Министерство образования и науки Российской Федерации
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(государственный университет)
ФАКУЛЬТЕТ УПРАВЛЕНИЯ И ПРИКЛАДНОЙ МАТЕМАТИКИ
КАФЕДРА ТЕОРЕТИЧЕСКОЙ И ПРИКЛАДНОЙ ИНФОРМАТИКИ

Технология построения онтологий из текстов на естественном языке

Магистерская диссертация
студента 873 группы
Хламова Максима Анатольевича

Научный руководитель
Рыков В.В., доцент, кандидат филологических наук

Рецензент
Устюжанин А.Е., к.ф.-м.н.

г. Долгопрудный
2014

Содержание

1. Введение

Существуют различные подходы, модели и языки описания данных и знаний. Однако все большую популярность последнее время приобретают онтологии. *Онтология* – по классическому определению Томаса Грубера [1]– *спецификация концептуализации*, формализованное представление основных понятий и связей между ними. Таким образом, онтология – это описание (подобно официальной спецификации на программный продукт) понятий (концептов) и отношений, которые могут существовать как агент или сообщество агентов. Это определение совместимо с использованием онтологии как набора описаний понятий (set-of-concept-definitions), но более общее. В философии у этого слова другой смысл (где онтология – это систематическая оценка существования (жизни)).

Онтологии часто приравнивают к таксономическими иерархиями классов, определениям класса и внутренним отношениям, но онтологии не должны быть ограничены этими формами. Онтологии также не ограничены консервативными определениями, то есть определениями в традиционном логическом смысле, которые только представляют терминологию и не прибавляют никакого знания относительно мира [2]. Для определения концепции, требуется определить аксиомы, которые ограничивают возможные интерпретации для определенных терминов.

Онтологии могут быть использованы, чтобы усовершенствовать существующие сетевые приложения и сделать возможным новые варианты использования сети. Предполагается, что уже в самом ближайшем будущем онтологии будут активно использоваться для различных целей. Приведем здесь некоторые примеры [5]:

- Для того чтобы позволить более интеллектуальное публичное представление, веб порталы могут определить онтологии для сообщества. Эта онтология может обеспечить терминологию для описания контента и аксиом, которые описывают термины, используя другие термины из онтологии. Например, онтология может включать такую терминологию, как "статья журнала", "публикация", "персона" и "автор". Эта онтология может включать определения, которые заявляют такие факты, как "все журнальные статьи есть публикации" или "авторы всех публикаций это люди". Если эти определения соединить с правдивыми фактами, то можно логически вывести новые факты. Эти выводы могут в свою очередь позволить пользователям получать результаты поиска от портала, которые невозможно получить от

традиционных поисковых систем. Здесь уже можно смело говорить о том, что система генерирует новые знания.

- Онтологии могут быть использованы для того, чтобы обеспечить семантическую аннотацию для коллекций изображений, звуковых и других нетекстовых объектов. Искусственному интеллекту сложнее извлекать поддающуюся интерпретации семантику из мультимедиа, чем семантику из текстов на естественном языке. Таким образом, такие типы ресурсов обычно индексируются с помощью надписей или метатегов. Однако так как разные люди могут описывать эти нетекстовые объекты различными способами, важно, чтобы функциональность поиска превышала простой поиск по ключевым словам. В идеале онтологии будут получать дополнительные знания о предметной области, которые могут быть использованы для более совершенного поиска изображений, видео, звуковых файлов и других мультимедийных объектов.

  Как пример мультимедиа коллекции рассмотрим архив изображений старинной мебели. Онтология старинной мебели была бы подходящим решением в поиске по такому архиву. Чтобы классифицировать различные типы мебели, можно использовать таксономию. Также необходима возможность получать знания из определений. Например, если индексатор выбирает значение "Из конца эпохи Георгов" для стиль/период для комода, из этого следует возможность вывести, что элемент данных "дата.созданно" должен иметь значение между 1760 и 1811 нашей эры, и что "культура" - британская. Доступность такого типа фоновых знаний значительно увеличивает поддержку того, что может быть выдано при индексации, также хорошо, как и при поиске. Другая особенность, которая может быть полезной, это поддержка представления знаний "по умолчанию". Примером таких знаний может быть то, что "Ящик комода поздней эпохи Георгов" в отсутствии другой информации предположительно сделан из красного дерева. Это знание ключевое для реальных семантических запросов, например запрос пользователя "старинная мебель для хранения из красного дерева" мог бы сопоставить изображения ящиков комода поздней эпохи Георгов, даже если ничего не сказано о типе дерева в аннотации изображения.
- Онтологии также могут быть полезны при разработке документации. Это вариант использования для технической документации огромного объема, такой как та, которая например, используется в аэрокосмической промышленности. Эта документация может быть нескольких различных типов, включая документацию разработки, документацию производства и документацию тестирования. Эти пакеты

документов имеют иерархическую структуру, но структура различается для разных пакетов.

2. Постановка задачи

2.1 Цель исследований

Цель исследования – разработать систему автоматического построения онтологий из текста на так называемом «усеченном» русском языке, который подробнее будет описан в основной часть данной работы.

В рамках данной работы ставятся следующие задачи:

- Изучение существующих подходов к построению онтологий
- Разработка формального языка описания онтологии, максимально похожего на естественный русский язык
- Программная реализация основных семантических отношений

Новизна работы состоит в построении такой системы с учетом особенностей русского языка.

Объектом исследования является построение технологии кодирования онтологий из текста на естественном языке.

2.2 Актуальность темы исследования.

Сегодня как никогда актуальной является задача построения онтологий различных предметных областей (ПО), например сложных технических объектов. [9]

При построении онтологий, которые зачастую можно рассматривать также и как базы знаний, приходится сталкиваться со следующими проблемами (некоторые из которых могут показаться странными для специалистов - технарей) [11]:

1. Инженер знаний - человек, занимающийся построением онтологии, - должен обладать знаниями во многих областях: искусственном интеллекте, представлении знаний, моделировании знаний. Также необходимо глубоко знать сопутствующие средства моделирования баз знаний, например графические средства. Недостаточное понимание тонкостей в вышеупомянутых областях может привести к недостатку ключевой информации в смоделированной системе.
2. Для построения онтологии нужно как минимум два человека — специалист в данной области и инженер знаний (как в ЭС). Иногда эти два специалиста могут выступать в одном лице — если этот специалист владеет когнитивными знаниями и технологиями. Однако чаще происходит наоборот — специалист по тем или

иным причинам не участвует в этом процессе и вникать зачастую в незнакомую ему область приходится инженеру знаний. Поэтому актуальной является технология построения онтологии, основанной на доступной информации, представленной в виде текстов, разного типа словарей и других материалов.

3. Не существует единственного правильного способа моделирования предметной области – всегда существуют жизнеспособные альтернативы. Лучшее решение почти всегда зависит от предполагаемого приложения и ожидаемых расширений.
4. Разработка онтологии – это обязательно итеративный процесс.
5. Понятия в онтологии должны быть близки к объектам (физическим или логическим) и отношениям в интересующей предметной области.

