ISSN 1814-1196
Научный вестник НГТУ
том 73, № 4, 2018, с. 27–46

http://journals.nstu.ru/vestnik
Science Bulletin of the NSTU
Vol. 73, No. 4, 2018, pp. 27–46

ИНФОРМАТИКА,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И УПРАВЛЕНИЕ

INFORMATICS,
COMPPUTER ENGINEERING
AND CONTROL

УДК 004.85

DOI: 10.17212/1814-1196-2018-4-27-46

О возможностях автоматической обработки текста для построения онтологии требований*

Т.В. АВДЕЕНКО[a], М.Ш. МУРТАЗИНА[b], М.Г. ГРИФ[c]

630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет

[a] *avdeenko@corp.nstu.ru* [b] *murtazina@corp.nstu.ru* [c] *grif@corp.nstu.ru*

Успех программного продукта зависит от того, насколько он, как инструмент решения различных задач, соответствует потребностям конечных пользователей. Именно поэтому инженерия требований как процесс, в рамках которого происходит извлечение, анализ, спецификация и валидация требований, играет ключевую роль в разработке программных продуктов. Трендами исследований последних лет в области совершенствования процесса инженерии требований являются методы работы с требованиями как с фактами из модели предметной области приложения, а также разработка моделей представления знаний о процессах спецификации требований, о типах требований и критериях качества требований. Наиболее подходящей формой представления таких знаний являются онтологии. В данной статье рассматриваются ключевые особенности онтологического подхода к инженерии требований. При этом внимание сфокусировано на возможностях представления знаний о предметной области программного продукта в форме онтологии, и особенно на исследованиях, направленных на автоматизацию данного процесса. Актуальность разработки подходов к автоматизации построения онтологий из текстов, содержащих требования, обусловливается изменчивостью требований со стороны стейкхолдеров и необходимостью быстрого сопоставления текстов требований с целью выявления концептов предметной области программного продукта и анализа соотношения концептов между собой. Основной задачей настоящей работы является изучение возможностей автоматической обработки текста на русском языке для построения онтологий требований. Рассматриваются инструменты автоматической обработки текста на русском языке, такие как ЭТАП-3, ABBYY Compreno, Texterra, Томита-парсера, RML (проект АОТ). На примере лингвистических инструментов RML и ЭТАП-3 анализируются результаты обработки текстов требований на естественном русском языке.

* *Статья получена 04 октября 2018 г.*

Работа поддержана грантом Министерства образования и науки РФ в рамках проектной части государственного задания, проект № 2.2327.2017/4.6 «Интеграция моделей представления знаний на основе интеллектуального анализа больших данных для поддержки принятия решений в области программной инженерии».

The work is supported by the grant of the Russian Ministry of Education and Science within the framework of the project part of the State task, project No 2.2327.2017/4.6 "Integration of Models for Representing Knowledge Based on Intellectual Analysis of Big Data to Support Decision Making in Program Engineering"

Ключевые слова: программная инженерия, инженерия требований, онтология, автоматическая обработка текста, пользовательская история, ЭТАП-3, RML, АОТ

ВВЕДЕНИЕ

В результате стремительного развития новых технологий, а также ужесточения требований со стороны заказчика разработка программного обеспечения (ПО) для производственных и непроизводственных организаций становится все более и более сложной задачей. Чтобы оставаться конкурентоспособными, ИТ-разработчики должны уметь создавать как можно более сложное ПО за меньшее время. Разработка сложного ПО должна удовлетворять большому числу пожеланий и требований различных сторон, часто конфликтующих друг с другом.

Классический жизненный цикл ПО включает такие этапы, как сбор и анализ требований, проектирование, кодирование и отладка, тестирование, эксплуатация и сопровождение. Разработка требований, известная как инженерия требований, является фундаментом всего процесса разработки ПО, определяя в конечном счете его успех. Под инженерией требований понимается процесс извлечения, оценки, спецификации, консолидации и изменения целей, функционала, свойств и ограничений, которым должно обладать разрабатываемое ПО [1]. Процесс инженерии требований может быть условно разбит на четыре этапа: извлечение, анализ, спецификация и валидация требований.

Заказчики и конечные пользователи обычно не знают, что собой представляют требования, и не умеют их формулировать. В большинстве случаев они имеют лишь некоторый образ того, как должно выглядеть ПО и какие цели они могут достигнуть с его помощью. Эта информация пользователей очень важна в инженерии требований, однако ее явно недостаточно. На начальном этапе разработки инженеры требований должны понять намерения (цели) заказчиков и конечных пользователей, в результате чего идентифицировать требования, которые обеспечат выполнение целей. Это первоначальное извлечение требований и их анализ называется «ранней инженерией требований», оказывающей большое влияние на последующий процесс. «Поздняя инженерия требований», к которой относятся спецификация и валидация требований, страдает от ошибок, допущенных при ранней инженерии.

Результат извлечения и анализа требований к программному продукту записывается в форме спецификации требования. Согласно ISO/IEC/IEEE 29148:2011, спецификация требований к программному продукту (software requirements specification) – это структурированный набор требований (функции, производительность, ограничения проектирования и атрибуты) к ПО и его внешним интерфейсам [2]. Спецификация требований является основой для разработки ПО и подготовки к тестированию. В работах [3, 4] сложность работы с требованиями названа в качестве основной причины провала проектов по разработке программного обеспечения. По мнению авторов этих работ, недостаточно хорошо выполненная спецификация требований не позволяет создать ПО, соответствующее потребностям заказчика.

Согласно публикации [5], валидация требований – это процесс, при котором: 1) множество требований является корректным, полным и конси-

стентным; 2) можно создать модель, которой удовлетворяет множество требований; 3) имеется возможность проверки того, что разработанное ПО удовлетворяет множеству требований.

Часто встречается ситуация, когда инженер не может идентифицировать достаточное количество требований для того, чтобы разрабатываемый программный продукт удовлетворял всем пожеланиям заказчика. Обычно в этом случае недостаточно и сопутствующей информации (приоритеты требований, риски, затраты, тест-кейсы), способствующей повышению качества требований. Кроме того, требования часто не связаны друг с другом. Все это ведет к отсутствию консистентности требований, а именно к противоречиям и избыточности.

В настоящее время хорошо известно [6], что отсутствующие, неполные и противоречивые требования приводят к ошибкам в проектировании, реализации и тестировании ПО и в конечном счете к несоответствующему качеству продукта. Превышение запланированного бюджета, срыв сроков реализации могут даже привести к тому, что проект будет прерван. Заказчики могут отказаться от такой системы, а конечные пользователи разочарованы в ней. В работе [7], содержащей обзор публикаций, исследующих причины провала проектов по созданию ПО, делается вывод, что «пять из восьми возможных причин провала базируются на требованиях». Таким образом, совершенствование методов работы с требованиями может существенным образом улучшить качество и безопасность конечного продукта ПО, уменьшить риск превышения сроков и запланированного бюджета, и, что важнее всего, уменьшить или даже исключить риск провала проекта.

Задача извлечения и анализа требований сходна с чрезвычайно сложной задачей извлечения знаний из экспертов при создании интеллектуальных экспертных систем как по своей исключительной важности для дальнейшего процесса, так и по своей сути. Поэтому для ее решения представляется целесообразной разработка средств автоматизированного извлечения формализованных требований из текстов требований на естественном языке на основе методологии искусственного интеллекта. Результатом применения таких методов может стать формальное представление требований в какой-либо модели представления знаний, подобное представлению элементов знаний (аксиом) в базах знаний интеллектуальных систем. Последующая валидация требований в этом случае может базироваться на использовании методов логического вывода для проверки полноты и непротиворечивости системы аксиом, отражающих формальное описание требований.

Актуальной моделью представления сложных знаний в системах искусственного интеллекта являются онтологии. Онтологии используются в инженерии знаний чаще всего для выполнения концептуального моделирования предметной области, где они интерпретируются как «явная спецификация концептуализации» [8]. Таким образом, онтология есть формальное описание объектов и их свойств, отношений, а также ограничений и правил, управляющих отношениями. Онтологии содержат явно определенные и одинаково понимаемые концепты и ограничения, представленные в машиночитаемом формате.

