представления данных онтологии OntoMathPro по модели RDF. Выделен класс «вейвлет Хаара» и некоторые его связи с другими концептами (построено при помощи инструмента для создания диаграмм и блок-схем *Diagrams.net*, <https://app.diagrams.net>)

На рисунке 3 приводятся графическая интерпретация абстрактного триплета и конкретный пример из онтологии OntoMathPro, представленный в формате RDF.

Для работы с данными, хранимыми в данном формате, был разработан SPARQL – язык запросов к данным, представленным по модели RDF.

Язык запросов SPARQL это декларативный язык запросов (как SQL) для исполнения операций манипуляции и определения данных на информации, представленной коллекцией утверждений (триплетов), определённых по RDF.

Данный язык запросов достаточно похож на SQL, но имеет определённую специфику, в силу сферы применения. Запросы SPARQL строятся вокруг ранее упомянутых триплетов, что позволяет искать информацию как по типам связей (предикатам), так и по концептам и их свойствам.

Далее будет рассмотрена структура группы онтологий математического знания OntoMath, основываясь на которой разрабатывалась модель вопросно-ответной системы в рамках данной магистерской диссертации.

4.3. Структура онтологии, применяемой вопросно-ответной системой

Рассмотрим разрабатываемую математиками Казанского (Приволжского) федерального университета с 2014 года по сегодняшнее время группу онтологий математического знания OntoMath, в которую входят OntoMathEdu и OntoMathPro (смотри подробнее [8, 21, 26–28]).

OntoMathEdu – образовательная математическая онтология, предназначенная для:

1. семантической разметки математических учебников;

2. автоматической рекомендации образовательного контента в соответствии с профилем обучающегося;
3. автоматической генерации тестовых заданий.

Также эта онтология математического знания может быть использована в качестве основы для некоторой справочной базы знаний, ориентированной на конечного пользователя, что говорит нам о том, что её можно применять в качестве базы знаний в рамках разработки вопросно-ответных систем, основанных на онтологическом подходе.

OntoMathPro – онтология математического знания профессионального уровня, предоставляющая разумный компромисс между простыми словарями и высоко формализованными моделями, нацеленными на вычислимую проверку; является идейным продолжением OntoMathEdu.

Графы знаний OntoMath в основе имеют общий подход к архитектуре и организованы на трёх основных уровнях:

1. Основополагающий слой онтологии. В качестве базовой онтологии выбрана UFO-A (Объединенная фундаментальная онтология, используемая для анализа структурных концептуальных конструкций моделирования).
2. Слой онтологии предметной области, который содержит независимые от языка концепции математического знания.
3. Лингвистический слой, содержащий многоязычные лексиконы, обеспечивающие лингвистическое обоснование понятий из слоя онтологии предметной области.

Однако сама по себе онтология не является средством, которым может воспользоваться пользователь, как это было бы возможно, например, с физически осязаемым бумажным справочником. Как уже было сказано ранее, в данном и предыдущих разделах, онтология – это лишь некое очень большое хранилище данных, хранимых специфичным образом. Для того чтобы

получить доступ к данным, хранимым в онтологии, необходимо организовать точку доступа SPARQL.

4.4. Взаимодействие с RDF-графами

Семантическая сеть - это не только размещение данных в сети. Речь идёт о создании ссылок, чтобы человек или машина могли исследовать сеть данных. Со связанными данными, когда у Вас есть некоторые из них, можно найти ещё другие данные, связанные с ними.

Подобно сети гипертекста, сеть данных состоит из документов в сети. Однако, в отличие от сети гипертекста, где ссылки являются якорями отношений в гипертекстовых документах, написанных на HTML, для данных они связывают произвольные объекты, описываемые RDF. URI идентифицируют любой вид объекта или концепции.

Предоставление точек доступа SPARQL (англ. SPARQL-endpoint) является рекомендованной практикой при публикации данных во всемирной паутине (смотри, например, [29]). Доступ к базе обычно осуществляется через SPARQL-endpoint по протоколу HTTP.

Точка доступа SPARQL – это точка присутствия в сети HTTP, которая способна принимать и обрабатывать запросы протокола SPARQL. Она идентифицируется URL-адресом, который обычно называют URL-адресом конечной точки SPARQL. За этим адресом скрывается веб-сервис, обрабатывающий SPARQL запросы на онтологии и возвращающий результаты запроса, либо сообщения о вызванных ошибках.

Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

Query Text

```
select distinct ?Concept where {[ ] a ?Concept} LIMIT 100
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see [details](#).)

Results Format: HTML

Execution timeout: 0 milliseconds (values less than 1000 are ignored)

Options: ☒ Strict checking of void variables

(The result can only be sent back to browser, not saved on the server, see [details](#))

Run Query Reset

Рисунок 4. Web-интерфейс SPARQL точки доступа онтологии OntoMathPro

К точке доступа, как правило, прилагается web-интерфейс, предоставляющий возможность человеку взаимодействовать с онтологией посредством SPARQL запросов. Интерфейс точки доступа OntoMathPro можно увидеть на рисунке 5.

Теперь, когда часть модели, связанная с базой знаний рассмотрена, можно перейти к рассмотрению взаимодействия с пользователем. Начать стоит с рассмотрения типов вопросительных предложений.

4.5. Типы вопросительных предложений

В этом разделе выделены основные типы вопросительных предложений в создаваемой вопросно-ответной системе.

От того, по какому принципу будут разделяться между собой и анализироваться вопросительные предложения, зависит весь подход к построению вопросно-ответной системы.

В работе [3] был сделан упор на выделение семантического фокуса вопроса и, как результат, были получены 16 классов вопросов, используемых в системе.

Для разработанной в настоящей работе модели не применяется семантический анализ в чистом виде, потому используемый в [3] принцип разделения вопросительных предложений не подошёл.

Так как создаваемая в настоящем исследовании вопросно-ответная система использует другие технологии и не основана на сторонней поисковой системе, а сама является вопросно-ответной системой, то требовался более «низкоуровневый» принцип распределения вопросов пользователей, определяющий бизнес-процессы внутри вопросно-ответной системы. Формат и принцип обработки вопроса в случае вопросно-ответной системы, основанной на работе с онто-семантическими графами, зависит от одного существенного факта – наличия грамматически выделенного вопросительного слова или словосочетания, далее будем называть его параметром вопросительного предложения, или параметром вопроса. Можно выделить 2 основных типа вопросительных предложений:

- параметризованные вопросы;
- вопросы без параметра (будем называть их общие вопросы).

Под параметризованным вопросом будем понимать такое вопросительное предложение, формируемое пользователем вопросно-ответной системы, которое соответствует некоторому известному системе шаблону вопросительного предложения, подразумевающего наличие ключевого слова, от которого непосредственно будет зависеть ответ.

Примером параметризованного вопроса выступает: «Что фигурирует в литературе под авторством Колмогоров?». Данный вопрос соответствует шаблону «что фигурирует в литературе под авторством». Здесь «что фигурирует в литературе под авторством» является неизменной частью

шаблона вопросительного предложения, а строка «Колмогоров» включена в качестве параметра запроса к вопросно-ответной системе, относительно которого и будет происходить поиск по онтологии математического знания OntoMathPro.

В онтологии будет произведён поиск концептов, содержащих в элементах аннотирования, где расположены ссылки на веб-страницы и литературу, указанная фамилия. Результат такого запроса приведён в качестве примера номер 3 в разделе Примеры работы вопросно-ответной системы.

Общий вопрос – это такой пользовательский вопрос на естественном языке, который имеет один конкретный ответ. Этот ответ не зависит ни от одного слова или словосочетания в вопросе, а при изменении шаблона получается другой по смысловой нагрузке запрос.

В качестве примера вопроса без параметров рассмотрим «Сколько всего концептов в онтологии?». Как и в случае с параметризованными вопросами, у него есть синонимичные формы, например краткая форма вопроса – «Сколько триплетов?». Но в данном примере невозможно выделить слово, либо группу слов, заменив которую семантически вопрос останется без изменений, а ответ на такую его изменённую формулировку будет другим, но всё ещё в рамках того же смыслового значения и цели вопроса. Результат подачи такого вопроса на вход разработанного прототипа вопросно-ответной системы, можно увидеть в примере 1 раздела Примеры работы вопросно-ответной системы.

С тем, как группы с конкретными синтаксическими классами вопросов представлены в программной реализации предлагаемой модели, можно ознакомиться в разделе 5.4. Наполнение карты преобразований.

4.6. Алгоритм обработки вопроса пользователя

Ниже приведён разработанный пошаговый алгоритм обработки пользовательских вопросов, на котором основывается предлагаемая модель вопросно-ответной системы, базирующаяся на онтологическом подходе (блок-схема алгоритма представлена на рисунке 6):

Шаг 1. На вход подаётся анализируемая строка T , представляющая собой вопросительное предложение пользователя.

Шаг 2. Извлечение шаблона P вопросительного предложения из анализируемой строки.

Шаг 3. Извлечение содержимого карты преобразований M из локального хранилища.

Шаг 4. Поиск подходящих под шаблон P элементов карты $M(P)$. Поиск производится с помощью регулярных выражений с параметрами игнорирования регистра.

Шаг 5. Если $M(P)$ – пустое множество, то выдача пользователю предупреждения о неудачной попытке обработать введённый вопрос и досрочное завершение выполнения алгоритма.

Шаг 6. Из $M(P)$ производится отбор варианта с наибольшим соответствием $M(P)_b$ шаблону P . Отбор производится по следующим критериям в порядке убывания важности:

- наличие / отсутствие параметров вопросительного предложения пользователя в шаблоне P при ожидании / не ожидании параметров в соответствии с картой преобразований;
- одинаковое число ожидаемых и обнаруженных параметров вопросительного предложения;

- наибольшая длина совпадающих частей шаблона P и обнаруженного соответствия элемента карты преобразований $M(P)_i$.

Шаг 7. Выделение соответствующей формы ответного предложения A_b из пар шаблон вопроса – шаблон ответа выбранного по наилучшему соответствию элемента карты $M(P)_b$.

Шаг 8. Выделение из $M(P)_b$ соответствующего вопросительному предложению шаблона SPARQL запроса Q_b .

Шаг 9. Извлечение из конфигурации дополнительных параметров к формированию SPARQL запроса:

- параметр оператора LIMIT (ограничение на максимальное число результатов поиска по онтологии), необходимый для недопущения чрезмерно долгого исполнения запросов на стороне RDF графа;
- адрес точки доступа SPARQL онтологии, заранее развёрнутой на том же, либо удалённом сервере, на которую будет передан SPARQL запрос.

Шаг 9. Формирование запроса Q_b^f к онтологии, путём подстановки параметров вопроса и параметра оператора LIMIT в запрос Q_b .

Шаг 10. Открытие точки доступа к онтологии по извлечённому из конфигурации URI.

Шаг 11. Передача сформированного SPARQL запроса Q_b^f на точку доступа SPARQL по протоколу HTTP.

Шаг 12. Исполнение запроса на подключённой онтологии.

Шаг 13. Получение данных $S(Q_b^f)$ в виде набора (коллекции) результатов (строк таблицы ответа) запроса.

Шаг 14. Обработка набора $S(Q_b^f)$ перед передачей на формирование ответа в готовый, первично обработанный набор $S(Q_b^f)_r$:

- очистка набора от вариантов с пустыми результатами параметров вопросительного предложения;

- отбор среди элементов только подходящих по локализации, если требовалось;
- преобразование результатов, приложенных к параметрам вопросительных предложений, в форматированную человеко-читаемую строку.

Шаг 15. Подстановка оставшихся после обработки вариантов ответа $S(Q_b^f)_r$ в шаблон ответа A_b

Шаг 16. Выдача пользователю сформированного ответа на естественном языке.

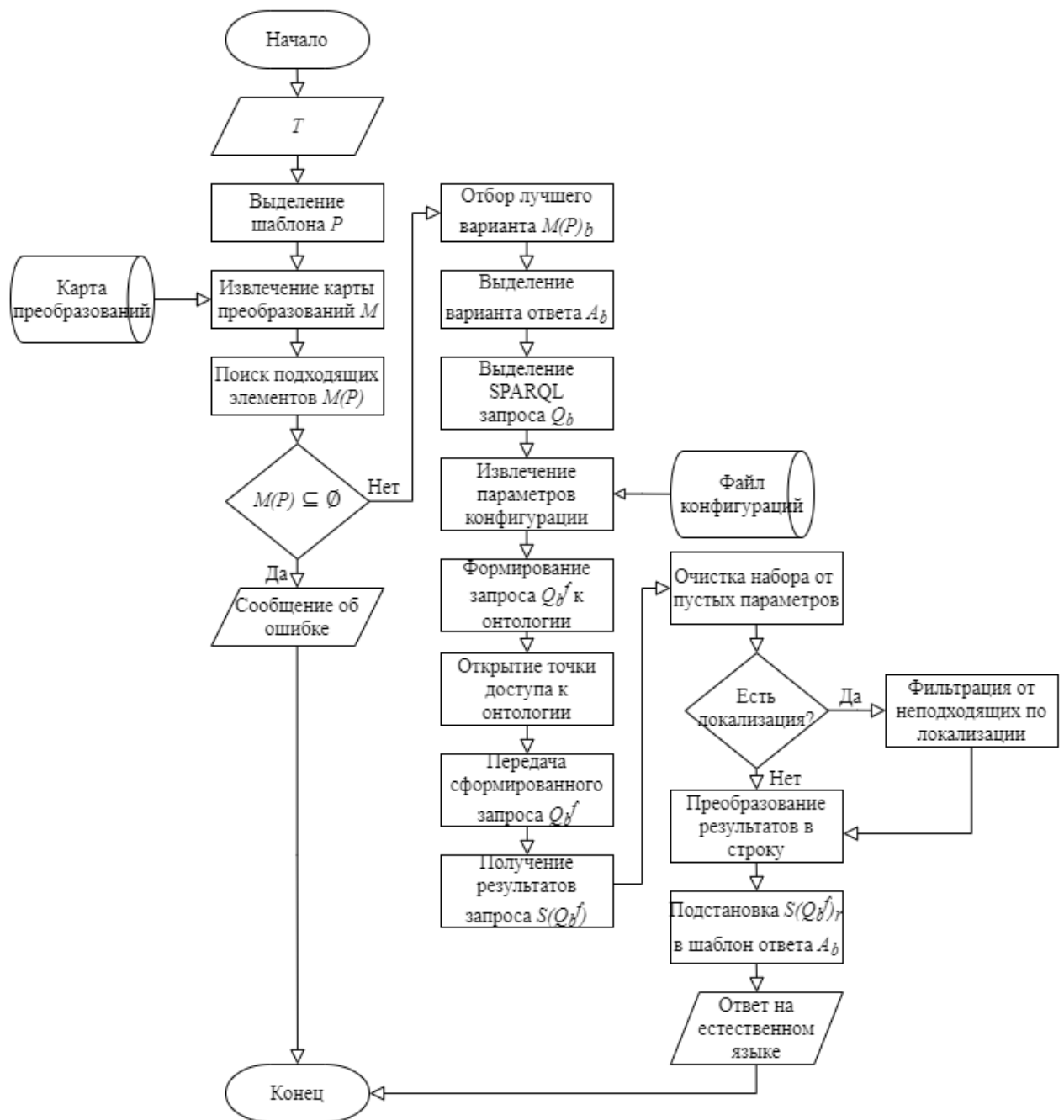


Рисунок 5. Блок-схема алгоритма обработки вопроса пользователя
(построено при помощи инструмента для создания диаграмм и блок-схем
Diagrams.net, <https://app.diagrams.net>)

В ходе работы был разработан и программно реализован алгоритм обработки пользовательских вопросов. Особенности программной реализации приведённого алгоритма и построенной на нём модели вопросно-ответной системы описаны в Главе 5 данной работы, посвящённой непосредственно реализации.

Глава 5. Реализация прототипа модели вопросно-ответной системы, основанной на онтологическом подходе

5.1. Программная реализация

В ходе диссертационного исследования был программно реализован прототип модели вопросно-ответной системы, основанной на онтологическом подходе. Данная система состоит из онтологии математического знания OntoMathPro, развернутой точки доступа SPARQL к ней и программного комплекса, реализующего алгоритм обработки пользовательских вопросов. Программа выполнена на языке программирования C# и ориентирована на русский язык. Однако

предусмотрено, что вопросы и результаты ответов могут поддерживаться на любых языках, поддерживаемых используемой онтологией.

5.2. Элементы адаптивности

5.2.1. Карта преобразований

Для того чтобы обеспечить взаимодействие пользователя с программным комплексом, и сокрытой внутри вопросно-ответной системы онтологией в частности, на естественном языке, необходимо проработать систему преобразования запросов пользователя во внутреннее представление системы, а также разработать способ обратного преобразования – из ответа системы в ответ на естественном языке.

Для решения поставленной задачи был разработан способ хранения данных, названный картой преобразований, и программный модуль вокруг него.

Так как целью работы является создание адаптивной модели вопросно-ответной системы, способ представления данных также должен приобрести свою специфику. Для того чтобы сохранялась возможность дополнения вариантов вопросов и ответов к системе и изменения существующих элементов карты преобразований, был разработан модуль работы с хранимыми данными.

Карта преобразований представляет собой текстовый файл, содержащий набор элементов карты, пример одного из них можно увидеть на рисунке 7.