Таким образом, существует потребность в создании удобных систем автоматического построения онтологий из текстов на естественном языке.

2.3 Степень разработанности

Как мы уже увидели, существует потребность в создании удобных систем автоматического построения онтологий. Можно выделить несколько подходов к решению этой проблемы:

1. Визуальный подход, позволяющий специалистам непосредственно "рисовать" онтологии, что помогает наглядно сформулировать и объяснить природу и структуру явлений. Любой программный графический пакет от PaintBrush до Visio можно использовать как первичный инструмент описания онтологий. К такого рода системам можно отнести CAKE (Computer Aided Knowledge Engineering), впервые описанный в работе [12] и ВИКОНТ - ВИзуальный Конструктор ОНТологий [13]. CAKE и ВИКОНТ позволяют визуально проектировать онтологии любой предметной области. В работе [14] был описан способ построения онтологий из UML-диаграмм.

   Основной недостаток систем такого рода – негибкость, невозможность описать множество отношений из-за того, что "рисунок" получается слишком загроможденным.

2. В статье [8] описан метод построения тезаурусов, терминологических онтологий из текста.

Возможность построения онтологий из текстов на «чистом» естественном языке полностью решила бы проблему построения онтологий. Однако такой подход имеет некоторые критические для автоматической обработки текста недостатки:

- Наличие у слова нескольких значений. Самая сложная неопределенность, возникающая в естественных языках, заключается в правильном выборе смысла слова в каком-либо определенном контексте.
- Присоединение предложных групп. В русском языке предложные группы можно добавлять к существительным, глаголам, прилагательным или наречиям.
- Ссылочные именные группы. В естественных языках объект ссылки местоимения или другой именной группы может потребовать неявных фоновых знаний.
- Модификаторы существительных. В отличие от прилагательных, таких как желтый, отражающих атрибут, слово жесткий, как в жесткий диск, приводит к значительному изменению значения существительного.
- Глубоко вложенные предложения. В формальных нотациях вводные слова или правила предшествования определяют группировку предложений, но группировка в русском языке часто зависит от контекста.
- Некоторые сложнейшие проблемы, продолжающие исследоваться лингвистами, включают именные группы во множественном числе, времена глаголов, модальность, а также неокончательное число контекстно-зависимых вопросов.

3. Построение онтологий из текстов на упрощенном естественном языке.
   Примером таких технологий является редактор Fluent Editor. Этот редактор решает задачу автоматического построения онтологии с помощью преобразования отдельных простых предложений на английском языке.[15] Основной недостаток данного редактора – возможность анализа предложений только на английском языке.

3. Основной раздел

3.1 Теоретическое обоснование

Выбранный в данной работе подход имеет под собой теоретическую основу.

Одной из задач, повышающих эффективность человеческой деятельности при проектировании технических, информационных и программных систем, является обеспечение естественного способа коммуникации с соответствующей автоматизированной системой, поддерживающей данный процесс. Имеется много разработок в этой области, однако универсальной системы, обеспечивающей наиболее естественный вариант взаимодействия (естественно-языковой интерфейс) пока не создано, так как ни одна из имеющихся лингвистических теорий не в состоянии описать естественный язык с необходимой точностью и полнотой. Это связано со сложностью объекта формализации – естественного языка (ЕЯ).

В результате анализа основных формальных лингвистических теорий, для которых существуют компьютерные реализации, можно сделать вывод о том, что проблема моделирования языковой деятельности решается кибернетическим методом черного ящика, сводится к выработке методов строгого лингвистического анализа и к построению точного описания лингвистических объектов и соответствующих понятий.

При этом четкое теоретико-множественное моделировании языковых объектов, которые по своей природе не могут рассматриваться независимо друг от друга, от механизмов реальной деятельности и от когнитивных структур, которыми пользуется человек, невозможно.

Необходимы методы моделирования, реализующие соответствующие взаимосвязи. Таким образом, является актуальной задача формализации ограниченного естественного языка Одной из классических работ в этом направлении является работа Новикова «Семантика текста и ее формализация» [16]. Новиков, анализируя подходы к структуре текста, отмечает, что «Из-за многоплановости, многоуровневости своей организации текст представляет собой довольно сложный объект исследования. Между его единицами могут быть определены различные системы отношений, то есть одному и тому же тексту может быть поставлено в соответствие несколько различных структур»

Процесс формализации текста как метод исследования обоснованно может быть причислен к продуктивным методам, потенциал которого не раскрыт в полной мере. В настоящее время происходит активное проникновение кибернетических методов в моделирование мыслительной деятельности человека. А.И.Новиков писал, что одним из параметров, который способен наиболее полно охарактеризовать этот формализации

текста, является соотношение имплицитной и эксплицитной информации. Выбор этого параметра А.И.Новиков объяснял просто - имплицитная информация представляет собой, с одной стороны, обязательную составляющую речемыслительного процесса, а с другой стороны, этот вид информации - основное препятствие в процессе формализации.

Прежде всего обратим внимание на основные требования и исходящие из них ограничения, которые присущи формализации и на которые ссылался А.И.Новиков. Исходные единицы должны быть однозначными, элементарными, количество их должно быть небольшим и конечным. Соответственно этапы построения выражений из таких единиц тоже будут элементарными, обусловленными и близкими к автоматическому исполнению. Формализации поддадутся только элементарные (с простой логической структурой) части с ограниченным запасом значений. При других условиях формализация невозможна.

Следует отметить еще одну особенность формализации – абстрагированность символов от содержания, которое символы замещают при формализации. Новиков писал, что невозможно добиться полного абстрагирования от содержания, следовательно «чистых» формальных систем не существует.

3.2 “Усеченный русский язык”

«Усеченный» русский язык (УРЯ) — формальный язык с синтаксисом, похожим на обычный русский язык и поддерживающим возможность перевода в логику первого порядка. Любой, кто способен читать по-русски, может читать на УРЯ без дополнительных тренировок. Однако, чтобы писать на УРЯ, необходима практика, чтобы учитывать синтаксические и семантические ограничения. Важнейшее ограничение УРЯ заключается в том, что значение каждого предложения на УРЯ определяется его переводом в логику первого порядка; не поддерживается никакая гибкость обычного русского языка и никакая его образность. Эти ограничения знакомы каждому, кто использовал языки запросов БД, разрабатывал ПО или формальные спецификации, такие как SQL, UML, OWL и др.. Так как эти языки могут быть автоматически переведены в логику и обратно, то они также могут быть переведены на УРЯ и наоборот. Следовательно, УРЯ может быть использован как удобочитаемый язык документации, который можно перевести в исполнимый язык. Основное синтаксическое ограничение — использование глаголов настоящего времени и существительных единственного числа, переменных вместо местоимений и небольшого подмножества синтаксических возможностей русского языка. Несмотря на эти ограничения, УРЯ похож на тот русский

язык, который используется в программных спецификациях, математических книгах и определениях и аксиомах формальной онтологии.