Инженерия требований предполагает явное описание знаний предметной области, распределенных по различным сферам, таким как опыт, функцио-

нальность, нефункциональные требования, стейкхолдеры и т. п. Необходимо сконцентрировать эти разнообразные знания в едином репозитории. Онтологии оказываются эффективной моделью представления, организации и рассуждений о сложных знаниях, какими являются спецификации требований.

Одним из первых проектов, показавших, насколько эффективны могут быть онтологии в области инженерии требований, стали исследования по созданию методологии OntoREM (Ontology-driven Requirements Engineering Methodology), которая была апробирована в компании Airbus для разработки требований к эксплуатационной пригодности воздушных судов [9]. Данные разработки стимулировали целый ряд исследований по использованию онтологического подхода к инженерии требований в целом и инженерии требований к программным продуктам в частности. Так, в работе [10] показывается, что онтологии могут применяться для представления знаний о структуре документов с требованиями, о типах требований и о предметной области программного продукта. В исследовании [11] предлагается применять OWL-онтологии для трассировки требований посредством четырех типов связей: Refines, Requires, Conflicts, and Contains. Эти связи позволяют строить правила для рассуждения о трассируемости, согласованности и полноте требований. В диссертации [12] предлагается подход к автоматизации процесса валидации и измерения знаний о требованиях к ПО. В статьях [13, 14] предлагается подход, основанный на фреймовой онтологии, которая описывает модель типов требований для проектов разработки ПО. В рамках онтологии заданы типы отношений, которые могут быть использованы для проверки свойств непротиворечивости и трассируемости. В работе [15] предлагается применение онтолого-ориентированного подхода к поддержке процесса инженерии требований в Scrum.

Трендом исследований последних лет в области совершенствования инструментария инженерии требований является разработка методов работы с требованиями как с фактами из модели предметной области приложения. Под фактом понимается «эмпирическое знание об объектах, их свойствах и ситуациях, зафиксированное в высказывании» [16, с. 208]. Представление знаний о предметной области программного продукта в виде онтологии позволяет выполнять автоматический поиск противоречий в модели требований. После построения онтологии, отражающей все множество требований на будущий программный продукт со стороны различных стейкхолдеров, проверка качества системы требований (качества спецификации) может проводиться на основе применения логического вывода к сформированной системе аксиом над концептами онтологии, записанных на языке математической логики. Однако само построение онтологии, включающее, помимо общих требований к программному продукту, специфические требования, являющиеся результатом моделирования определенной предметной области, для которой создается продукт, является чрезвычайно сложной задачей. Данная задача по сложности сопоставима с задачей извлечения знаний из экспертов при построении экспертных систем, для решения которой в настоящее время активно развиваются методы машинного обучения.

Обычно требования к ПО фиксируются в виде текстового описания, либо на естественном языке, либо на более ограниченном языке, предполагающем использование более строгого порядка слов и лексики контролируемого

языка, например, в виде пользовательской истории. Построение и актуализация онтологии предметной области ПО с учетом постоянной изменчивости бизнес-требований и требований пользователей является крайне трудоемкой работой, в этой связи активно исследуется задача автоматического построения онтологии [17, 18].

В данной работе рассматриваются основные системы автоматической обработки текста на русском языке и анализируются их возможности для построения онтологии требований. В разделе 1 проводится обзор систем автоматической обработки текста и выделяются подходящие системы, которые потенциально могут быть использованы для построения онтологии. В разделе 2 приводятся результаты обработки текстов требований на естественном русском языке системой RML и дается интерпретация результатов. В разделе 3 мы исследуем возможности обработки спецификации требований системой ЭТАП-3. В заключение авторы делают выводы о возможностях автоматизированного построения онтологии требований с использованием систем автоматической обработки текста.

1. ОБЗОР СИСТЕМ АВТОМАТИЧЕСКОЙ ОБРАБОТКИ ТЕКСТА НА РУССКОМ ЯЗЫКЕ

Автоматическое построение онтологии из текстов требований на естественных языках предполагает извлечение концептов (классов) онтологии, а также отношений над концептами. Применительно к терминологии компьютерной лингвистики отношения над концептами онтологии могут быть отнесены к семантическим отношениям. Отношения над концептами могут быть получены в результате семантического анализа текста, которому предшествуют графематический, морфологический и синтаксический анализ [19].

Семантика как раздел лингвистики занимается изучением значения языковых единиц. Смысл единицы русского языка зависит от ее соотношения с остальными единицами языка, от ее лексической и синтаксической сочетаемости с ними. Так, слова «грамм», «килограмм», «тонна» являются словами одного лексического уровня – единицы измерения, но различны по числовому значению. Одной из приоритетных проблем в процессе автоматической обработки текста является проблема разрешения лексической неоднозначности. В случае неправильного определения смысл может быть полностью искажен. Рассмотрим наиболее распространенные системы семантического анализа русского языка.

Многоцелевой лингвистический процессор ЭТАП-3

ЭТАП-3 (акроним от «Электротехнический автоматический перевод-3») – это система, разработанная научными сотрудниками Института проблем передачи информации им. А.А. Харкевича. Цель программы – это анализ и синтез текстов по модели «Смысл–Текст», созданной И.А. Мельчуком при активном участии А.К. Жолковского и Ю.Д. Апресяна. К появлению экспериментальной системы ЭТАП привели многолетние исследования под руководством академика РАН Ю.Д. Апресяна [20]. В 1974 году были созданы системы машинного перевода ЭТАП-1 (с французского на русский) и ЭТАП-2 (с английского на русский) [21]. Следующим полигоном для испытания положений модели «Смысл–Текст» стала система ЭТАП-3, которая

разрабатывается с 1980-х годов по настоящее время. Правила и словари для анализа и синтеза текстов в системе ЭТАП-3 были частично подготовлены И.А. Мельчуком. С использованием системы ЭТАП-3 к настоящему времени разработаны система машинного перевода, модуль универсального сетевого языка UNL (Universal Networking Language), компьютерный учебник лексики и в полуавтоматическом режиме размечен корпус русских текстов СинТагРус [22].

В системе ЭТАП-3 использовано 78 видов синтаксических отношений [23]. Наиболее интересными для рассмотрения являются лексические функции, которые решают задачи разрешения синтаксической омонимии, разрешения лексической неоднозначности, идиоматического перевода. Пример синтаксической омонимии – «контроль правительства». Первый смысловой вариант – правительство контролирует кого-то, второй – правительство контролируется. Процесс разрешения синтаксической омонимии заключается в определении роли слова «контроль» (осуществлять или быть под, находиться под, подвергаться). В предложении «Президент осуществляет контроль правительства» контролируется правительство. Приведем пример для второго значения: «Президент находится под контролем правительства». Система ЭТАП-3 способна также решить проблему лексической неоднозначности. В таких случаях смысл глагола определяется существительным, с которым глагол образует словосочетание. Корректное значение определяется в случае, когда глагол употреблен в контексте существительного, например: «держать слово», «держать пари», «держать экзамен». Ознакомиться с демонстрационной версией лингвистического процессора ЭТАП-3 может любой желающий на официальном сайте Лаборатории компьютерной лингвистики ИППИ РАН (http://proling.iitp.ru/ru/etap3)

Технология ABBYY Compreno

ABBYY Compreno – это система анализа и понимания текстов на естественном языке, работа которой основана на модели предметной области в форме онтологии, правилах извлечения информации и статистике сочетаемости, которая была собрана разработчиками на корпусах параллельных текстов. Работы над этой системой ведутся более 20 лет. В отличие от других подобных технологий, ABBYY Compreno выполняет полный семантико-синтаксический анализ текста, извлекает сущности, события и связи между ними.

Анализ текста система ABBYY Compreno выполняет в четыре этапа: лексико-морфологический анализ, синтаксический анализ, семантический анализ и прагматический уровень анализа. На первом этапе исходный текст разбивается на абзацы, предложения и слова. Далее для слов определяются лексемы и морфологические признаки. На втором этапе проводится полный синтаксический анализ. На третьем этапе работы системы осуществляется семантический анализ: определяются значения слов, строится семантический граф предложения на основе данных о синтаксическом разборе. Следующий этап заключается в прагматическом уровне анализа. На этом этапе текст анализируется через прагматический слой, применяются онтологии и правила для извлечения нужных объектов. Итоговый результат – это универсальное представление информации, которое позволяет структурировать контент в нужном виде с точки зрения пользователя технологии.