```

true
где фигурирует (.*) ; что фигурирует в литературе под авторством (.* )
были обнаружены следующие косвенно связанные сущности: {0} ; обнаружено следующее: {0}
SELECT DISTINCT ?label ?comment
WHERE
{{
  ?entity rdfs:comment ?comment.
  ?entity rdfs:label ?label.
  FILTER CONTAINS(LCASE(STR(?comment)), LCASE("{0}"))
  FILTER langMatches(lang(?label),"ru")
}}
+

```

Рисунок 6. Элемент карты преобразований (полное содержимое приведено в Приложении 1)

Элементы карты имеют следующую структуру:

1. Флаг наличия параметров в вопросе пользователя. Если установлено значение true, значит необходимо использовать модуль обработки вопросов с параметром, если false – модуль обработки общих вопросов.
2. Вариации (синонимичные формы) вопроса, решаемого SPARQL запросом данного элемента карты преобразований. Здесь через точку с запятой указываются все формы вопроса, который может быть решён данным преобразованием.
3. Соответствующие формулировкам вопроса варианты ответа – разделённые точкой с запятой формы ответа на естественном языке в количестве, равном количеству синонимичных форм вопроса, с указанием комбинацией символов “{0}” места, куда будет подставлена информация, полученная из используемого графа знаний.
4. Соответствующий шаблон SPARQL запроса.
5. Признак окончания очередного элемента карты преобразований.

Описанное представление данных плохо подходит для эффективного применения в рамках высоконагруженных информационных систем, работающих в реальном времени, поэтому применяется, как было сказано в начале данного раздела, исключительно с целью предоставления удобного и

простого способа расширения системы профильными для предметной области, но не специалистами в рамках разработки программных систем, сотрудниками используемой онтологии.

При первом запуске вопросно-ответной системы, подключённый файл с картой преобразований парсится, проверяется на корректность, и, если всё в порядке, сохраняется в формате JSON.

При дальнейшем развитии и расширении системы будет иметь смысл заменить техническое хранение данных в JSON на сохранение их в базу данных. Хорошим решением может послужить документоориентированная СУБД MongoDB, на которую очень хорошо накладывается структура хранимых данных. Либо, как другой вариант, при наличии большого объёма оперативной памяти на сервере, можно прибегнуть к использованию in-memory базы, такой как Redis, либо же использовать их комбинацию, а именно, MongoDB применить в качестве основного хранилища элементов карты преобразований, а Redis применять в качестве кеша над базой данных.

Также предусмотрена возможность обновления содержимого карты при наличии уже развёрнутой на сервере вопросно-ответной системы. Для этого необходимо обновить содержимое первого файла и инициировать Get запрос на контроллер сервиса по URI «/api/Questions». После этого, как и в случае с первым запросом к системе, произойдёт валидация и разбор содержимого карты преобразований и преобразование её в более удобный для использования программным комплексом JSON формат. Ниже (рисунок 8) представлен элемент карты преобразований, изображённый на рисунке 7, после предобработки и сохранения в новом формате.

```
[...,{ "HasParams":true, "VariantAnswerPairs":{"где фигурирует
(.*)" : "были обнаружены следующие косвенно связанные сущности:
{0}", "что фигурирует в литературе под авторством
(.*)" : "обнаружено следующее: {0}"}, "QueryTemplate":"SELECT
DISTINCT ?label ?comment WHERE {{ ?entity rdfs:comment ?comment.
?entity rdfs:label ?label. FILTER CONTAINS(LCASE(STR(?comment)),
LCASE(\"{0}\")) FILTER langMatches(lang(?label),\"ru\") }} "},...]
```

Рисунок 8. Фрагмент карты преобразований после преобразования в JSON

В процессе работы системы, необходимые элементы карты будут десериализованы и кешированы на время исполнения запроса.

5.2.2. Конфигурируемость

Приложения ASP.NET Core поддерживают настройку посредством поставщиков конфигурации. Существуют различные поставщики конфигурации, но все они работают с данными в парах «ключ-значение» (смотри подробнее [30]). В качестве поставщика в реализации, предлагаемой в данной работе, применяется поставщик конфигурации файла, работающий с файлом параметров appsettings.json.

Удобство такого подхода заключается в трёх пунктах:

- простота и доступность;
- не требуется пересборка (rebuild) решения;
- не требуется перезапуск серверной части.

В файл конфигураций вынесены две группы параметров. Первая – это системные параметры, не влияющие на ход исполнения алгоритма обработки вопросов пользователей вопросно-ответной системы. Сюда входят:

- настройка уровней логирования Serilog;
- настройка CORS-политики.

Вторая группа параметров, решающих конкретно вопрос адаптивности приведённого решения, включает настройки взаимодействия с онтологией:

- адрес SPARQL точки доступа используемой в данный момент онтологии (по умолчанию используется точка доступа онтологии OntoMathPro (<http://lobachevskii-dml.ru:8890/sparql>));
- ограничение на число элементов выборки (по умолчанию 100).

Вынесение адреса точки доступа запросов SPARQL позволяет подменять источник данных, с которым работает вопросно-ответная система, прямо во время исполнения (run-time), без необходимости в остановке и пересборке решения.

Ограничение на максимальное число элементов в результате запроса (результатирующей выборке) необходимо для поддержания адекватной скорости ответа вопросно-ответной системы. При работе с базами знаний, значительно превышающими по числу концептов предметную онтологию OntoMathPro (например, онтология общих знаний WikiData, на момент написания данной магистерской диссертации, содержит 93935225 концептов, при 880569 концептах у OntoMathPro – более чем в 100 раз больше узлов на RDF графе!), запрос будет исполняться неоправданно долго (вплоть до десятков минут), либо и вовсе будет прерван сервисом онтологии, при использовании ею параметров ограничений на максимальную длительность исполнения запроса.

5.3. Тестирование

Тестирование программного продукта является неотъемлемой частью разработки программного продукта. Без полноценно проведенного тестирования невозможно оценить, насколько качественно программный продукт выполнен. Чем качественнее будут проведены тесты, чем полнее будут разработанные тестовые сценарии, тем меньше вероятность, что будут упущены ошибки, которые в будущем пришлось бы исправлять.

Так как изначальные требования к финальному представлению разрабатываемого программного комплекса были плохо формализованы и работа заключается скорее в разработке нового решения и итеративном прототипировании задумки, нежели просто реализации программного обеспечения по строгому ТЗ, остро встала необходимость грамотной организации рабочего процесса.

Исходя из сложившейся ситуации был выбран подход разработки через тестирование, как наиболее подходящий способ организации процесса разработки.

TDD, test-driven development или разработка через тестирование — это методология разработки программного обеспечения, которая основывается на повторении коротких циклов разработки: изначально пишется тест, покрывающий желаемое изменение, затем пишется программный код, который реализует желаемое поведение системы и позволит пройти написанный тест, а затем проводится рефакторинг написанного кода с постоянной проверкой прохождения всех тестов [31].

На рисунке 9 можно увидеть графическое представление цикла разработки по TDD.

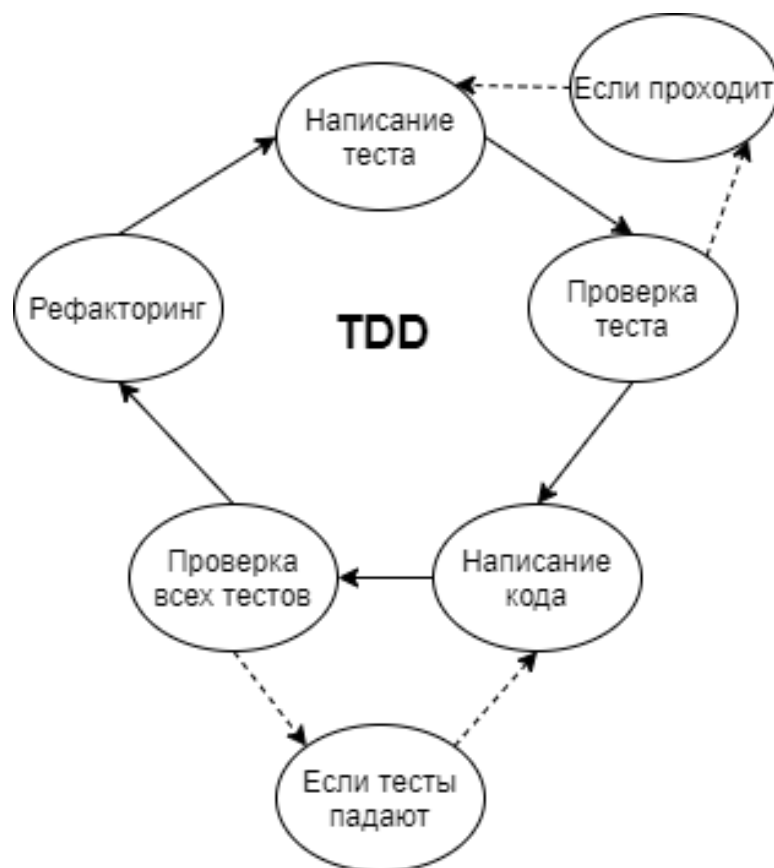


Рисунок 9. Цикл разработки по TDD (построено при помощи инструмента для создания диаграмм и блок-схем *Diagrams.net*, <https://app.diagrams.net>)

Ниже представлен исходный код одного из модульных тестов для класса `DataOperationState`, проверяющий метод `Processing` на верное исполнение (при `Success`, равном `true`, метод должен выполняться успешно и по завершении увеличить свойство `Code` на указанное значение).