Несмотря на широкую применимость, УРЯ не претендует на статус стандарта; стандартом является логика первого порядка, а УРЯ — лишь удобная нотация, позволяющая сделать логику легче для чтения и написания. Так как УРЯ обладает всей выразительной силой логики первого порядка, есть возможность перевести утверждение логики в исчисление предикатов или во многие другие нотации. Обратный перевод из логики в УРЯ можно сделать автоматическим, но с некоторыми оговорками:

1. Если утверждение логики было получено из УРЯ, тогда описания имен и других слов, использованных для перевода УРЯ в логику, также должны быть использованы и для перевода из логики в УРЯ.
2. Если утверждение логики было получено не из УРЯ, тогда перевод в УРЯ может быть выполнен только в том случае, если отображения символов, использованных в логике, в слова в УРЯ были заданы той же информацией, что и в описаниях УРЯ.
3. Так как и УРЯ, и логика предоставляют много способов описания одного и того же утверждения, обратный перевод может быть не идентичен исходному утверждению УРЯ, но при этом логически эквивалентен.

В качестве примеров обратных переводов рассмотрим следующие три предложения на УРЯ, являющиеся логически эквивалентными:
Любое простое число меньше 3, есть четное.

Для любого числа x, если x есть простое, и x меньше 3, тогда x есть четное.

Для любого x, если x есть число, x есть простое, и x меньше 3, тогда x есть четное.

4. Всегда можно доказать то, что обратный перевод эквивалентен исходному утверждению. Фактически, число обменов и замен, требуемых для доказательства, прямо пропорционально длине предложения на УРЯ.

Возможность переводов в обоих направлениях позволяет использовать УРЯ в качестве языка документации, который всегда согласован с реализацией: любые изменения в документации либо в реализации могут быть всегда переведены в реализацию или документацию соответственно. Ошибки и опечатки, которые довольно трудно обнаружить в незнакомой записи, в УРЯ часто бывает найти легче, и они могут быть найдены людьми, никогда не изучавшими УРЯ.

В качестве иллюстрации УРЯ, рассмотрим несколько примеров, демонстрирующих некоторые виды разрешенных предложений и их переводы в исчисление предикатов (ИП). Следующий пример содержит два квантора: квантор общности любая в УРЯ; квантор существования пробел в УРЯ. Также здесь видно как глагол есть связывает именную группу любая кошка с предложной группой на коврике.

УРЯ: Любая кошка есть на коврике.

ИП: $(\forall$ x:Кошка)$(\exists$ y:Коврик)На(x,y)

В следующем примере используется некоторый для выражения квантора существования. Также здесь демонстрируется трехместный предлог между, который содержит список из двух значений в качестве второго и третьего аргументов. Чтобы избежать неопределенности, списки должны быть заключены в круглые скобки.

УРЯ: Некоторый человек есть между скалой и труднодоступным местом.

ИП: $(\exists$ x:Человек$)(\exists$ y:Скала$)(\exists$ z:Место)(Между(x,y,z) $\wedge$ Труднодоступное(z))

3.2.1 Устранение неопределенностей

В отличие от русского языка, чьи основы требуют разрешения многих неопределенностей, УРЯ имеет синтаксис, позволяющий решить любую неопределенность в момент разбора предложения. Описания, необходимые для перевода с УРЯ на логику первого порядка, также могут быть использованы во время разбора в момент выбора подходящего смысла для каждого слова и создания перевода на другие нотации. Далее рассмотрим типы неопределенностей, возникающие в русском языке, и ограничения, накладываемые УРЯ для их избегания:

1. **Несколько смыслов слова.** Самая сложная неопределенность, возникающая в естественных языках, заключается в правильном выборе смысла слова в каком-либо определенном контексте. УРЯ решает эту проблему жестко: нужный смысл слова однозначно определяется синтаксисом, а смысл определенного пользователем слова ограничен единственным отношением, заявленным для этого слова.

2. **Присоединение предложных групп.** В русском языке предложные группы можно добавлять к существительным, глаголам, прилагательным или наречиям. В УРЯ единственным предлогом, который можно подсоединить к существительному, является из, все остальные предлоги можно добавить лишь к глаголам.
3. **Действие кванторов.** В отличие от русского языка, позволяющего кванторам общности и существования смешиваться в разных контекстах, УРЯ ограничивает положение кванторов общности двумя вариантами: подлежащее (субъект) высказывания или префикс, помещаемый в начало высказывания; оставшиеся кванторы, стоящие в других местах, должны быть кванторами существования. Это ограничение позволяет определять области действия кванторов из синтаксиса.
4. **Ссылочные именные группы.** В естественных языках объект ссылки местоимения или другой именной группы может потребовать неявных фоновых знаний. В УРЯ местоимения заменяются явными переменными, а переменные могут быть пропущены только по узко специализированным условиям.
5. **Модификаторы существительных.** В отличие от прилагательных, таких как желтый, отражающих атрибут, слово жесткий, как в жесткий диск, приводит к значительному изменению значения существительного. Подобные модификаторы должны быть объединены с существительным как предопределенный терм, включающий знак подчеркивания, например жесткий_диск.
6. **Глубоко вложенные предложения.** В формальных нотациях вводные слова или правила предшествования определяют группировку предложений, но группировка в русском языке часто зависит от контекста. В УРЯ вводные слова применяются для глубоко вложенных предложений и для списков из более, чем двух элементов.
7. **Возможности, не доступные FOL.** Некоторые сложнейшие проблемы, продолжающие исследоваться лингвистами, включают именные группы во множественном числе, времена глаголов, модальность, а также неокончательное число контекстно-зависимых вопросов — все они исключены ограничениями синтаксиса и семантики УРЯ.

Возможность присоединения предложных групп к различным частям речи избегается УРЯ посредством ограничения на то, что предлоги могут быть добавлены только к глаголу или к отглагольному причастию. В русском языке большое количество возможностей может смутить говорящего, как например в следующем предложении:

Вася видит человека на холме с телескопом.

Предлог на можно присоединить как к глаголу видит, так и к существительному человека, а предлог с можно присоединить к видит, человеку или холму. В УРЯ единственным предлогом, который можно добавить к существительному, является из. Поэтому единственно допустимая интерпретация заключается в том, что процесс наблюдения имеет место на холме и с помощью телескопа. Другие возможности могут быть выражены на УРЯ с помощью определительных придаточных:

Вася видит человека, который есть на холме, с телескопом.

Вася видит человека на холме, который есть с телескопом.

Вася видит человека, который есть на холме, который есть с телескопом.

Вася видит человека, который есть на холме и который есть с телескопом.

Несмотря на то, что все пять предложений точно выражены на УРЯ, первые три могут смутить русскоговорящего человека, незнакомого с особенностями УРЯ. Поэтому эти предложения можно выразить точно как с точки зрения УРЯ, так и обычного русского языка:

На холме, с телескопом, Вася видит человека.

С телескопом, Вася видит человека, который есть на холме.

На холме, который есть с телескопом, Вася видит человека.