Основным достоинством технологии ABBYY Compreno является определение смысла многозначных слов с помощью разрешения проблемы омонимии. Анализируя текст, программа определяет омонимы и находит корректное значение, полученное в результате анализа контекста. Данное свойство позволяет существенно повышать соответствие результатов поискового запроса, а также точность выявления конкретных объектов в текстах.

ABBYY Compreno используется для анализа сложных лингвистических связей между словами. Определение подобных связей играет важную роль в поисковых и аналитических задачах анализа текста. Благодаря семантико-синтаксическому анализу система ABBYY Compreno способна учесть множество особенностей естественного языка, которые обычно создают препятствия для качественного автоматического определения отношений в текстах. К преимуществам данной технологии относятся точный анализ, быстрый запуск проекта и качественная работа с информацией на русском языке [24]. Система выпускается под коммерческой лицензией, демонстрационная версия предоставляется только по именному запросу от сотрудников компаний.

Технология Texterra

Texterra – это технология многоязычного интеллектуального анализа текста на основе методов обработки текста, которые используют знания, извлекаемые из контента пользователей. Технология Texterra обеспечивает быстрое масштабируемое решение для интеллектуального анализа текста. Технология Texterra позволяет выполнять машинную обработку естественных языков по модели аннотирования текстов, которая аналогична модели, используемой в Apache UIMA [25].

Разработка технологии Texterra была начата в 2007 году Институтом системного программирования им. В.П. Иванникова совместно с компанией Hewlett Packard. В 2010–2013 годах развитие технологии Texterra проходило в рамках сотрудничества с компанией Samsung, дальнейшее развитие Texterra связано со стартовавшим в 2012 году исследовательским проектом Talisman (Tracking And Learning Insights form Social Media ANalysis), направленным на создание технологии, способной отследить и выделить фиктивные аккаунты в социальных сетях [26, 27] .

Технология Texterra предназначена для получения знаний на основе источников, находящихся в открытом доступе, таких как «Википедия», «Викиданные» и «МедиаВики». В качестве решаемых задач с использованием данной технологии приводятся следующие: анализ отзывов пользователей социальных медиа с целью мониторинга репутации людей, организаций и товаров, семантический поиск документов и автоматическое построение предметно-специфичных баз знаний. В основе технологии Texterra лежат методы извлечения данных и методы компьютерной лингвистики. На верхнем уровне технологии Texterra находится четыре модуля: модуль лингвистического анализа, модуль базы знаний, модуль извлечения информации и модуль анализа эмоциональной окраски. Технология Texterra применяется для решения задач семантического поиска и построения баз знаний. В системе реализованы возможности распознавания именованных сущностей, привязки к понятиям базы знаний фрагментами текстов, определяющих семантику, и извлечения основных понятий из текста. Ознакомиться с возможностями технологии Texterra можно на сайте Института системного программирования им. В.П. Иванникова РАН (https://texterra.ispras.ru/)

Томита-парсер

Томита-парсер – это анализатор текста на естественном языке, который позволяет извлекать из текста факты (структурированные данные). Томита-парсер использует алгоритм GLR-парсинга. Грамматики Томита-парсера работают с цепочками. Цепочка – это одно предложение. Из цепочки выделяются подцепочки, которые, в свою очередь, интерпретируются в разбитые по полям факты. Основные компоненты парсера [28]:

- газеттир – словарь ключевых слов, используемый в процессе анализа контекстно-свободными грамматиками;
- грамматика – множество правил, записанных на языке контекстно-свободных грамматик, определяющих синтаксическую структуру выделяемых из текста цепочек;
- множество описаний типов фактов, порождаемых в результате отображения синтаксической структуры во множество линейно организованных фактов.

Томита-парсер распространяется бесплатно, но без правил грамматики. В качестве входа для извлечения данных из фразы на естественном языке используется текстовый файл в кодировке «Юникод» (UTF-8). Результаты извлечения фактов можно получить в формате XML, что удобно для их последующей обработки.

Пакет лингвистических инструментов RML

RML (акроним от «рабочее место лингвиста») – это пакет лингвистических инструментов для анализа текста на английском, русском и немецком языках. В настоящее время пакет лингвистических инструментов RML реализуется коллективом авторов под руководством А.В. Сокирко. Пакет лингвистических инструментов RML с 2010 года доступен под лицензией GNU Public Licence [29]. Разработка проекта была начата в компании «Диалинг» (г. Москва) в конце 1990-х годов. Теоретические принципы анализа текста в системе «Диалинг» были основаны на многолетних исследованиях Н.Н. Леонтьевой. В 1998 году Н.Н. Леонтьева, получив предложение о сотрудничестве от президента компании, собрала команду для разработки системы коммерческого русско-английского машинного перевода «Диалинг». В системе «Диалинг» основополагающим семантическим понятием стало понятие «семантическое отношение». Под семантическим отношением авторы системы «Диалинг» понимали некую универсальную связь, усматриваемую носителем языка в тексте. Семантическое отношение записывается как R(A,B), где R – название отношения, A – зависимый член, B – управляющий член семантического отношения [30]. Например, отношение для фразы «картина Шишкина» будет иметь вид *автор* (*Шишкин, картина*).

Проект по разработке системы «Диалинг» просуществовал до 2001 года и был закрыт как нерентабельный. После закрытия проекта «Диалинг» А.В. Сокирко вместе с единомышленниками продолжил работу над лингвистическими процессорами системы «Диалинг» [31]. С этого момент проект развивался над названием АОТ (акроним от «автоматическая обработка текста»). АОТ включает такие компоненты, как графематический, морфологический, синтаксический и семантический анализаторы. Русский морфологический словарь проекта, как и многие современные морфологические словари,

базируется на грамматическом словаре А.А. Зализняка. Морфологический словарь русского языка проекта АОТ в настоящее время, помимо самого проекта, используется как основа словарей в других лингвистических проектах, например, в проекте по созданию размеченного корпуса текстов OpenCorpora. Данный корпус доступен под лицензией CC-BY-SA [32]. В проекте OpenCorpora словарь АОТ был адаптирован для задач контроля качества разметки корпуса: произведена унификация решеток парадигм и уменьшено количество лемм-омонимов [33].

Таким образом, в этом разделе мы рассмотрели ряд инструментов автоматической обработки текста на русском языке: ЭТАП-3, ABBYY Compreno, Texterra, Томита-парсер, RML. Для дальнейшего исследования были выбраны ЭТАП-3 и RML. Выбор первого обусловлен тем, что это инструмент, который был использован для разметки русских текстов СинТагРус. Данный корпус после проверки людьми стал использоваться инструментами автоматической обработки текста, чьи алгоритмы основаны на машинном обучении. Выбор второго, в свою очередь, обусловлен тем, что данный пакет полностью находится в свободном доступе и его морфологический словарь русского языка был использован как основа морфологической разметки текстового корпуса OpenCorpora.

2. ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ RML ДЛЯ ИЗВЛЕЧЕНИЯ СЕМАНТИЧЕСКИХ ОТНОШЕНИЙ ОНТОЛОГИИ ТРЕБОВАНИЙ

Для иллюстрации возможностей системы RML рассмотрим примеры разбора двух предложений, содержащих требования. Первое предложение записано с учетом рекомендаций, что текст документов с требованиями должен состоять из коротких и ясных предложений. Второе требование сформулировано с применением хорошо зарекомендовавшей себя в гибких методологиях разработки техники пользовательских историй.

Требование 1. *По запросу руководства диспетчер должен отправить отчет о состоянии системы.*

Требование 2. *Как инженер, я хочу видеть год постройки каждой котельной, чтобы планировать ремонтные работы.*

Для обработки этих предложений были использованы синтаксические и семантические анализаторы системы RML. На рис. 1 и 2 показаны результаты анализа поверхностного синтаксиса для *Требования 1,* а на рис. 3 и 4 – для *Требования 2*.

Из рис. 1–4 видно, что для *Требования 1*, сформулированного в форме простого предложения, была определена одна клауза (элементарное предложение, вершиной которого является глагол либо или элемент, заменяющий его). Для предложения *Требование 2,* которое представляет собой сложноподчиненное предложение, определено две клаузы. Тип вершины клаузы для предложения *Требование 1* обозначен типом «КР_ПРИЛ» (краткая форма прилагательного). Для предложения *Требование 2* вершина главной клаузы имеет тип «ГЛ_ЛИЧН» (личная форма глагола), вершина подклаузы – «ИНФ» (неопределенная форма глагола).