```

[TestMethod]
public void ProcessingTest()
{
    var generator = new DataOperationState
    {
        Success = true,
        Code = 0
    };
    var inc = 150;
    dop.Processing(s => { }, inc);
    Assert.IsTrue(
        dop.Code != 0,
        "Неправильно инкрементирует коды ошибок");
}

```

Рисунок 10. Один из модульных тестов класса `DataOperationState`

5.4. Наполнение карты преобразований

В рамках выполнения данной магистерской диссертации было решено заложить в прототип модели вопросно-ответной системы как возможность отвечать на вопросы пользователя, так и посредством естественно-языкового запроса иметь возможность получать сведения о самой используемой онтологии.

Например, в качестве запроса сведений по используемой онтологии был реализован элемент карты преобразований, основанный на интересе к числу вершин представленного онто-семантического графа. Данный вопрос реализован в виде двух синонимичных форм шаблона вопросительного предложения:

1. «сколько всего концептов в онтологии»;

2. «сколько триплетов»;

и соответствующих им форм шаблона ответа на естественном языке:

1. «всего в используемой онтологии содержится $\{0\}$ концептов»;

2. «в онтологии содержится $\{0\}$ триплетов».

Синонимичные формулировки вопросительных предложений рассматриваются с точки зрения термина синоним в семантическом анализе.

Синонимы - слова, звучание и написание которых различно, но при этом у них похожее значение (например, полином-многочлен). Чаще всего они принадлежат к одной и той же части речи (смотри, например, [4]).

Каким образом будет выглядеть SPARQL запрос, решающий данную задачу, видно на рисунке 11.

```

SELECT DISTINCT (COUNT(?s) AS ?triples)
WHERE
{
    ?s ?p ?o
}

```

Рисунок 11. SPARQL запрос, решающий задачу определения числа концептов в используемой онтологии (полная карта преобразований приведена в Приложении 1)

Запрос основан на следующей идее: изначально необходимо получить все существующие триплеты `?s ?p ?o` в онто-семантическом графе, далее, с помощью оператора `DISTINCT`, необходимо избавиться от повторяющихся по субъектам триплетов и, с помощью функции `COUNT`, сосчитать их количество.

Также разработанная модель вопросно-ответной системы способна обрабатывать вопросы поиска гипонимов, гиперонимов, синонимов к указываемым терминам в срезе зависимости определений друг от друга по конкретным предикатам, подразумевающим данную семантическую связь определений, производить поиск по элементам аннотирования с целью выделения источников литературы по имени автора и другое.

В силу гибкости предлагаемой модели, группы семантических вопросов могут расширяться и далее по необходимости при развитии вопросно-ответной системы и используемых с ней онтологий.

В качестве примера вопроса, семантически обозначающего интерес пользователя к получению известных гипонимов относительно заданного термина, можно рассмотреть поиск дочерних классов, в терминологии RDF представленного предикатом `rdfs:subClassOf`. Заложенные в систему варианты такого пользовательского вопроса:

1. «дочерние элементы класса (.*)»;
2. «что входит в состав класса (.*)»;
3. «зависимые определения (.*)».

Соответствующие шаблоны ответных предложений в естественно-языковой форме:

4. «дочерние элементы: {0}»;
5. «в состав класса входит: {0}»;
6. «обнаруженные зависимые определения: {0}».

ГИПОНИМ – понятие, выражающее частную сущность по отношению к другому, более общему понятию (смотри, например, [4]).

Решаются формы данного вопросительного предложения посредством много более сложного SPARQL запроса (смотри рисунок 12).