Несмотря на то, что компьютеры в отличие от людей не так хороши в разрешении неопределенностей, они гораздо лучше людей справляются с обнаружением этих неопределенностей. Следовательно, можно разработать инструментарий УРЯ, проверяющий предложения на УРЯ, которые могут быть неоднозначными в обычном русском языке, и предлагающий возможные варианты, не содержащие неопределенностей как на УРЯ, так и на русском языке.

Некоторые глаголы могут объединяться с предлогом из, например состоит из. К счастью эти глаголы не имеют непосредственного объекта. Во избежание возможных неопределенностей ни один глагол в УРЯ, имеющий предлог из, также не имеет непосредственного объекта.

3.2.2 Предложения

Предложения УРЯ похожи на подмножество предложений обычного русского языка. Как и предложения на русском языке каждое предложение на УРЯ состоит из одного или нескольких *высказываний*, а каждое высказывание имеет один главный глагол. Предложения можно разделить по двум принципам: в зависимости от их структуры предложения могут быть *простыми*, *составными* или *сложными*; в зависимости от их использования предложения могут быть *повествовательными*, *вопросительными* или *повелительными*. Рассмотрим несколько примеров:

1. **Простое.** Простое предложение состоит только из одного высказывания. Высказывание может быть коротким или длинным, как в примерах:

Снежок - это кот.

Маша дает собаке очень сочную кость с горячей сковороды.

2. **Составное.** Составное предложение имеет одно *главное высказывание* и одно или несколько *подчиненных* высказываний, *зависящих* от главного высказывания. УРЯ имеет два типа подчиненных высказываний: *определительное придаточное*, начинающееся с *относительного местоимения* который; и *условное выражение*, присоединяющееся к главному высказыванию фразами только если, тогда и только тогда или раздельной комбинацией если и тогда. Рассмотрим примеры:

Любая кошка, которая есть на коврике, есть довольная.

Если число есть четное,
тогда число есть не нечетное.

3. **Сложное.** Сложное предложение состоит из двух или более независимых предложений, которые могут быть как простыми, так и составными. Предложения присоединяются словом и или словом одно из двух, а также одним или более вхождением слова или. Рассмотрим примеры:

Некоторая кошка есть на коврике, кошка есть рыжая,
и коврик есть голубой.

Для любого целого_числа, одно из двух целое_число есть четное,
или целое_число есть нечетное.

Некоторые сложные предложения, как например эти, можно записать простыми предложениями:

Некоторая рыжая кошка есть на голубом коврике.

Любое целое_число есть одно из двух: четное или нечетное.

Когда обе версии предложений УРЯ переводятся на исчисление предикатов, результат получается одним и тем же либо с небольшими различиями. Переводы на исчисление предикатов двух предложений про целые числа одинаковы, а переводы о кошке различаются только порядком следования предикатов:

$(\exists x{:}\text{Кошка})(\exists y{:}\text{Коврик})(\text{На}(x,y) \wedge \text{Рыжая}(x) \wedge \text{Голубой}(y))$

$(\exists x{:}\text{Кошка})(\exists y{:}\text{Коврик})(\text{Рыжая}(x) \wedge \text{Голубой}(y) \wedge \text{На}(x,y))$

4. **Повествовательное.** Повествовательное предложение объявляет суждение, которое может быть простым фактом или составным утверждением, имеющим множество условий и возможностей. Все примеры в первой части раздела являются повествовательными предложениями.
5. **Вопросительное.** Вопросительное предложение задает вопрос. Оно обычно следует за последовательностью повествовательных предложений, утверждающих некоторые факты или аксиомы, из которых может быть получен ответ на вопрос. Для последующего вопроса о числах, ответ может быть выведен из самого вопроса и предшествующих определений семантики слов четный, число и меньше. В случае с вопросом о довольных кошках должен быть объявлен только синтаксис кошки, коврика и довольный; ответ можно вывести из информации, данной в условном выражении.

Какое четное число есть меньше 3?

Если любая кошка, которая есть на коврике, есть довольная,
правда ли, что некоторая кошка, которая есть недовольная, есть на коврике?

Последовательность предложений, разделенных точкой с запятой, называется *распространенным предложением*. Главное предназначение распространенного предложения заключается в том, чтоб позволить области действия переменных и кванторов распространяться за пределы отдельного предложения. В следующем примере первая именная группа переводится на квантор существования, такой как (∃ x:Number). В следующем предложении *ссылочная* именная группа это число представлена ссылкой на ту же переменную x.

Четное число больше 6 есть меньше 9;
что есть это число?
Знак вопроса в конце последнего предложения ограничивает область действия всех кванторов в распространенном предложении.

3.2.3 Слова

Слова УРЯ разделяются на небольшое количество *зарезервированных слов* и на неограниченное число *объявленных слов*. Все зарезервированные слова записываются как и русские слова, но их значения ограничены только одним смыслом либо малым числом смыслов, различимых синтаксисом. Имеется десять типов зарезервированных слов:

1. **Булевы операторы:** не, и, одно из двух, или, ни одно из двух, тоже не, если, тогда.
2. **Кванторы:** некоторый, нечто, некто, любой, всё, все, нет, ничто, никто.
3. **Специальные глаголы:** есть, имеет, “- это”.
4. **Вопросы:** кто, что, когда, где, какой.
5. **Относительное местоимение:** который.
6. **Соединитель списков:** и.
7. **Специальные списки:** пустой, остальные, ничего больше, никого больше.
8. **Маркеры аргументов:** из, чем, как.
9. **Специальные фразы:** существует; как этот; только если; тогда и только тогда; ложно, что; истинно, что.

Некоторые зарезервированные слова имеют более одного смысла, но правильный смысл всегда может быть получен из синтаксиса.

Объявленные слова могут обозначать имена, существительные, глаголы, прилагательные, наречия или предлоги. Это могут быть слова с тем же написанием, что и слова русского языка, слова, составленные с помощью подчеркивания жесткий_диск, или произвольные

последовательности букв, цифр и подчеркиваний. Перед использованием слова в УРЯ его синтаксис и отображение в отношение логики первого порядка должны быть заданы *декларацией*.

3.2.4 Имена

Имя — это слово, которое отсылает к некоторой сущности, называемой *ссылкой*. Имена УРЯ представляют собой строки символов, заключенных в одинарные кавычки; любая одинарная кавычка, появляющаяся в имени, должна быть продублирована, например 'O''Reilly'. Если символы, появляющиеся в имени, являются буквами, цифрами и подчеркиваниями, заключение в кавычки можно опустить.

Имена можно объявить в *декларации*, являющейся повелительным предложением, начинающемся с глагола *объявить*. Эта декларация также может обозначать одноместный предикат, принимающий значение истины для сущности, на которую ссылается имя.

Объявить 'Grand Central Terminal' как имя строения,
Снежок как имя кота,
Сью как имя женщины.

Имена Снежок и Сью представляются кванторами существования (∃Пушок:Кот) и (∃Сью:Женщина). Имя, включающее специальные символы, переводится во внутреннее генерируемое имя, начинающееся с подчеркивания и не содержащее никаких специальных символов:

(∃_GrandCentralTerminal:Строение)
(_GrandCentralTerminal='Grand Central Terminal')

Область действия подобных имен распространяется на весь текст, в котором данная декларация встречается. После того, как имя объявлено в тексте, оно может быть использовано в любом последующем предложении в тексте как законченная именная группа сама по себе.