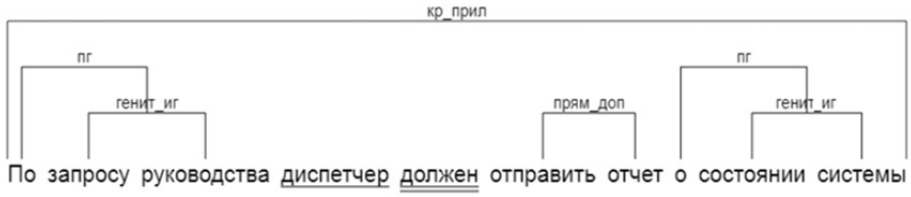


Рис. 1. Результаты анализа поверхностного синтаксиса предложения в виде системы составляющих, полученные на сайте aot.ru (*Требование 1*)

Fig. 1. Results of the shallow parsing of the sentence in the form of a system of components obtained on the site aot.ru (*Requirement 1*)

```
- <sent>
  - <synvar>
      <clause type="КР_ПРИЛ">По запросу руководства диспетчер должен отправить отчет о
        состоянии системы .</clause>
      <group type="ГЕНИТ_ИГ">запросу руководства</group>
      <group type="ПГ">По запросу руководства</group>
      <group type="ПРЯМ_ДОП">отправить отчет</group>
      <group type="ГЕНИТ_ИГ">состоянии системы</group>
      <group type="ПГ">о состоянии системы</group>
    </synvar>
    <rel grmchld="но,ср,рд,ед," lemmchld="РУКОВОДСТВО" grmprnt="но,мр,дт,ед," lemmprnt="ЗАПРОС"
      gramrel="но,мр,дт,ед," name="ГЕНИТ_ИГ"> запросу -> руководства </rel>
    <rel grmchld="но,мр,дт,ед," lemmchld="ЗАПРОС" grmprnt="" lemmprnt="ПО" gramrel="дт," name="ПГ">
      По -> запросу </rel>
    <rel grmchld="но,мр,вн,ед," lemmchld="ОТЧЕТ" grmprnt="дст,пе,св," lemmprnt="ОТПРАВИТЬ"
      gramrel="дст,пе,св," name="ПРЯМ_ДОП"> отправить -> отчет </rel>
    <rel grmchld="но,жр,рд,ед," lemmchld="СИСТЕМА" grmprnt="но,ср,пр,ед," lemmprnt="СОСТОЯНИЕ"
      gramrel="но,ср,пр,ед," name="ГЕНИТ_ИГ"> состоянии -> системы </rel>
    <rel grmchld="но,ср,пр,ед," lemmchld="СОСТОЯНИЕ" grmprnt="" lemmprnt="О" gramrel="пр,"
      name="ПГ"> о -> состоянии </rel>
    <rel grmchld="од,мр,им,ед," lemmchld="ДИСПЕТЧЕР" grmprnt="кач,но,од,мр,ед,"
      lemmprnt="ДОЛЖЕН" gramrel="" name="ПОДЛ"> должен -> диспетчер </rel>
  </sent>
```

Рис. 2. Синтаксический разбор предложения (*Требование 1*), выполненный системой RML

Fig. 2. The syntactic analysis of the sentence (*Requirement 1*) made by the RML system

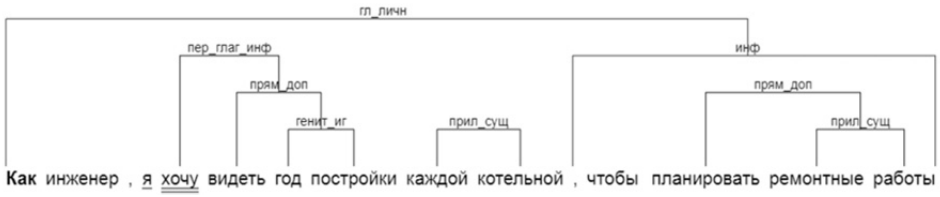


Рис. 3. Результаты анализа поверхностного синтаксиса в виде системы составляющих, полученные на сайте aot.ru (*Требование 2*)

Fig. 3. Results of the shallow parsing of the sentence in the form of a system of components obtained on the site aot.ru (*Requirement 2*)

Результаты синтаксического разбора передаются семантическому анализатору, который строит семантическую структуру каждого поданного на вход предложения. Семантическая структура состоит из семантических узлов и семантических отношений. Под семантическим узлом понимается «объект текстовой семантики, у которого заполнены все валентности, как эксплицитно выраженные в тексте, так и имплицитные – те, которые получаются из экстралингвистических источников» [31]. При анализе для каждой «жесткой группы»

или одиночного слова создается узел с морфологическими характеристиками. Предлоги становятся атрибутами узлов, которыми они управляли. Далее анализируются выделенные синтаксические отношения. Подробно с процедурой анализа можно ознакомиться на официальном сайте проекта АОТ на странице «Первичный семантический анализ» (http://www.aot.ru/docs/seman.html).

```
- <sent>
   - <synvar>
        <clause type="ИНФ">, чтобы планировать ремонтные работы .</clause>
        <group type="ПРИЛ_СУЩ">ремонтные работы</group>
        <group type="ПРЯМ_ДОП">планировать ремонтные работы</group>
     </synvar>
   - <synvar>
        <clause type="ГЛ_ЛИЧН">Как инженер , я хочу видеть год постройки каждой котельной ,
            чтобы планировать ремонтные работы .</clause>
        <group type="ГЕНИТ_ИГ">год постройки</group>
        <group type="ПРЯМ_ДОП">видеть год постройки</group>
        <group type="ПЕР_ГЛАГ_ИНФ">хочу видеть год постройки</group>
        <group type="ПРИЛ_СУЩ">каждой котельной</group>
     </synvar>
     <rel grmchld="кач,но,вн,мн," lemmchld="РЕМОНТНЫЙ" grmprnt="но,жр,вн,мн,"
        lemmprnt="РАБОТА" gramrel="вн,мн," name="ПРИЛ_СУЩ"> работы -> ремонтные </rel>
     <rel grmchld="но,жр,вн,мн," lemmchld="РАБОТА" grmprnt="дст,пе,нс," lemmprnt="ПЛАНИРОВАТЬ"
        gramrel="дст,пе,нс," name="ПРЯМ_ДОП"> планировать -> работы </rel>
     <rel grmchld="но,од,жр,пр,тв,дт,рд,ед," lemmchld="КАЖДЫЙ" grmprnt="но,жр,пр,тв,дт,рд,ед,"
        lemmprnt="КОТЕЛЬНАЯ" gramrel="жр,пр,тв,дт,рд,ед," name="ПРИЛ_СУЩ"> котельной ->
        каждой </rel>
     <rel grmchld="дст,пе,нс," lemmchld="ВИДЕТЬ" grmprnt="дст,пе,нс,1л,нст,ед," lemmprnt="ХОТЕТЬ"
        gramrel="дст,пе,нс,1л,нст,ед," name="ПЕР_ГЛАГ_ИНФ"> хочу -> видеть </rel>
     <rel grmchld="но,жр,рд,ед," lemmchld="ПОСТРОЙКА" grmprnt="но,мр,вн,ед," lemmprnt="ГОД"
        gramrel="но,мр,вн,ед," name="ГЕНИТ_ИГ"> год -> постройки </rel>
     <rel grmchld="но,мр,вн,ед," lemmchld="ГОД" grmprnt="дст,пе,нс," lemmprnt="ВИДЕТЬ"
        gramrel="дст,пе,нс," name="ПРЯМ_ДОП"> видеть -> год </rel>
     <rel grmchld="1л,им,ед," lemmchld="Я" grmprnt="дст,пе,нс,1л,нст,ед," lemmprnt="ХОТЕТЬ"
        gramrel="" name="ПОДЛ"> хочу -> я </rel>
     <rel grmchld="дст,пе,нс," lemmchld="ПЛАНИРОВАТЬ" grmprnt="дст,пе,нс,1л,нст,ед,"
        lemmprnt="ХОТЕТЬ" gramrel="" name="ПОДКЛАУЗА"> хочу -> планировать </rel>
  </sent>
```

Рис. 4. Синтаксический разбор предложения (*Требование 2*), выполненный системой RML