```
SELECT DISTINCT ?name ?comment
WHERE {{
    ?target rdfs:label ?name.
    ?target rdfs:comment ?comment.
    ?target rdfs:subClassOf ?parent.
    ?parent rdfs:label ?label.
    FILTER(LCASE(STR(?label)) = LCASE("{0}"))
    FILTER langMatches(lang(?name),"ru")
}}
```

Рисунок 12. SPARQL запрос, решающий задачу поиска гипонимов к заданному параметром ?label термину (полная карта преобразований приведена в Приложении 1)

В данном случае для решения поставленной задачи необходимо произвести комплексную группировку триплетов относительно целевого параметра по нескольким различным предикатам. Для этого необходимо отобрать взаимосвязанные узлы графа по следующим принципам:

1. в выборку должны попасть объекты наименований ?name, связанные с целевым термином через предикат rdfs:label;
2. также в выборку необходимо поместить элемент аннотирования ?comment, определяемый предикатом онтологии rdfs:comment;
3. поиск целевого концепта в онтологии будет производиться по его наименованию ?label и, следовательно, поиск подходящего концепта будет

производиться по предикату `rdfs:label`, аналогично поиску имени дочерних концептов (смотри первый пункт);

4. и самое главное – связь дочернего и родительского концептов по предикату `rdfs:subClassOf`, связывающая целевой концепт онтологии с искомыми дочерними.

Также в данном запросе фигурируют две фильтрации:

- первая обеспечивает независимую от регистра входного параметра вопросительного предложения формулировку запроса;
- вторая – отбор среди синонимов наименований форму с русской локализацией.

В целом, вторую фильтрацию можно сделать гибче, выделив желаемые для результатов поиска локализации либо как ещё один параметр вопросительного предложения, либо как элемент конфигурации вопросно-ответной системы. Однако так как изначально ставилась задача работы с вопросами на русском языке, и подразумевались ответы в естественно-языковой форме также на русском языке, данная возможность доработки хоть и предусмотрена, но осталась для дальнейшего развития системы.

Полное содержимое сформированной карты преобразований можно увидеть в Приложении 1.

5.5. Примеры работы вопросно-ответной системы

Как было сказано в разделе 5.2.2. Конфигурируемость, разработанный прототип модели вопросно-ответной системы имеет 2 метода API (смотри рисунок 13):

1. Get запрос на обновление карты преобразований.
2. Post запрос для подачи вопросительного предложения пользователя на вход системы.

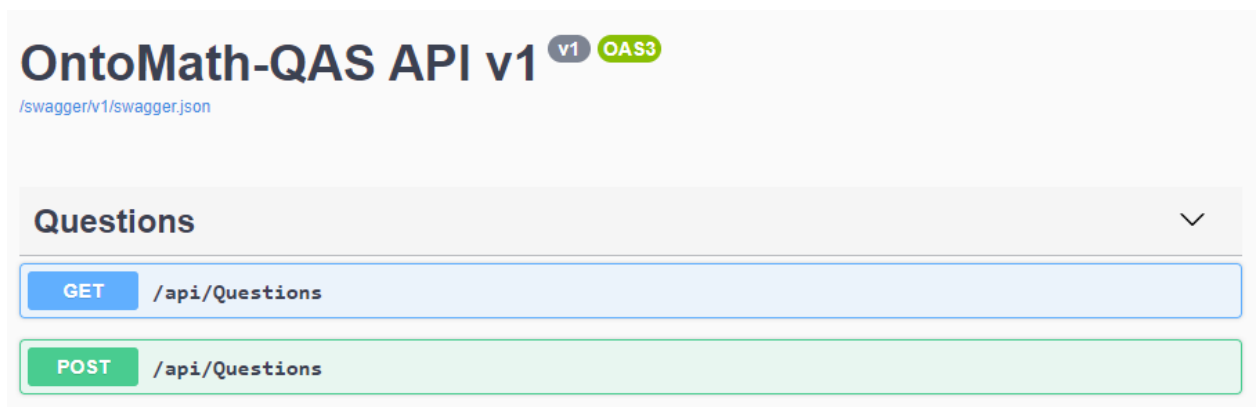


Рисунок 13. Web-интерфейс, предоставляющий доступ к API разработанного прототипа вопросно-ответной системы

В результате инициации Get запроса обновления карты преобразований система вернёт либо код 200 с пустым телом ответа, в случае успеха её обновления (рисунок 14), либо, в случае неудачи, выдаст сообщение об ошибке (рисунок 15).

Ошибка при обновлении карты преобразований может возникнуть в следующих случаях:

- отсутствие файла с картой преобразований;
- нарушение структуры файла ReadableMap.txt;
- несоответствие числа синонимичных форм вопросительного предложения числу шаблонов ответа;
- сетевые ошибки доступа к серверу.

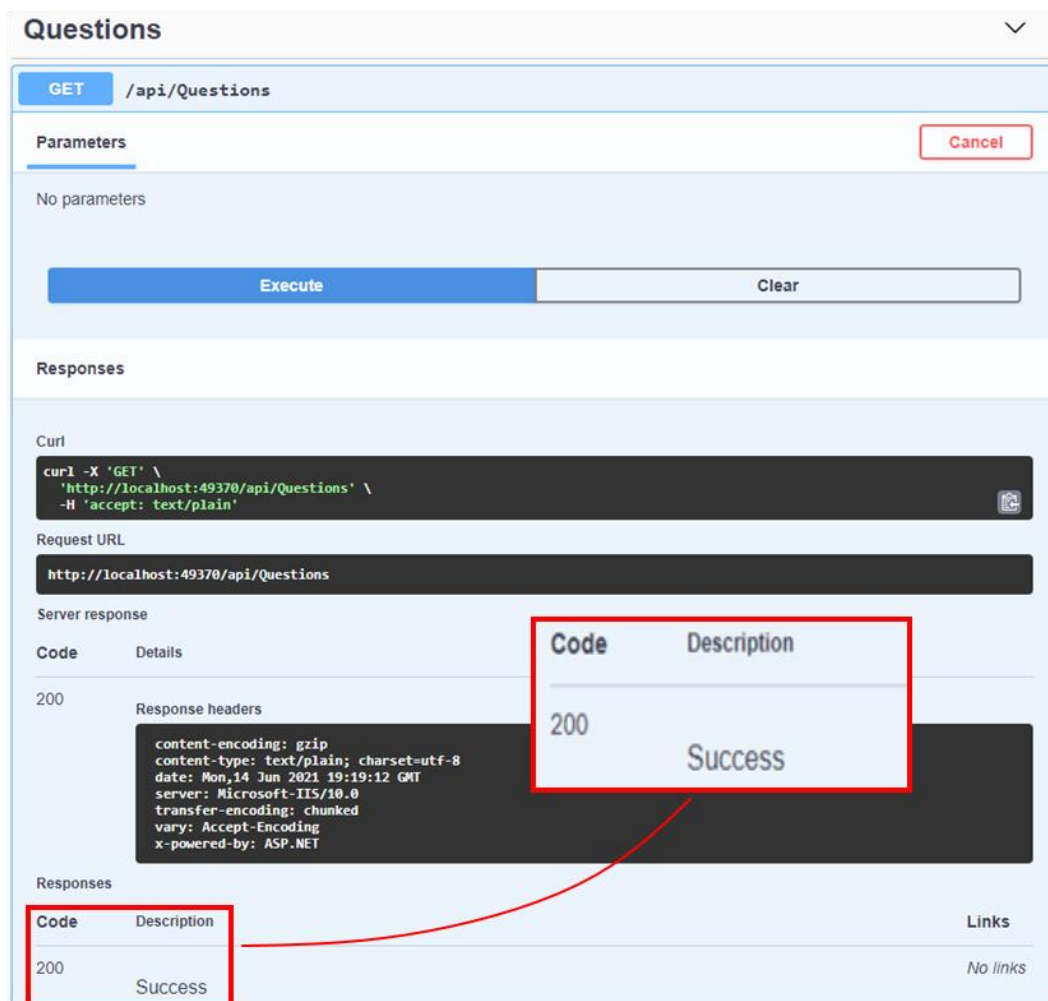


Рисунок 14. Успешное обновление карты преобразований

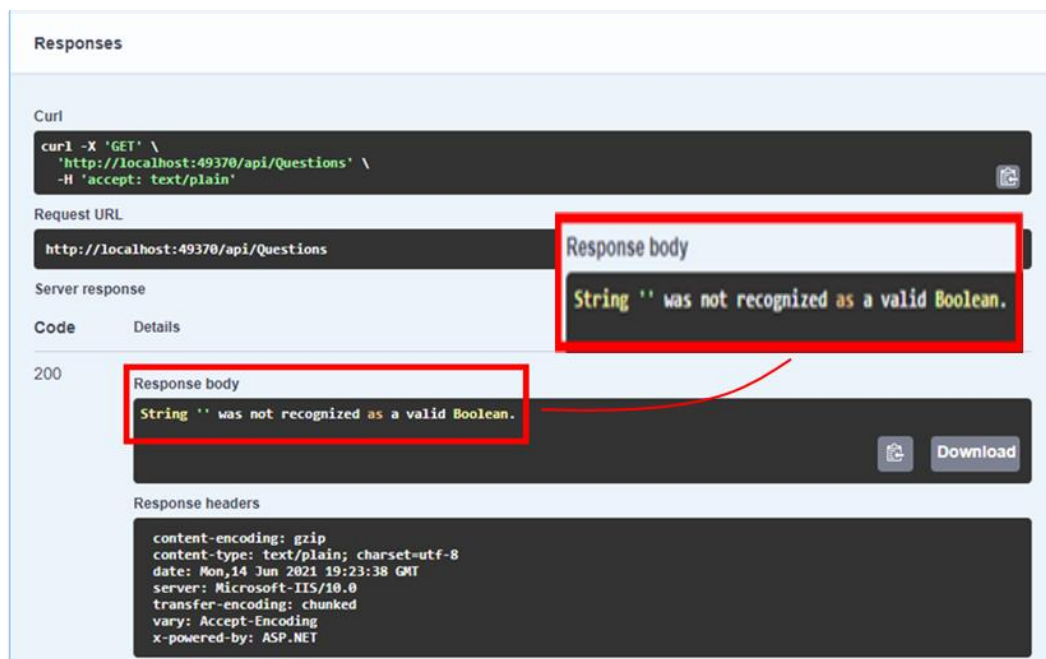


Рисунок 75. Ошибка при обновлении карты преобразований в случае отсутствия первого параметра одного из элементов карты преобразований

Далее рассмотрим работу вопросно-ответной системы именно на нескольких конкретных примерах вопросов пользователя, отличающихся по типу вопросительного предложения.

Пример 1. Рассмотрим общий вопрос, подразумевающий единственный однозначный ответ: «Сколько всего концептов в онтологии?».

В качестве результата обработки данного пользовательского вопроса ожидается предложение на естественном языке, содержащее число – количество концептов в онтологии разработанной вопросно-ответной системы.

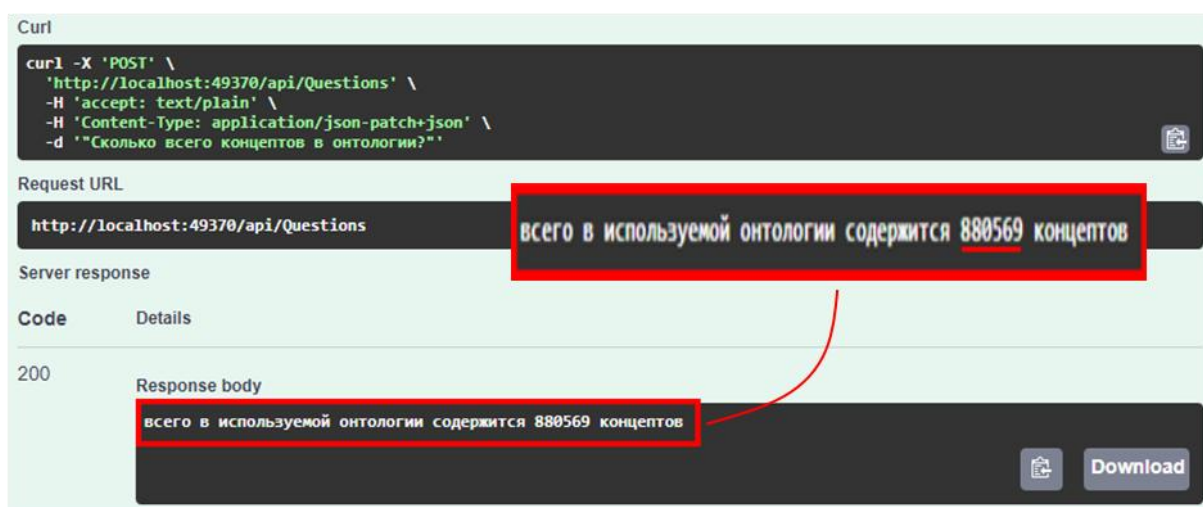


Рисунок 16. Ответ системы на вопрос о количестве концептов в онтологии

И действительно, система выдаёт ответ «всего в используемой онтологии содержится 880569 концептов», а как внимательный читатель мог помнить из раздела 5.2.2. Конфигурируемость, именно столько узлов на текущий момент хранится в онто-семантическом графе OntoMathPro.

Также разработанная модель вопросно-ответной системы способна обрабатывать вопросы поиска гипонимов, гиперонимов и синонимов к указываемым терминам, производить поиск по элементам аннотирования с целью выделения источников литературы по имени автора и многое другое. Для подтверждения, рассмотрим параметризованный запрос поиска гипонимов к определению Вейвлет.

Пример 2. Параметризованный вопрос поиска гипонимов термина Вейвлет: «Подскажи-ка, что входит в состав класса Вейвлет».

Параметром вопросительного предложения в данном случае будет являться подстрока «Вейвлет», от которого и зависит результирующая выборка.