Имя также может быть объявлено именной группой. Следующее предложение объявляет те же три имени, что и в предыдущей декларации, а также заявляет отношение их ссылок:

The woman Sue sees the cat Yojo
in the building 'Grand Central Terminal'.

Три имени представляются в логике первого порядка так же и с той же областью действия, как если б они появились в декларации. Если имена уже были объявлены, то же

суждение может быть объявлено с помощью имен без добавления существительного типа cat:

Sue sees Yojo in 'Grand Central Terminal'.

После того, как имя было объявлено, оно может быть использовано в любой последующей точке того же текста. Оно может быть законченной именной группой само по себе либо может предваряться существительным, определяющим предикат, который является истиной, если отправляет к его ссылке.

Понятия должны начинаться с маленькой буквы, в отличие от экземпляров классов. Из понятий, состоящих из нескольких слов, например «жесткий диск», необходимо формировать слова с подчеркиванием: жесткий_диск.

Два типа сущностей, целые числа и символьные строки, имеют специальные имена, не требующие объявления, так как синтаксис имени единственно определяет ссылку и ее тип:

1. **Целые числа**. Любое имя, состоящее из строки из одной или более цифр, принимается за имя целого числа. Имя типа 437 может быть использовано без декларации, хотя расширенная форма целое_число 437 допустима.
2. **Символьные строки**. Символьная строка, заключенная в двойные кавычки принимается за имя как таковое; любые двойные кавычки внутри строки должны удваиваться. Как и в случае имен целых чисел, закавыченная строка типа "abc" может быть использована без декларации, но также допустимо и использование расширенной формы строка "abc".

В добавок к именам, чья область действия простирается на весь текст, имеются временные имена, называемые *переменными*, чья область действия ограничивается отдельным предложением или даже еще меньше. Переменной является любое имя, состоящее из одной из букв x, y, z или w, следующей за необязательной строкой цифр. При первом появлении в предложении переменная должна входить как в именную группу, начинающуюся с квантора существования, так и в именную группу, начинающуюся с квантора общности. В последующие появления в предложении переменная может появляться само по себе или в именных группах типа множество x.

3.2.5 Существительные

В отличие от имени, которое ссылается на особую сущность в предметной области, существительные могут ссылаться на любую сущность заданного типа. УРЯ поддерживает четыре типа существительных: *категориальные существительные*, *относительные существительные*, *функциональные существительные* и *вариативные относительные или функциональные существительные*. Все эти существительные можно представить в логике первого порядка с помощью меток типа и предикатов или отношений.

1. **Категориальные существительные**. Существительное типа собака или стол, которое обозначает тип, или *категорию,* сущности, называется категориальным существительным. Оно представляется *меткой типа* в кванторе, например (∃ x:Dog), или одноместным предикатом, например Dog(x). Следующая запись объявляет три категориальных существительных:

Объявить собака как существительное,
стол как существительное,
гончая как существительное с отношением(собака).

По умолчанию представлением для существительного является одноместный предикат или метка типа с тем же написанием, что и само существительное; объявление гончая показывает, как задается предикат с отличающимся написанием. В результате оба существительных собака и гончая представлены одним и тем же предикатом: собака(x).

2. **Относительные существительные**. Существительное типа ребенок или служащий имеет два назначения: как и категориальное существительное, оно обозначает тип сущности, но также оно обозначает и двухместное отношение к некой другой сущности. Когда относительное существительное представляется двухместным отношением, например ребенок(x,y) или служащий(x,y). С одним аргументом соответствующий одноместный предикат ребенок(x) или служащий(x) истинен в зависимости от некой сущности, называемой *фокусом* двухместного отношения. Другой аргумент отношения, называемый *коррелятом*, может быть фокусом другого относительного существительного, например родитель или работодатель. В предложении Вася есть ребенок Маши, Вася является фокусом, а

Маша — коррелятом; но они меняются ролями в предложении Маша есть родитель Васи.

Объявить ребенок как существительное (x1 ребенок x2),
служащий как существительное (x1 служащий x2),

В этой декларации выражения в скобках являются *образцами*, демонстрирующими, как существительное используется в предложении УРЯ. Две переменные x1 и x2 показывают, что каждое существительное представлено двухместным отношением. В этом примере переменная x1 демонстрирует фокус, но в следующем примере фокусом является уже переменная x2:

Объявить родитель как существительное (x2 родитель x1) с отношением(ребенок),
работодатель как существительное (x2 работодатель x1) с отношением(служащий).

При переключении фокуса и коррелята, те же два отношения ребенок(x1,x2) и служащий(x1,x2) могут использоваться для определения относительных существительных родитель и работодатель.

3. **Функциональные существительные**. Существительное типа мать является особым случаем относительного существительного, имеющим отдельное значение для каждого варианта его коррелята. Поскольку ребенок имеет двух родителей, а каждый родитель имеет одного или более детей, каждый ребенок имеет ровно одну мать. Следовательно, относительное существительное мать называется *функциональным существительным*, так как обозначает функцию из множества детей в множество матерей.

Объявить мать как функциональное существительное (x1 мать x2),
сумма как функциональное существительное (x3 сумма x1 и x2) с отношением("+").

В первой строке мать определяется так же, как и ребенок, но с дополнительным спецификатором функциональное. Во второй строке сумма определяется трехместным отношением, чья запись является специальным символом "+", который должен быть заключен в кавычки. Пример показывает, что фокус функции, также называемый *результатом*, — это третий аргумент отношения.

4. **Вариативные относительные существительные**. Несмотря на то, что большинство относительных и функциональных существительных в русском языке представляются двухместными отношениями, некоторые относительные существительные имеют *вес* больше, чем 2, а некоторые называются *вариативными* из-за того, что имеют вариативный вес. Примером вариативного функционального существительного является множество, которое можно представить одноместным предикатом множество(x), чтобы показать, что x — это множество или оно может иметь *n*+1 аргументов для демонстрации множества из *n* элементов, состоящих из его последних *n* аргументов. Три точки в следующей записи показывают ноль или больше дополнительных аргументов:

Объявить множество как функциональное существительное (x1 множество из x2 ...).

Следующее предложение объявляет пустое_множество в качестве имени множества и показывает, что оно не содержит ни одного элемента:

Ничто есть элемент множества пустое_множество.

В переводе в логику первого порядка слово ничто представляется квантором существования с отрицанием без метки типа ~(∃x). Отсутствие метки типа означает, что квантор распространяет свою область действия на все в предметной области.

ИП: (∃пустое_множество:Множество)~(∃x)(x ∈ пустое_множество)

С тех пор, как все имена имеют глобальную область действия, декларация пустое_множество в качестве имени обязывает поместить квантор существования в начало. Квантор существования для переменной остается для всей области действия отрицания.