Fig. 4. The syntactic analysis of the sentence (*Requirement 2*) made by the RML system

Результаты работы семантического анализатора для предложений *Требование 1* и *Требование 2* представлены на рис. 5 и 6.

```
Nodes:
  Node 0 По ЗАПРОСУ: ЗАПРОС С л но,мр,дт,ед, -> С л но,мр,дт,ед,
  Node 1 РУКОВОДСТВА: РУКОВОДСТВО С л но,ср,рд,ед, -> С л но,ср,рд,ед,
  Node 2 ДИСПЕТЧЕР: ДИСПЕТЧЕР С л од,мр,им,ед, -> С л од,мр,им,ед,
  Node 3 ДОЛЖЕН: ДОЛЖЕН КР_ПРИЛ  кач,но,од,нст,мр,ед, -> КР_ПРИЛ  кач,но,од,нст,мр,ед,
  Node 4 ОТПРАВИТЬ: ОТПРАВИТЬ ИНФИНИТИВ  дст,пе,св, -> ИНФИНИТИВ  дст,пе,св,
  Node 5 ОТЧЕТ: ОТЧЕТ С л но,мр,вн,ед, -> С л но,мр,вн,ед,
  Node 6 о СОСТОЯНИИ: СОСТОЯНИЕ С л но,ср,пр,ед, -> С л но,ср,пр,ед,
  Node 7 СИСТЕМЫ: СИСТЕМА С л но,жр,рд,ед, -> С л но,жр,рд,ед,
Relations:
  SUB (ДИСПЕТЧЕР, ДОЛЖЕН) = подл (2, 3)
  CONTEN (ОТПРАВИТЬ, ДОЛЖЕН) = п_доп (4, 3)
  OBJ (ОТЧЕТ, ОТПРАВИТЬ) = п_доп (5, 4)
  THEME (СОСТОЯНИИ, ОТЧЕТ) = к_доп (6, 5)
  ACT (СИСТЕМЫ, СОСТОЯНИИ) = к_доп (7, 6)
  LOK (ЗАПРОСУ, ДОЛЖЕН) = X! (0, 3)
  AGENT (РУКОВОДСТВА, ЗАПРОСУ) = к_доп (1, 0)
  SUB (ДИСПЕТЧЕР, ОТПРАВИТЬ) =  (2, 4)
Aux Relations:
  SUB (ДИСПЕТЧЕР, ОТПРАВИТЬ) =  (2, 4)
```

Рис. 5. Семантические отношения в предложении (*Требование 1*), выявленные системой RML

Fig. 5. Semantic relations in a sentence (*Requirement 1*) detected by the RML system

```
Nodes:
  Node 0 КАК: КАК Н ст вопр, -> Н ст вопр,
  Node 1 ИНЖЕНЕР: ИНЖЕНЕР С л од,мр,им,ед, -> С л од,мр,им,ед,
  Node 2 Я: Я МС  1л,им,ед, -> МС  1л,им,ед,
  Node 3 ХОЧУ: ХОТЕТЬ Г  дст,пе,нс,1л,нст,ед, -> Г  дст,пе,нс,1л,нст,ед,
  Node 4 ВИДЕТЬ: ВИДЕТЬ ИНФИНИТИВ  дст,пе,нс, -> ИНФИНИТИВ  дст,пе,нс,
  Node 5 ГОД: ГОД С л но,мр,вн,ед, -> С л но,мр,вн,ед,
  Node 6 ПОСТРОЙКИ: ПОСТРОЙКА С л но,жр,рд,ед, -> С л но,жр,рд,ед,
  Node 7 КАЖДОЙ: КАЖДЫЙ МС-П т но,од,жр,пр,тв,дт,рд,ед, -> МС-П т но,жр,пр,тв,дт,рд,ед,
  Node 8 КОТЕЛЬНОЙ: КОТЕЛЬНАЯ С л но,жр,пр,тв,дт,рд,ед, -> С л но,жр,рд,ед,
  Node 9 ЧТОБЫ: ЧТОБЫ СОЮЗ зг  -> СОЮЗ зг
  Node 10 ПЛАНИРОВАТЬ: ПЛАНИРОВАТЬ ИНФИНИТИВ  дст,пе,нс, -> ИНФИНИТИВ  дст,пе,нс,
  Node 11 РЕМОНТНЫЕ: РЕМОНТНЫЙ П  кач,но,вн,мн, -> П  кач,но,вн,мн,
  Node 12 РАБОТЫ: РАБОТА С л но,жр,вн,мн, -> С л но,жр,вн,мн,
Relations:
  SUB (Я, ХОЧУ) = подл (2, 3)
  CONTEN (ВИДЕТЬ, ХОЧУ) = к_доп (4, 3)
  BELNG (ПОСТРОЙКИ, КОТЕЛЬНОЙ) = к_доп (6, 8)
  PROPERT (КАЖДОЙ, КОТЕЛЬНОЙ) = ПРИЛ_СУЩ (7, 8)
  METHOD (КАК, ХОЧУ) = Х! (0, 3)
  CONTEN (ГОД, ВИДЕТЬ) = п_доп (5, 4)
  OBJ (РАБОТЫ, ПЛАНИРОВАТЬ) = п_доп (12, 10)
  PURP (РЕМОНТНЫЕ, РАБОТЫ) = ПРИЛ_СУЩ (11, 12)
  SUB (Я, ВИДЕТЬ) =  (2, 4)
Aux Relations:
  SUB (Я, ВИДЕТЬ) =  (2, 4)
```

Рис. 6. Семантические отношения в предложении (*Требование 2*), выявленные системой RML

Fig. 6. Semantic relations in the sentence (*Requirement 2*) detected by the RML system

Из данных на рис. 6 видно, что для части узлов семантические отношения не выделяются. Например, не определено отношение для узла «ИНЖЕНЕР». По результатам проведенных разборов предложений можно отметить, что чем сложнее структура предложения и чем больше в предложении знаков пунктуации, тем меньше выявляется семантических отношений. Рассмотрим результаты, полученные для требования, выраженного предложением с несколькими однородными членами.

Требование 3. *Информация о котельной включает идентификатор котельной (ID), год постройки, средний расход угля, средний расход электроэнергии, средний расход холодной воды, температуру подачи горячей воды*.

В предложении *Требование 3* выделено всего четыре семантических отношения (рис. 7).

```
Nodes:
  Node 0 ИНФОРМАЦИЯ: ИНФОРМАЦИЯ С л но,жр,им,ед, -> С л но,жр,им,ед,
  Node 1 о КОТЕЛЬНОЙ: КОТЕЛЬНАЯ С л но,жр,пр,ед, -> С л но,жр,пр,ед,
  Node 2 ВКЛЮЧАЕТ: ВКЛЮЧАТЬ Г  дст,пе,нс,3л,нст,ед, -> Г  дст,пе,нс,3л,нст,ед,
  Node 3 ИДЕНТИФИКАТОР: ИДЕНТИФИКАТОР С л но,мр,вн,ед, -> С л но,мр,вн,ед,
  Node 4 КОТЕЛЬНОЙ: КОТЕЛЬНАЯ С л но,жр,рд,ед, -> С л но,жр,рд,ед,
  Node 5 ID: ID С л но,зв,пр,тв,вн,дт,рд,им, -> С л но,зв,пр,тв,вн,дт,рд,им,
  Node 6 ГОД: ГОД С л но,мр,вн,им,ед, -> С л но,мр,вн,им,ед,
  Node 7 ПОСТРОЙКИ: ПОСТРОЙКА С л но,жр,рд,ед, -> С л но,жр,рд,ед,
  Node 8 СРЕДНИЙ: СРЕДНИЙ П  но,од,мр,вн,им,ед, -> П  но,од,мр,вн,им,ед,
  Node 9 РАСХОД: РАСХОД С л но,мр,вн,им,ед, -> С л но,мр,вн,им,ед,
  Node 10 УГЛЯ: УГОЛЬ С л но,мр,рд,ед, -> С л но,мр,рд,ед,
  Node 11 СРЕДНИЙ: СРЕДНИЙ П  но,од,мр,вн,им,ед, -> П  но,од,мр,вн,им,ед,
  Node 12 РАСХОД: РАСХОД С л но,мр,вн,им,ед, -> С л но,мр,вн,им,ед,
  Node 13 ЭЛЕКТРОЭНЕРГИИ: ЭЛЕКТРОЭНЕРГИЯ С л но,жр,рд,ед, -> С л но,жр,рд,ед,
  Node 14 СРЕДНИЙ: СРЕДНИЙ П  но,од,мр,вн,им,ед, -> П  но,од,мр,вн,им,ед,
  Node 15 РАСХОД: РАСХОД С л но,мр,вн,им,ед, -> С л но,мр,вн,им,ед,
  Node 16 ХОЛОДНОЙ: ХОЛОДНЫЙ П  кач,но,од,жр,пр,тв,дт,рд,ед, -> П  кач,но,од,жр,пр,тв,дт,рд,ед,
  Node 17 ВОДЫ: ВОД С л но,мр,вн,им,мн, -> С л но,мр,вн,им,мн,
  Node 18 ТЕМПЕРАТУРУ: ТЕМПЕРАТУРА С л но,жр,вн,ед, -> С л но,жр,вн,ед,
  Node 19 ПОДАЧИ: ПОДАЧА С л но,жр,рд,ед, -> С л но,жр,рд,ед,
  Node 20 МИА: ГОРЯЧЕЙ: ГОРЯЧИЙ П  кач,сравн,но,од,жр,рд,ед, -> П  кач,сравн,но,од,жр,рд,ед,
  Node 21 ВОДЫ: ВОДА С л но,жр,рд,ед, -> С л но,жр,рд,ед,
Relations:
  PROPERT (СРЕДНИЙ, РАСХОД) = ПРИЛ_СУЩ (8, 9)
  PROPERT (СРЕДНИЙ, РАСХОД) = ПРИЛ_СУЩ (11, 12)
  PROPERT (СРЕДНИЙ, РАСХОД) = ПРИЛ_СУЩ (14, 15)
  PROPERT (МИА: ГОРЯЧЕЙ, ВОДЫ) = ПРИЛ_СУЩ (20, 21)
```