Наиболее подходящим синонимом среди шаблонов, покрывающих данную формулировку вопроса, будет «что входит в состав класса».

В качестве ответа на вопрос ожидается увидеть «в состав класса входит:» и перечисление наименований связанных отношением `rdfs:subClassOf` концептов онтологии с соответствующими им элементами аннотирования `rdfs:comment` для возможного более подробного ознакомления.

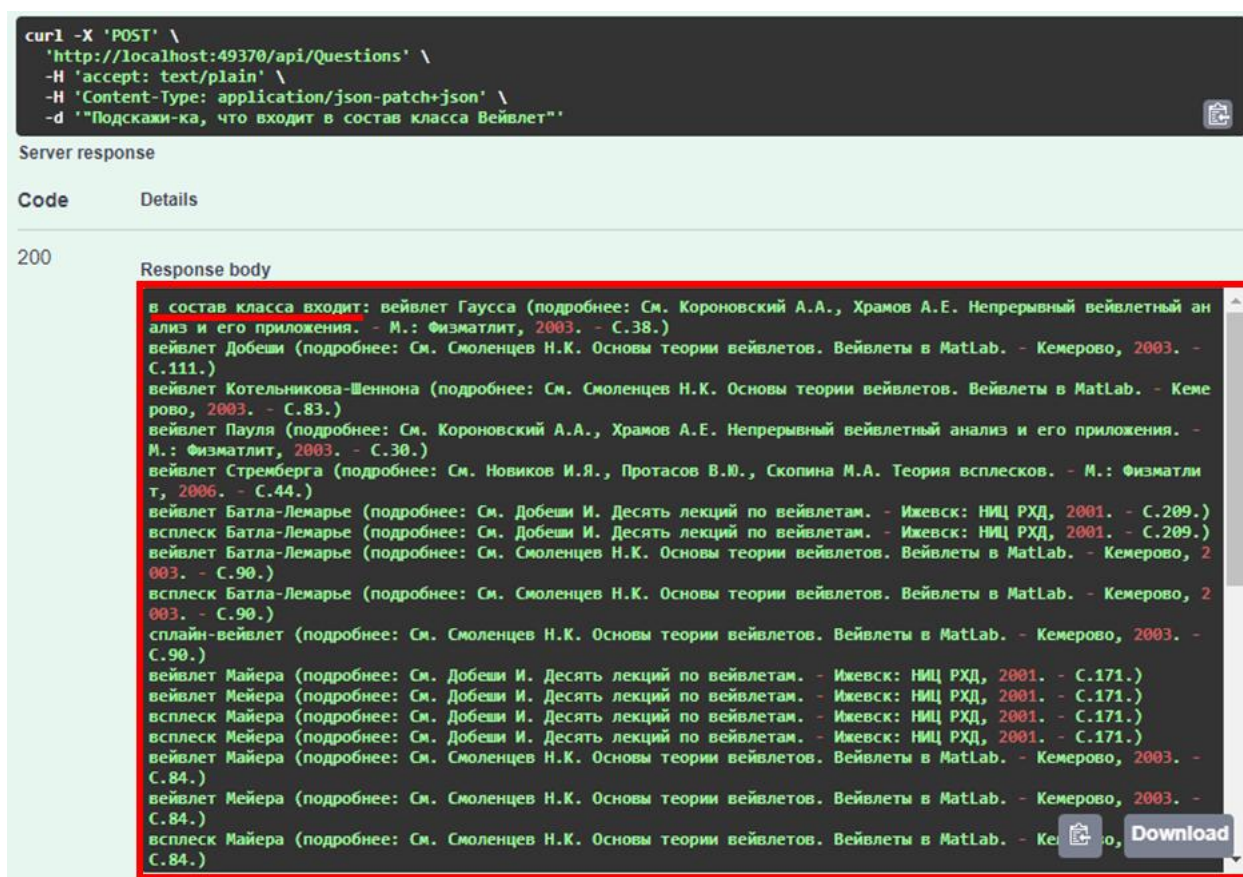


Рисунок 17. Ответ системы на вопрос о гипонимах определения Вейвлет

Как видно из рисунка 17, именно это мы и получим в качестве ответа.

Пример 3. Рассмотрим в качестве ещё одного примера вопросительное предложение «Что фигурирует в литературе под авторством Колмогоров», использованное как пример параметризованного вопроса в разделе 4.5. Типы вопросительных предложений.

Данный вопрос соответствует шаблону «что фигурирует в литературе под авторством (.*)». Здесь «что фигурирует в литературе под авторством» является неизменной частью шаблона вопросительного предложения, а строка «Колмогоров» включена в качестве параметра запроса к вопросно-ответной системе, относительно которого и будет происходить поиск по онтологии математического знания OntoMathPro.

В онтологии будет произведён поиск концептов, содержащих в элементах аннотирования, в которых помещены ссылки на веб-страницы и литературу, указанная фамилия. Результат такого запроса приведён на рисунке 18.

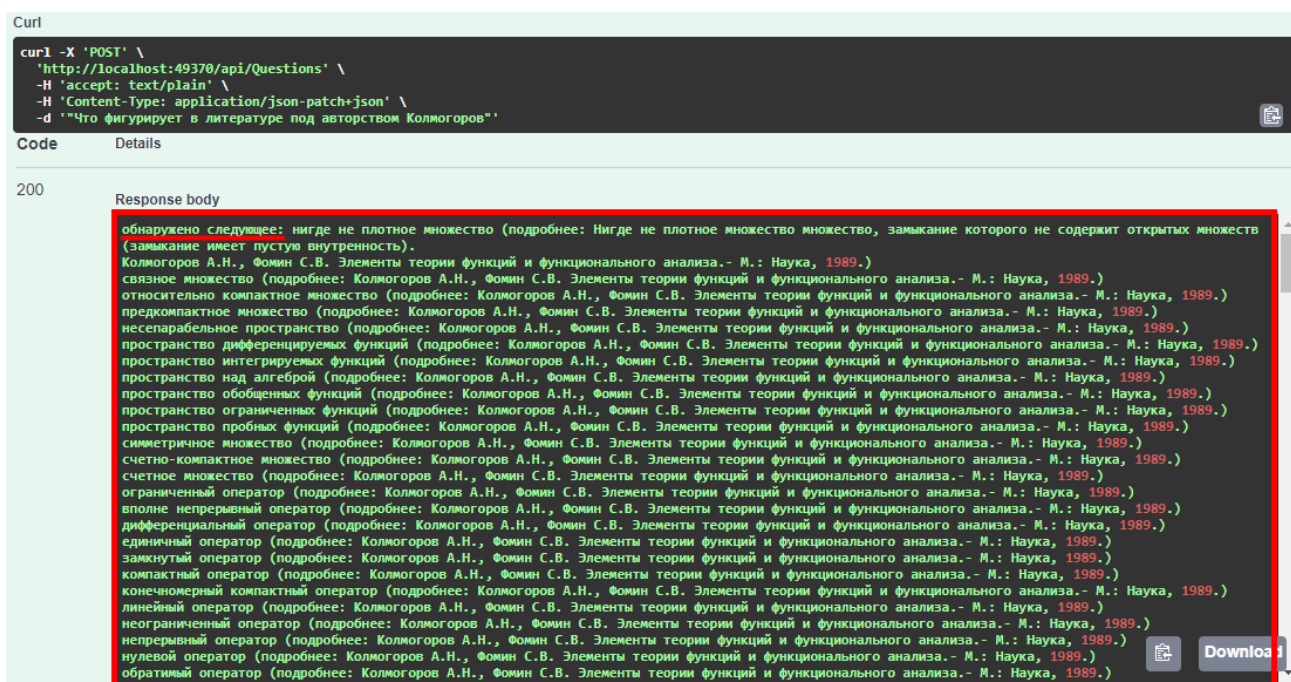


Рисунок 18. Поиск концептов в онтологии OntoMathPro по фамилии автора литературы, в которой фигурируют эти термины

В качестве ответа системы получено: «обнаружено следующее:» и разделённые переходом на новую строку концепты, содержащиеся в онтологии, в элементах аннотации к которым фигурирует фамилия Колмогоров, с указанием этих аннотаций после «(подробнее: ».

Заключение

По результатам выполнения магистерской диссертации была предложена модель вопросно-ответной системы, основанной на онтологическом подходе, и разработан прототип, реализующий её и отработанный с онтологией профессионального математического знания OntoMathPro.

Основными результатами работы являются:

1. Классификация типов вопросительных предложений, используемых предлагаемой моделью.
2. Алгоритм обработки вопросительных предложений пользователя вопросно-ответной системы.
3. Разработанная система преобразования вопросов пользователей в SPARQL запрос и результатов запроса к онтологии в соответствующий естественно-языковой ответ (карта преобразований).
4. Построенная в соответствии с указанным в предыдущих пунктах модель вопросно-ответной системы, основанной на онтологическом подходе.
5. Карта преобразований вопросов пользователей с заготовленными SPARQL запросами для различных вопросительных предложений.
6. Реализованный прототип предложенной в работе модели вопросно-ответной системы, работающий с онтологией математического знания OntoMathPro.
7. Содействие в развитии разрабатываемой в Казанском (Поволжском) федеральном университете группы онтологий математического знания OntoMath.

Предлагаемая модель вопросно-ответной системы позволит вывести онтологию профессионального математического знания OntoMathPro из состояния, когда ей может воспользоваться только технически подкованный

специалист, на уровень много более простого доступа пользователей к концептам, которыми она располагает. Также, в силу предлагаемого в работе гибкого подхода к настройке вопросно-ответной системы, её возможно станет применять с другими RDF графами в случае возникновения такой необходимости.

Материалы и коды, использованные в магистерской работе, выложены на GitHub – <https://github.com/EgoPingvina/Question-answering-system>.

Список использованных источников

1. **Hilbert, M.** How to Measure «How Much Information»? Theoretical, Methodological, and Statistical Challenges for the Social Sciences / M. Hilbert // International Journal of Communication. – 2012. – № 6. – P. 1042-1055.
2. **Lyman, P.** How much information / P. Lyman, H. R. Varian // University of California. – 2003.
3. **Мочалова, А. В.** Семантический анализатор русскоязычного текста для вопросно-ответной системы: диссертация ... канд. техн. наук: 05.13.18 / Мочалова Анастасия Викторовна. – Петрозаводск, 2017. – 128 с.
4. Первичный семантический анализ: сайт – 2018. – URL: <http://www.aot.ru/docs/seman.html> (дата обращения: 21.02.2021).
5. **Damljanovic, D.** Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction / D. Damjanovic, M. Agatonovic, H. Cunningham // Extended Semantic Web Conference. – Springer, Berlin, Heidelberg, 2010. – P. 106-120.
6. Стемматизация и генерация словоформ в казахском языке для систем автоматической обработки текстов / В. Б. Барахнин и др. // Вычислительные технологии. – 2017. – Т. 22. – № 4.
7. Combining semantic information in question answering systems / P. Moreda et al // Information Processing & Management. – 2011. – V. 47. – № 6. – P. 870-885.