3.2.6 Глаголы и прилагательные

Глаголы и прилагательные могут быть переведены двумя способами: наиболее кратко с помощью *относительного представления* или более полно и детализировано с помощью *ролевого представления*. В относительной форме глагол, например **дает**, представлен трехместным предикатом или отношением, а прилагательное, например **сочная,**

представлен одноместным предикатом. Следующий пример демонстрирует относительное представление предложения, содержащего глагол и прилагательное:

УРЯ: Маша дает собаке сочную кость.

ИП: $(\exists x{:}\text{Собака})(\exists y{:}\text{Кость})$
$(\text{Человек}(\text{Маша}) \wedge \text{Дает}(\text{Маша},x,y) \wedge \text{Сочная}(y))$

Ролевое представление, обычно используемое в лингвистике, переводит **дает** в одноместный предикат типа метка, который связан со своими участниками двухместными отношениями: агент (**Agnt**) для дателя; тема (**Thme**) для подарка; и получатель (**Rcpt**) для получателя. Ролевое представление в ИП:

ИП: $(\exists x{:}\text{Собака})(\exists y{:}\text{Кость})(\exists z{:}\text{Дает})(\exists w{:}\text{Сочная})$
$(\text{Человек}(\text{Маша}) \wedge \text{Agnt}(z,\text{Маша}) \wedge \text{Rcpt}(z,x)$
$\wedge \text{Thme}(z,y) \wedge \text{Attr}(y,w))$

Ролевое представление является более гибким, потому что позволяет добавлять любое число модификаторов в глаголы и прилагательные. Следующее предложение, например, нельзя было бы представить в относительной форме, так как трехместное отношение для **дает** и одноместный предикат для **сочная** не оставляют возможности для дополнительных уточнений.

УРЯ: Маша дает собаке очень сочную кость с горячей сковороды.

Таким образом, УРЯ отлично подходит для языка описания онтологии.

3.3 Реализованные возможности

Для того, чтобы преодолеть трудности, возникающие при построении онтологий, обозначенные в разделе, посвященном актуальности темы работы, было решено использовать «усеченный» русский язык.

В данном разделе дано описание реализованных возможностей. На данный момент можно строить основу онтологии: важнейшие семантические отношения «род - вид», «часть - целое», создавать экземпляры классов, указывать эквивалентные классы и указывать некоторые ограничения.

Опишем контекстно-свободную грамматику «усеченного» русского языка в форме Бэкуса-Наура.

1. <paragarph>::= {<sentence>}
   Параграф – набор разрешенных предложений, описывающих необходимую для построения онтологию.
2. <subject>::=(((‘Любой’|‘Любая’|‘Любое’)<name> | <instance>) (‘ - это’|’ - часть’) <name>)|((‘Любой’|‘Любая’|‘Любое’)’ состоит из ’(‘ровно’|’не менее’|’не более’){<digit>}<name>)(‘Что-то является’<name>’тогда и только тогда, когда оно является’<name>)
   Во второй части предложения после слов ‘ - часть’, ’ состоит из ’(‘ровно’|’не менее’|’не более’ и при определении эквивалентных классов существительное (существительное с модификатором) стоит не в именительном падеже. С помощью модуля pymorphy определяющее существительное ставится в именительный падеж, а также нормализуются прилагательные.
3. <instance> ::= <bigName>
   Экземпляр объекта - любое сочетание прописных и строчных букв, цифр и знака подчеркивания, начинающееся с прописной буквы.
4. <name>::=<smallLetter>{<digit>|<smallLetter>|<bigLetter>|’_’}
   Имя класса - любое сочетание прописных и строчных букв, цифр и знака подчеркивания, начинающееся со строчной буквы.
5. <bigName> ::= <bigLetter>{<digit>|<smallLetter>|<bigLetter>|'_'}
6. <digit> ::= '0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8|'9'

3.4 Стандарт OWL

Semantic Web - это направление развития Всемирной паутины, целью которого является представление информации в виде, пригодном для машинной обработки. Semantic Web будет основываться на способности XML определять настраиваемые схемы разметки и гибкий подход RDF к представлению данных. Первый уровень над RDF, требуемый для Semantic Web - это язык онтологий, который может формально описать значение терминов, используемых в веб-документах. Если, как предполагается, на машины возложить полезные рассудочные задачи в отношении этих документов, то данный язык должен выйти за пределы основной семантики RDF Схем. Документ [18] обеспечивает больше деталей об онтологиях, мотивирует

потребность в языке веб-онтологий, описывая случаи использования, и формулирует цели, требования и правила использования OWL.

OWL был разработан, чтобы удовлетворить эту потребность в языке веб-онтологий. OWL - часть растущего комплекта рекомендаций W3C, связанных с Semantic Web. OWL разработан для использования приложениями, которые должны обрабатывать содержимое информации, а не только представлять эту информацию людям. OWL обеспечивает более полную машинную обработку Веб-контента, чем та, которую поддерживают XML, RDF, и RDF Schema (RDF-S), предоставляя наряду с формальной семантикой дополнительный терминологический словарь. OWL предназначен для тех случаев, когда содержащаяся в документах информация должна быть обработана приложениями, в противоположность ситуациям, где нужно только представить содержимое документов людям. OWL может использоваться, чтобы явно представлять значения терминов и отношения между этими терминами в словарях.

3.5 Пример использования

В работе [11] описан способ построения онтологий сложных технических объектов на примере АНПА – автономного необитаемого подводного аппарата**.** В этой работе онтология строится вручную. В качестве примера приведем построение онтологии такого же автономного необитаемого подводного аппарата, используя описываемую в данной работе технологию.

Используемый для примера текст:

1. Любой носитель - это техническое_устройство.
2. Любой автономный_аппарат - это носитель.
3. Любой буксируемый_аппарат - это носитель.
4. Любой тяжелый_аппарат - это носитель.
5. Любой автономный_необитаемый_подводный_аппарат - это самоходный_необитаемый_подводный_аппарат.
6. Любой гибридный_аппарат - это самоходный_необитаемый_подводный_аппарат.
7. Любой подводный_телеуправляемый_аппарат - это самоходный_необитаемый_подводный_аппарат.
8. Любой носитель - это техническое_устройство.

9. Любой полуавтономный_аппарат - это самоходный_необитаемый_подводный_аппарат.
10. Что-то является системой_энергообеспечения тогда и только тогда, когда оно является источником_тока.
11. AUSS - это автономный_необитаемый_подводный_аппарат.
12. Любое техническое_устройство - часть самоходного_необитаемого_подводного_аппарата.
13. Любой самоходный_необитаемый_подводный_аппарат состоит из не более чем 3 технических_устройств.
14. Любая система_энергообеспечения состоит из ровно 4 технических_устройств.
15. Любая система_снабжения состоит из не менее чем 5 технических_устройств.

Далее показан синтаксис OWL [17] на основе данного примера.

3.5.1 Иерархия классов

В предложениях 1 – 9 определяется иерархия используемых в проектируемой онтологии понятий.

Новый класс описывается так:

<owl:Class rdf:about=”class_name”/>,

где class_name – имя нового класса.