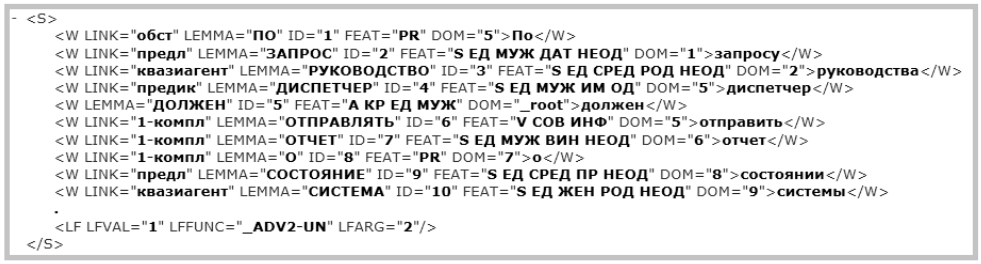
Рис. 7. Семантические отношения в предложении (*Требование 3*), выявленные системой RML

Fig. 7. Semantic relations in the sentence (Requirement 3) detected by the RML system

Таким образом, можно заключить, что для целей извлечения семантических отношений для онтологии требований рассматриваемый программный инструментарий подходит лишь частично, так как он может пропускать значимые понятия. При использовании системы RML потребуется добавлять дополнительные правила обработки токенов, чтобы не потерять значимые понятия и отношения, а также дополнительные правила обработки результатов синтаксического анализа, чтобы выявить отношения «пропущенных» значимых понятий.

3. ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ СИСТЕМЫ ЭТАП-3 ДЛЯ ИЗВЛЕЧЕНИЯ СЕМАНТИЧЕСКИХ ОТНОШЕНИЙ ОНТОЛОГИИ ТРЕБОВАНИЙ

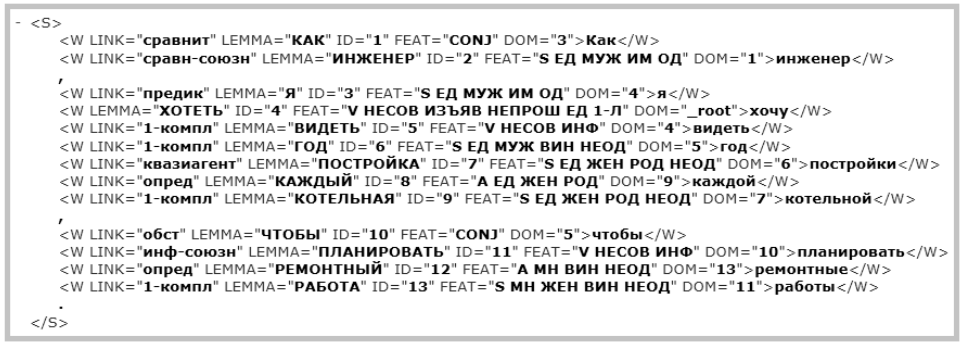
Рассмотрим возможности системы ЭТАП-3 при разборе предложений *Требование 1* и *Требование 2*, которые приведены на рис. 8 и 9.



```
- <S>
    <W LINK="обст" LEMMA="ПО" ID="1" FEAT="PR" DOM="5">По</W>
    <W LINK="предл" LEMMA="ЗАПРОС" ID="2" FEAT="S ЕД МУЖ ДАТ НЕОД" DOM="1">запросу</W>
    <W LINK="квазиагент" LEMMA="РУКОВОДСТВО" ID="3" FEAT="S ЕД СРЕД РОД НЕОД" DOM="2">руководства</W>
    <W LINK="предик" LEMMA="ДИСПЕТЧЕР" ID="4" FEAT="S ЕД МУЖ ИМ ОД" DOM="5">диспетчер</W>
    <W LEMMA="ДОЛЖЕН" ID="5" FEAT="A КР ЕД МУЖ" DOM="_root">должен</W>
    <W LINK="1-компл" LEMMA="ОТПРАВЛЯТЬ" ID="6" FEAT="V СОВ ИНФ" DOM="5">отправить</W>
    <W LINK="1-компл" LEMMA="ОТЧЕТ" ID="7" FEAT="S ЕД МУЖ ВИН НЕОД" DOM="6">отчет</W>
    <W LINK="1-компл" LEMMA="О" ID="8" FEAT="PR" DOM="7">о</W>
    <W LINK="предл" LEMMA="СОСТОЯНИЕ" ID="9" FEAT="S ЕД СРЕД ПР НЕОД" DOM="8">состоянии</W>
    <W LINK="квазиагент" LEMMA="СИСТЕМА" ID="10" FEAT="S ЕД ЖЕН РОД НЕОД" DOM="9">системы</W>
    .
    <LF LFVAL="1" LFFUNC="_ADV2-UN" LFARG="2"/>
  </S>
```


Рис. 8. Разбор синтаксической структуры предложения (*Требование 1*), выполненный системой ЭТАП-3

Fig. 8. Analysis of the syntactic structure of the sentence (*Requirement 1*) made by the ETAP-3 system



```
- <S>
    <W LINK="сравнит" LEMMA="КАК" ID="1" FEAT="CONJ" DOM="3">Как</W>
    <W LINK="сравн-союзн" LEMMA="ИНЖЕНЕР" ID="2" FEAT="S ЕД МУЖ ИМ ОД" DOM="1">инженер</W>
    ,
    <W LINK="предик" LEMMA="Я" ID="3" FEAT="S ЕД МУЖ ИМ ОД" DOM="4">я</W>
    <W LEMMA="ХОТЕТЬ" ID="4" FEAT="V НЕСОВ ИЗЪЯВ НЕПРОШ ЕД 1-Л" DOM="_root">хочу</W>
    <W LINK="1-компл" LEMMA="ВИДЕТЬ" ID="5" FEAT="V НЕСОВ ИНФ" DOM="4">видеть</W>
    <W LINK="1-компл" LEMMA="ГОД" ID="6" FEAT="S ЕД МУЖ ВИН НЕОД" DOM="5">год</W>
    <W LINK="квазиагент" LEMMA="ПОСТРОЙКА" ID="7" FEAT="S ЕД ЖЕН РОД НЕОД" DOM="6">постройки</W>
    <W LINK="опред" LEMMA="КАЖДЫЙ" ID="8" FEAT="A ЕД ЖЕН РОД" DOM="9">каждой</W>
    <W LINK="1-компл" LEMMA="КОТЕЛЬНАЯ" ID="9" FEAT="S ЕД ЖЕН РОД НЕОД" DOM="7">котельной</W>
    ,
    <W LINK="обст" LEMMA="ЧТОБЫ" ID="10" FEAT="CONJ" DOM="5">чтобы</W>
    <W LINK="инф-союзн" LEMMA="ПЛАНИРОВАТЬ" ID="11" FEAT="V НЕСОВ ИНФ" DOM="10">планировать</W>
    <W LINK="опред" LEMMA="РЕМОНТНЫЙ" ID="12" FEAT="A МН ВИН НЕОД" DOM="13">ремонтные</W>
    <W LINK="1-компл" LEMMA="РАБОТА" ID="13" FEAT="S МН ЖЕН ВИН НЕОД" DOM="11">работы</W>
    .
  </S>
```