8. Образовательная математическая онтология OntoMathEdu: структура и отношения / Л. Р. Шакирова и др. // Научный сервис в сети Интернет: труды XXI Всероссийской научной конференции (23–28 сентября 2019 г., г. Новороссийск). М.: ИПМ им. МВ Келдыша. – 2019. – С. 653-661.

9. **Guo, Q.** Question answering based on pervasive agent ontology and Semantic Web / Q. Guo, M. Zhang // Knowledge-Based Systems. – 2009. – V. 22. – № 6. – P. 443-448.
10. **Jung, H.** Automated conversion from natural language query to SPARQL query / H. Jung, W. Kim // Journal of Intelligent Information Systems. – 2020. – P. 1-20.
11. **Sarrouti, M.** SemBioNLQA: A semantic biomedical question answering system for retrieving exact and ideal answers to natural language questions / M. Sarrouti, S. O. El Alaoui // Artificial Intelligence in Medicine. – 2020. – V. 102. – P. 101767.
12. **Яхьева, Г. Э.** Вопросно-ответная система для управления информационными рисками на основе теоретико-модельной формализации предметных областей / Г. Э. Яхьева и др. // Информационные технологии. – 2017. – Т. 23. – № 2. – С. 97-106.
13. **Мальковский, М. Г.** Прикладное программное обеспечение: системы автоматической обработки текстов / М. Г. Мальковский, Т. Ю. Грацианова, И. Н. Полякова. // Издательский отдел факультета вычислительной математики и кибернетики МГУ им. М.В. Ломоносова. – Москва, 2000. – 7 с.
14. **Онтологии и тезаурусы: модели, инструменты, приложения: учебное пособие / Б.В. Добров, В.В. Иванов, Н.В. Лукашевич, В.Д. Соловьев.** – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2009. – 173 с.: ил. – (Серия «Основы информационных технологий»).
15. **Андреичев, М. Д.** Разработка программного комплекса генерации вопросов по заданным субъектам при помощи семантической сети / М. Д. Андреичев, А. М. Бирюков // Молодёжная школа-конференция «Будущее математического образования». – 2021. – С. 8-11.

16. **Vargas-Vera, M.** AQUA—ontology-based question answering system / M. Vargas-Vera, E. Motta // Mexican International Conference on Artificial Intelligence. – Springer, Berlin, Heidelberg, 2004. – P. 468-477.
17. **Филонов, Д. Р.** Вопросно-ответная система для поддержки абитуриентов с использованием современных мессенджеров / Д. Р. Филонов и др. // Моделирование и анализ информационных систем. – 2018. – Т. 25. – № 4. – С. 411-420.
18. Наукометрия. Изучение развития науки как информационного процесса / В. В. Налимов, З. М. Мульченко. – Букинист, 1969. – 192 с.
19. Вебинар Clarivate. Вступление в наукометрию: сайт. – 2020. – URL: <https://clarivate.ru/webinars> (дата обращения: 01.12.2020). – Режим доступа: для зарегистрированных пользователей.
20. Вебинар Clarivate. Поиск в пристатейной литературе: доступные сокровища Web of Science: сайт. – 2020. – URL: <https://clarivate.ru/webinars> (дата обращения: 03.12.2020). – Режим доступа: для зарегистрированных пользователей.
21. OntoMathPro: центр математического связывания открытых данных: сайт. – 2016. – URL: <https://github.com/CLLKazan/OntoMathPro> (дата обращения: 16.12.2020).
22. Он-лайн форум Big Data 2021: сайт. – 2021. – URL: <https://www.osp.ru/lp/bigdata2021> (дата обращения: 25.03.2021). – Режим доступа: для зарегистрированных пользователей.
23. Digital Decisions. Вопросно-ответные системы: о методах поиска ответа на запрос: сайт – 2021. – URL: <https://dddecisions.ai/qasystems> (дата обращения: 02.06.2021).
24. Среда описания ресурса (RDF): Понятия и Абстрактный Синтаксис: сайт – 2014. – URL: https://www.w3.org/2007/03/rdf_concepts_ru (дата обращения: 12.11.2020).

25. RDF Primer: сайт – 2014. – URL: <https://www.w3.org/TR/rdf-primer> (дата обращения: 14.11.2020).
26. Modeling and evaluation of the mathematical educational ontology / L. Shakirova et al. // CEUR Workshop Proceedings. – 2020. – V. 2543. – P. 305-319.
27. **Nevzorova, O.** OntoMathEdu Educational Mathematical Ontology: Annotation of Concepts. / O. Nevzorova et al. // SCOPUS16130073-2020-2648-SID85092283061. – 2020.
28. **Nevzorova, O. A.** OntoMath PRO ontology: a linked data hub for mathematics / O. A. Nevzorova et al. // International Conference on Knowledge Engineering and the Semantic Web. – Springer, Cham, 2014. – P. 105-119.
29. **Berners-Lee, T.** Linked Data. Design Issues / T. Berners-Lee // Linked Data: website. – 2006. – URL: <https://www.w3.org/DesignIssues/LinkedData.html> (дата обращения: 16.01.2021).
30. MSDN: Конфигурация в .NET Core: сайт. – 2021. – URL: <https://docs.microsoft.com/ru-ru/aspnet/core/fundamentals/configuration/?view=aspnetcore-5.0> (дата обращения: 02.06.2021).
31. **Beck, K.** Test-driven development: by example. / K Beck. – Addison-Wesley Professional, 2003.
32. Исторический-сайт.рф: сайт. – 2020. – URL: <https://исторический-сайт.рф> (дата обращения: 14.06.2021).
33. Электронный фонд правовых и нормативно-технических документов: сайт. – 2012. – URL: <https://docs.cntd.ru/document/9041994> (дата обращения: 14.06.2021).

Приложение 1. Карта преобразований