Подкласс объявляется следующим образом:

```
<owl:Class rdf:about="subclass_name">
    <rdfs:subClassOf rdf:resource="class_name"/>
</owl:Class>,
```

где subclass_name – имя класса, стоящего ниже в иерархии, class_name – имя родительского класса.

Таким образом, первое предложение в синтаксисе OWL выглядит так:

<owl:Class rdf:about=” техническое_устройство”/>,

```
<owl:Class rdf:about="носитель">
    <rdfs:subClassOf rdf:resource="техническое_устройство"/>
</owl:Class>
```

Все классы, которые как мы считаем, не являются подклассами в нашей онтологии, в OWL автоматически становятся подклассами встроенного самого общего класса по имени Thing(Вещь), который является классом всех индивидов и суперклассом для всех OWL классов. Есть также встроенный наиболее специфичный класс по

имени Nothing (Ничто), который не имеет никаких представителей и является подклассом всех OWL классов.

На рисунке 1 представлена получившаяся иерархия. Здесь и далее представлены скриншоты из одного из самых популярных на данный момент редакторов онтологий Protégé.

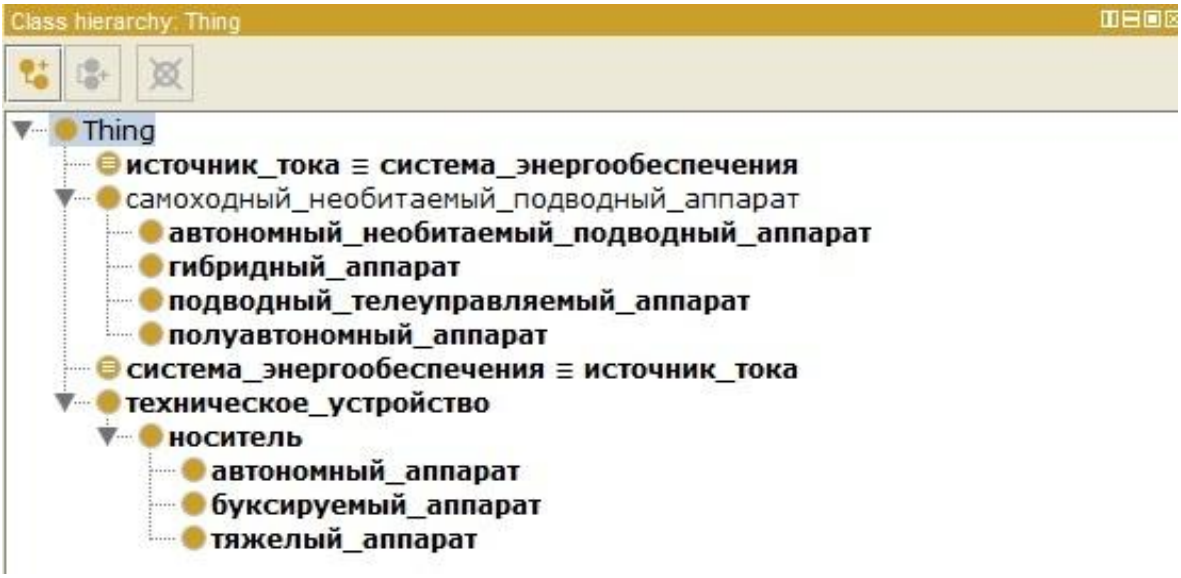

Рис. 1

3.5.2 Эквивалентные классы

В предложении 10 определяются эквивалентные классы.

Синтаксис OWL:

<owl:Class rdf:about="class_name1">

<equivalentClass rdf:resource="class_name2"/>

</owl:Class>,

где class_name1 и class_name1 – имена классов.

Таким образом, мы описываем, что классы имеют одних и тех же представителей.

Это свойство может использоваться для создания синонимичных классов как в нашем примере:

<owl:Class rdf:about="система_энергообеспечения">

<equivalentClass rdf:resource="источник_тока"/>

</owl:Class>

Из этого можно вывести, что любой индивид, который является представителем классы “ система_энергообеспечения ”, является также и представителем класса “ источник_тока” и наоборот.

3.5.3 Экземпляры объектов

В предложении 11 определяется экземпляр объекта класса “автономный_необитаемый_подводный_аппарат ”.

Синтаксис OWL:

<class_name rdf:ID="individual_name">
< class_name >,

где individual_name – имя экземпляра объекта.

Так, в нашем примере получится следующее:

<автономный_необитаемый_подводный_аппарат rdf:ID="AUSS">
</автономный_необитаемый_подводный_аппарат>

Рис. 2

3.5.4 Отношения “часть – целое”

Для определения этих свойств введем два свойства-объекта (ObjectProperty): consistsOf и bePartOf, а также объявим их обратными друг другу:

<owl:ObjectProperty rdf:about="consistsOf"/>
<owl:ObjectProperty rdf:about=" bePartOf">
    <owl:inverseOf rdf:resource="/consistsOf"/>
</owl:ObjectProperty>

Далее, в синтаксисе OWL предусмотрена возможность определения различных ограничений на свойства. В данной работе рассмотрено указание количества элементов – кардинальность. Ограничения кардинальности в OWL известны как локальные ограничения, так как они распространяются на свойства каждого конкретного класса. То

есть, эти ограничения задают кардинальность данного свойства для представителей данного класса.

minCardinality – встроенное свойство OWL, показывающее, что любой представитель данного класса будет связан этим свойством по крайней мере с указанным количеством индивидов.

maxCardinality - встроенное свойство OWL, показывающее, что любой представитель данного класса будет связан этим свойством не более чем с указанным количеством индивидов.

qualifiedCardinality - встроенное свойство OWL, показывающее, что любой представитель данного класса будет связан этим свойством не более чем с указанным количеством индивидов.

Для указания необходимых ограничений для класса используется следующий синтаксис:

```
<owl:Class rdf:about="class_name1">
        <rdfs:subClassOf>
            <owl:Restriction>
               <owl:onProperty rdf:resource="property_name"/>
               <owl:onClass rdf:resource="class_name2"/>
               <owl: cardinality_type
rdf:datatype="&xsd;nonNegativeInteger">number</owl:cardinality_type>
            </owl:Restriction>
          </rdfs:subClassOf>
</owl:Class>
```

где class_name1 – имя класса, на который накладывается ограничение,

class_name2 – имя класса, который является значением,

property_name – название свойства,

number – количество индивидов класса-значения,

cardinality_type – вид ограничения: minCardinality, maxCardinality либо qualifiedCardinality.