Рис. 9. Разбор синтаксической структуры предложения (*Требование 2*), выполненный системой ЭТАП-3

Fig. 9. Analysis of the syntactic structure of the sentence (*Requirement 2*) made by the ETAP-3 system

В результате, который возвращает система ЭТАП-3, сохраняется порядок слов в предложении. Система ЭТАП-3 строит дерево зависимостей, уз-

лами которого являются слова предложения, а ветвями – имена синтаксических отношений. Благодаря наличию большого числа типов синтаксических отношений предложение любой сложности может быть представлено в виде дерева зависимостей, некоторые из них встречаются достаточно редко. В частности, в проанализированном фрагменте спецификации требований было выявлено 23 типа синтаксических отношений. Таким образом, можно заключить, что для целей извлечения семантических отношений для онтологии требований результаты обработки текста применимы при разработке системы правил конвертации токенов и синтаксических отношений в классы и свойства онтологии.

Таким образом, можно заключить, что в качестве входных данных для построения онтологии по текстам требований более подходят результаты, полученные с помощью системы ЭТАП-3. Результат ее включает семантические роли в синтаксических отношениях, он более очевиден и предсказуем, а это облегчит задачу разработки набора продукционных правил для обработки результатов работы парсера, особенно если в правилах обработки результатов требований учитывать фиксированную структуру требования (например, в виде пользовательской истории). Такой подход, возможно, позволит получить результаты достаточно высокого качества.

ЗАКЛЮЧЕНИЕ

В данной работе были исследованы возможности автоматической обработки текста на русском языке для целей инженерии требований. Методы, лежащие в основе инструментов автоматической обработки текста, делятся на две группы: основанные на правилах и основанные на машинном обучении. Работа инструментов первой группы основывается на контекстно-свободной грамматике и расширяющих ее теориях с целью описать такие явления, как эллипсис (намеренный пропуск слов) или разрывные именные группы. Подходы, основанные на правилах, предполагают создание набора правил, отражающих все возможные грамматические зависимости. В рамках данной работы были детально изучены два основных представителя лингвистического анализатора, использующих для анализа правила грамматического разбора. На практике предусмотреть все ситуации не представляется возможным, поэтому возможны ситуации, когда лингвистический парсер первой группы для некоторых слов предложения не выдаст никого результата, что мы и пронаблюдали при анализе текстов требований в системе RML.

Инструменты второй группы обучаются на размеченном корпусе текстов, поэтому не требуют ручного написания правил грамматики. На основе данных о выявленных закономерностях в обучающей выборке такие инструменты обрабатывают новые тексты. Корпус может подготавливаться вручную или полуавтоматически, как это было сделано с корпусом СинТагРус при помощи системы ЭТАП-3. Разметка текстового корпуса даже в полуавтоматическом режиме требует значительных трудозатрат, поэтому исторически первыми появились подходы, основанные на правилах. В настоящее время уже проделана огромная работа по созданию размеченных корпусов. Поэтому инструменты, основанные на машинном обучении, начинают выходить на первый план. К числу наиболее эффективных парсеров второй

группы относится универсальный языконезависимый инструмент для работы с деревьями зависимостей MaltParser. Модель русского языка для MaltParser обучена на корпусе СинТагРус, который, в свою очередь, был подготовлен с использованием системы ЭТАП-3. В связи с вышеперечисленным и легкостью применения самого MaltParser он также может быть использован для получения исходных данных для построения онтологии требований.

СПИСОК ЛИТЕРАТУРЫ

1. *Lamsweerde A.* Reasoning about alternative requirements options // Conceptual modeling: foundations and applications. – Berlin: Springer, 2009. – P. 380–397. – (Lecture notes in computer science; vol. 5600). – doi: 10.1007/978-3-642-02463-4_20.

2. ISO/IEC/IEEE 29148:2011. Systems and software engineering. Life cycle processes. Requirements engineering. – [S. l.]: IEEE, 2011. – 83 p.

3. *Leffingwell D., Widrig D.* Managing software requirements: a use case approach. – Boston: Addison-Wesley, 2003. – 544 p.

4. *Wiegers K., Beatty J.* Software requirements. – 3rd ed. – Redmond, WA: Microsoft Press, 2013. – 637 p.

5. *Bahill A.T., Henderson S.J.* Requirements development, verification, and validation exhibited in famous failures // Systems Engineering. – 2005. – Vol. 8, N 1. – P. 1–14. – doi: 10.1002/sys.20017.

6. *Kott A., Peasant J.* Representation and management of requirements: the RAPID-WS Project // Concurrent Engineering. – 2005. – Vol. 3, N 2. – P. 93–106. – doi: 10.1177/1063293X9500300203.

7. *Alexander I., Stevens R.* Writing better requirements. – Boston: Addison-Wesley, 2002. – 159 p.

8. *Gruber T.R.* A translation approach to portable ontology specifications // Knowledge Acquisition. – 1993. – Vol. 5. – P. 199–220. – doi: 10.1006/knac.1993.1008.

9. Ontology-driven requirements engineering: building the OntoREM meta model / M. Kossmann, R. Wong, M. Odeh, A. Gillies // Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications. – Damascus, Syria, 2008. – P. 1–6. – doi: 10.1109/ICTTA.2008.4530315.

10. The use of ontologies in requirements engineering / V. Castañeda, L. Ballejos, M.L. Caliusco, M.R. Galli // Global Journal of Research in Engineering. – 2010. – Vol. 10, iss. 6. – P. 2–8.

11. *Goknil A., Kurtev I., Berg K. van den*. A metamodeling approach for reasoning about requirements // Model Driven Architecture – Foundations and Applications: 4th European Conference, ECMDA-FA 2008, Berlin, Germany, June 9–13, 2008: proceedings. – Berlin; Heidelberg: Springer, 2008. – P. 310–325.

12. *Siegemund K.* Contributions to ontology-driven requirements engineering: diss. to obtain the academic degree Doctoral engineer (Dr.-Ing.). – Dresden: Technischen Universität Dresden, 2014. – 236 p.

13. *Пустовалова Н.В., Авдеенко Т.В.* Построение согласованной модели требований для процесса программной инженерии // Труды СПИИРАН. – 2016. – Вып. 1 (44). – С. 31–49. – doi: 10.15622/sp.44.3.

14. *Avdeenko T.V., Pustovalova N.V.* The ontology-based approach to support the requirements engineering process // 13th International Scientific-Technical Conference on Actual problems of Electronic Instrument Engineering (APEIE-2016): proceedings, Novosibirsk, 3–6 October 2016. – Novosibirsk: NSTU, 2016. – Vol. 1, N 2. – P. 513–518.

15. *Муртазина М.Ш., Авдеенко Т.В.* Онтологический подход к поддержке процесса инженерии требований в Scrum // Сборник трудов ИТНТ-2018: IV Международная конференция и молодежная школа «Информационные технологии и нанотехнологии», Самара, 24–27 апреля 2018 г. – Самара: Новая техника, 2018. – С. 2610–2620.

16. Модели и методы построения информационных систем, основанных на формальных, логических и лингвистических подходах / И.С. Ануреев, Т.В. Батура, О.И. Боровикова, Ю.А. Загорулько, И.С. Кононенко, А.Г. Марчук, П.А. Марчук, Ф.А. Мурзин, Е.А. Сидорова, Н.В. Шилов; отв. ред. А.Г. Марчук. – Новосибирск: Изд-во СО РАН, 2009. – 330 с.