```
true
дочерние элементы класса (.*) ; что входит в состав класса
(.*) ; зависимые определения (.)
дочерние элементы: {0} ; в состав класса входит: {0} ; обнаруженные
зависимые определения: {0}
SELECT DISTINCT ?name ?comment
WHERE {{
    ?target rdfs:label ?name.
    ?target rdfs:comment ?comment.
    ?target rdfs:subClassOf ?parent.
    ?parent rdfs:label ?label.
    FILTER(LCASE(STR(?label)) = LCASE("{0}"))
    FILTER langMatches(lang(?name), "ru")
}}
+
true
что нужно знать перед изучением (.*) ; в какой класс входит (.*) ; в какие
классы входит (.)
сперва вам стоит ознакомиться с: {0} ; входит в: {0} ; входит в: {0}
SELECT DISTINCT ?label ?comment
WHERE {{
    ?target rdfs:label ?name.
    ?target rdfs:subClassOf ?parent.
    ?parent rdfs:label ?label.
    ?parent rdfs:comment ?comment.
    FILTER(LCASE(STR(?name)) = LCASE("{0}"))
    FILTER langMatches(lang(?label), "ru")
}}
+
true
где фигурирует (.*) ; что фигурирует в литературе под авторством (.)
были обнаружены следующие косвенно связанные сущности: {0} ; обнаружено
следующее: {0}
SELECT DISTINCT ?label ?comment
WHERE
{{
    ?entity rdfs:comment ?comment.
    ?entity rdfs:label ?label.
    FILTER CONTAINS(LCASE(STR(?comment)), LCASE("{0}"))
    FILTER langMatches(lang(?label), "ru")
}}
+
```

```

false
сколько всего концептов в онтологии; сколько триплетов
всего в используемой онтологии содержится {0} концептов; в онтологии
содержится {0} триплетов
SELECT DISTINCT (COUNT(?s) AS ?triples)
WHERE
{
    ?s ?p ?o
}
+
true
синонимы (.* )
существуют следующие варианты названий: {0}
SELECT DISTINCT ?synonym ?comment
WHERE
{{
    ?target rdfs:label ?synonym.
    ?target rdfs:label ?concrete.
    ?target rdfs:comment ?comment.
    FILTER(LCASE(STR(?concrete)) = LCASE("{0}"))
}}
+
true
термины содержащие (.* )
обнаруженные совпадения: {0}
SELECT DISTINCT ?label ?comment
WHERE
{{
    ?target rdfs:label ?label.
    ?target rdfs:comment ?comment.
    FILTER CONTAINS(LCASE(STR(?label)), LCASE("{0}"))
}}
+

```