В рассматриваемом примере получается следующее:

```
<owl:Class rdf:about=" техническое_устройство">
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource=" bePartOf"/>
         <owl:onClass rdf:resource=" самоходный_необитаемый_подводный_аппарат"/>
```

```
      </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about=" самоходный_необитаемый_подводный_аппарат">
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource=" consistsOf"/>
         <owl:onClass rdf:resource=" техническое_устройство"/>
         <owl:maxQualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">3</owl:maxQualifiedCardinality>
      </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

Рис. 3

```
<owl:Class rdf:about=" система_энергообеспечения">
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource=" consistsOf"/>
         <owl:onClass rdf:resource=" exническое_устройство"/>
         <owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">4</owl:qualifiedCardinality>
      </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

Рис. 4

```
<owl:Class rdf:about=" система_снабжения">
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource=" consistsOf"/>
         <owl:onClass rdf:resource=" техническое_устройство"/>
```

```
        <owl:minQualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">5</owl:minQualifiedCardinality>
      </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

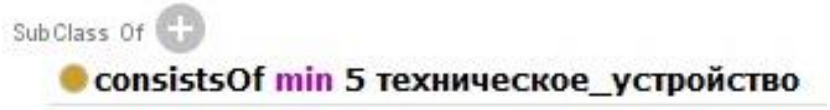

Рис. 5

4. Выводы

Таким образом, мы можем сделать следующие выводы:

1. Анализ существующих методов решения поставленной проблемы показал необходимость создания системы автоматической генерации онтологий из текста на русском языке.
2. Построение достаточно сложного объекта — такого как онтология, кодирующая информацию общей предметной области в различных конфигурациях сервисных систем вполне возможно реализовать через последовательность простых предложений на ограниченном родном естественном языке, построенных в процессе семантизации из исходного корпуса текстов.
3. Качество проведенной семантизации, выражающейся в наборе простых предложений на ограниченном естественном языке, несмотря на кажущуюся простоту, существенно зависит от полноты и других качеств используемого корпуса текстов. А также, как и в случае прямого кодирования через редактор онтологий, от квалификации инженера знаний.
4. Разработанная технология, построенная на наборе простых предложений на ограниченном и специализированном родном естественном языке, оказалась легкой и качественной для быстрого построения базовой онтологии. Интеграция знаний на базе такой технологии также происходит быстрее и качественнее.
5. Разработанную технологию можно легко адаптировать под запросы пользователей.
6. С помощью предложенного метода можно в удобном виде описать любые объекты предметной области и взаимосвязи между ними.

5. Заключение

Результаты работы опубликованы в статье [11] и в тезисах докладов на конференциях [20], [21], [22].

В рамках данной работы был предложен способ автоматической генерации онтологии из текстов на “усеченном” русском языке. Данный способ позволяет полностью описывать предметную область и интегрировать знания в формате OWL, таким образом облегчая построение онтологии конечным пользователем. В дальнейшем планируется расширять возможности данной технологии, добавить возможность проверки получающейся онтологии на целостность и непротиворечивость.

Литература

1. *Gruber T.R.* The role of common ontology in achieving sharable, reusable knowledge bases // Principles of Knowledge Representation and Reasoning. // Proceedings of the Second International Conference. J.A. Allen, R. Fikes, E. Sandewell – eds. Morgan Kaufmann – 1991 - pp. 601-602.

2. *Enderton H.* A Mathematical Introduction to Logic. // New York, Academic Press. - 1972

3. *Gruber, T. R.* Ontolingua: A Mechanism to Support Portable Ontologies. // Knowledge Systems Laboratory Technical Report KSL 91-66, Final Version. Stanford University. - 1992

4. *Gruber, T. R.* A translation approach to portable ontologies. // *Knowledge Acquisition*, 5(2): - 1993, - pp.199-220
5. *Jeff Heflin* OWL Web Ontology Language Use Cases and Requirements // W3C Recommendation – 2004 - http://www.w3.org/TR/2004/REC-webont-req-20040210/
6. *Рыков В.В.* Обработка нечисловой информации. Управление знаниями. – М.: МФТИ, 2008.
7. *Рыков В.В.* Когнитивные технологии в информационных системах – риторическое исчисление
8. *http://habrahabr.ru/post/219243/*
9. *Рыков В. В.* Знания, гуманитарные технологии и поддержка инновационной деятельности.
10. *Рыков В. В.* Пространство понимания и коллективная деятельность
11. *Бронецкий А.Е. Клименко С.В., Рыков В. В., Хламов М. А.* Технологии построения онтологий сложных технических объектов (на примере АНПА – Автономного необитаемого подводного аппарата**)** // Международная конференция «Физико-техническая информатика (СРТ2013). Кипр, Ларнака, 12-19 мая 2013 г. – М.: ИФТИ, 2013.
12. *Гаврилова Т.А.* Спецификация знаний через структурирование: введение в CAKE-технологию // Сборник трудов III конференции по искусственному интеллекту. т. 2. - Тверь. 1992 - с. 113-116.
13. *Гаврилова Т.А. и др.* Визуализация онтологий как инструмент приобретения знаний // Труды 4-го международного семинара по прикладной семиотике, семиотическому и интеллектуальному управлению ASC/IC'99. Москва, 1999 - с. 34-41.
14. *Mescherin S., Kirillov I., Klimenko S.* Leveraging UML and concept maps for constructing crisis management ontology // Proceedings of Cyberworlds 2012 international conference, IEEE Xplore, Washington, 2012.

15. *P. Kaplanski, A. Wroblewska, A. Zieba, P. Zarzycki* Practical applications of controlled natural language with description logics and OWL. FluentEditor and OASE., 2012 - http://www.cognitum.eu/research/publications/Cognitum_Semantics%20_CNL2012.pdf
16. *Новиков А.И.* Семантика текста и ее формализация. - Изд-во "Наука", 1983 - 213 с.
17. *Mike Dean, Guus Schreiber* OWL Web Ontology Language Reference // W3C Recommendation – 2004 - http://www.w3.org/TR/owl-ref/
18. http://www.w3.org/TR/webont-req/
19. *Долганов А.А.* Проблемы формализации и повторного использования накопленных знаний. Дипломная работа. - М.: СТАНКИН, 2010
20. *Составители М. Л. Ремнёва, А. А. Поликарпов, О. В. Кукушкина.* Русский язык: исторические судьбы и современность: V Международный конгресс исследователей русского языка (Москва, МГУ имени М. В. Ломоносова, филологический факультет, 18–21 марта 2014 г.): Труды и материалы / — М.: Изд-во Моск. ун-та, 2014. — 848 с.
21. *Хламов М.А.* Технология построения онтологий сложных технических объектов из их описания на естественном языке. // Труды 56-й научной конференции МФТИ: Всероссийской научной конференции «Актуальные проблемы фундаментальных и прикладных наук в современном информационном обществе», Всероссийской молодежной научно-инновационной конференции «Физико-математические науки: актуальные проблемы и их решения». Управление и прикладная математика. Том 2. - М.:МФТИ, 2013. - 189с.
22. *Хламов М.А.* Возможность построения онтологий из текста на естественном языке. // Конференция “Корпусные технологиии компьютерные методы в современной гуманитарной науке”, Нижний Новгород, 11-12 апреля 2014 г.
23. *Хламов М.А.* Построение онтологий технических объектов из их описания на естественном языке // Русский язык: исторические судьбы и современность. V Международный конгресс исследователей русского языка, Москва, МГУ имени М. В. Ломоносова 18–21 марта 2014 года