17. Automated extraction of conceptual models from user stories via NLP / M. Robeer, G. Lucassen, J.M.E.M. van der Werf, F. Dalpiaz, S. Brinkkemper // 2016 IEEE 24th International Requirements Engineering (RE) Conference: proceedings. – Beijing, 2016. – P. 196–205. – doi: 10.1109/RE.2016.40.

18. *Assawamekin N., Sunetnanta T., Pluempitiwiriyawej C.* Ontology-based multiperspective requirements traceability framework // Knowledge and Information Systems. – 2010. – Vol. 25, N 3. – P. 493–522. – doi: 10.1007/s10115-009-0259-2.

19. Автоматическая обработка текстов на естественном языке и анализ данных: учебное пособие / Е.И. Большакова, К.В. Воронцов, Н.Э. Ефремова, Э.С. Клышинский, Н.В. Лукашевич, А.С. Сапин. – М.: Изд-во НИУ ВШЭ, 2017. – 269 с.

20. Слово и язык: сборник статей к восьмидесятилетию академика Ю.Д. Апресяна / отв. ред.: И.М. Богуславский, Л.Л. Иомдин, Л.П. Крысин. – М.: Языки славянских культур, 2011. – 735 с.

21. Система машинного перевода «Кросслятор 2.0» и анализ ее функциональности для задачи трансляции знаний / В.А. Галактионов, А.М. Мусатов, О.Ю. Мансурова, С.В. Ёлкин, Э.С. Клышинский, В.Ю. Максимов, С.Н. Аминева, Р.В. Жирнов, С.Ю. Игашов, Т.Н. Мусаева. – М.: б. и., 2007. – 27 с.

22. Многоцелевой лингвистический процессор ЭТАП-3 [Электронный ресурс]. – URL: http://iitp.ru/ru/ru/researchlabs/922.htm (дата обращения: 10.12.2018).

23. Development of a dependency treebank for Russian and its possible applications in NLP / I. Boguslavsky, I. Chardin, S. Grigorieva, N. Grigoriev, L. Iomdin, L. Kreidlin, N. Frid // Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002). – Las Palmas, 2002. – Vol. 3. – P. 852–856.

24. ABBYY Intelligent Search SDK [Электронный ресурс]. – URL: https://www.abbyy.com/ru-ru/isearch/compreno/ (дата обращения: 10.12.2018).

25. Texterra: a framework for text analysis / D.Yu. Turdakov, N.A. Astrakhantsev, Ya.R. Nedumov, A.A. Sysoev, I.A. Andrianov, V.D. Mayorov, D.G. Fedorenko, A.V. Korshunov, S.D. Kuznetsov // Programming and Computer Software. – 2014. – Vol. 40, iss. 5. – P. 288–295.

26. Texterra и анализ текстов [Электронный ресурс]. – URL: https://morphs.ru/posts/2017/03/18/texterra (дата обращения: 10.12.2018).

27. *Турдаков Д.* Анализ социальных сетей: охота на ботов и троллей [Электронный ресурс]. – URL: https://www.osp.ru/netcat_files/userfiles/TBD_1_2017/Turdakov_TBD.pdf (дата обращения: 28.08.2018).

28. Томита-парсер [Электронный ресурс]. – URL: https://tech.yandex.ru/tomita/ (дата обращения: 10.12.2018).

29. Seman [Electronic resource]. – URL: https://sourceforge.net/projects/seman/ (accessed: 10.12.2018).

30. *Сокирко А.В*. Семантические словари в автоматической обработке текста (по материалам системы ДИАЛИНГ): дис. … канд. техн. наук: 05.12.17. – М., 2001. – 88 с.

31. Автоматическая обработка текста [Электронный ресурс]. – URL: http://aot.ru/ (дата обращения: 10.12.2018).

32. Открытый корпус [Электронный ресурс]. – URL: http://opencorpora.org/ (дата обращения: 10.12.2018).

33. Quality assurance tools in the OpenCorpora project / V. Bocharov, S. Bichineva, D. Granovsky, N. Ostapuk, M. Stepanova // Компьютерная лингвистика и интеллектуальные технологии: материалы ежегодной Международной конференции «Диалог» (Бекасово, 25–29 мая 2011 г.). – М.: Изд-во РГГУ, 2011. – С. 101–109.

Авдеенко Татьяна Владимировна, доктор технических наук, профессор кафедры теоретической и прикладной информатики, ведущий научный сотрудник Новосибирского государственного технического университета. Основные направления научных исследований: представление знаний, машинное обучение, математическое моделирование. Имеет более 150 научных публикаций. E-mail: avdeenko@corp.nstu.ru

Муртазина Марина Шамильевна, кандидат филосовских наук, аспирант, старший научный сотрудник Новосибирского государственного технического университета. Основные направления научных исследований: инженерия знаний, инженерия требований, управление в социально-экономических системах. Имеет более 40 научных публикаций. E-mail: murtazina@corp.nstu.ru

Гриф Михаил Геннадьевич, доктор технических наук, профессор кафедры автоматизированных систем управления Новосибирского государственного технического университета. Основные направления научных исследований: оптимальное проектирование человеко-машинных систем, интеллектуальный анализ текста, системы компьютерного сурдоперевода. Имеет более 300 научных публикаций. E-mail: grif@corp.nstu.ru

Avdeenko Tatiana Vladimirovna, D. Sc. (Eng.), professor at the department of theoretical and applied informatics, chief research worker, Novosibirsk State Technical University. Her research interests are knowledge representation, machine learning, and mathematical modeling. She is the author of more than 150 scientific publications. E-mail: avdeenko@corp.nstu.ru

Murtazina Marina Shamil'evna, PhD (Philosophy), a postgraduate student, a leading research worker, Novosibirsk State Technical University. Her research interests are focused on knowledge engineering, requirements engineering, management in economic and social systems. She is the author of more than 40 scientific publications. E-mail: murtazina@corp.nstu.ru

Grif Mikhail Gennadyevich, D. Sc. (Eng.), professor at the department of automated Conctrol systems, Novosibirsk State Technical University. His research interests include an optimal design of man-machine systems, text mining, and computer-generated finger-speech systems. He is the author of more than 300 scientific publications. E-mail: grif@corp.nstu.ru

DOI: 10.17212/1814-1196-2018-4-27-46

On the possibilities of automatic text processing for constructing the ontology of requirements[*]

T.V. AVDEENKO[a], M.SH. MURTAZINA[b], M.G. GRIF[c]

Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russia

[a] avdeenko@corp.nstu.ru [b] murtazina@corp.nstu.ru [c] grif@corp.nstu.ru

Abstract

The success of a software product depends on how well it meets the needs of end users as a tool for solving various problems. That is why requirements engineering, as a process in which the elicitation, analysis, specification and validation of the requirements take place, plays a key role in the development of software products. The trend of recent studies in the field of improving the process of requirements engineering is the development of methods for working with the requirements as with facts from the application domain model and the development of models for the representation of knowledge about requirements specification processes, re-

[*] *Received 04 October 2018.*

The reported study was funded by Russian Ministry of Education and Science, according to the research project No. 2.2327.2017/4.6.

quirements types, requirements quality criteria. The most suitable form of such complex knowledge representation is ontology. The present article discusses the key features of the ontological approach to the requirements engineering. Basic attention is focused on the possibilities of representing knowledge about the application domain of a software product in the form of ontology and, in particular, on automating this process. The relevance of developing the approaches to automatization of building the ontologies from natural text requirements is determined by the variability of the requirements from the stakeholders' part and the need for a quick comparison of the texts of requirements in order to identify concepts of the application domain and to analyze the relationships between the concepts. The main objective of this paper is to study the possibilities of automatic Russian text processing to build the requirements ontologies. We consider tools for automatic Russian text processing, such as ETAP-3, ABBYY Compreno, Texterra, Tomita-parser, RML (the AOT project). Using an example of the linguistic tools RML and STAGE-3, the results of processing the text of requirements in natural Russian have been analyzed.

Keywords: software engineering, requirements engineering, ontology, text processing, user story, ETAP-3, RML, AOT

REFERENCES

1. Lamsweerde A. Reasoning about alternative requirements options. *Conceptual modeling: foundations and applications*. Berlin, Springer, 2009. *Lecture Notes in Computer Science*, vol. 5600, pp. 380–397. doi: 10.1007/978-3-642-02463-4_20.

2. ISO/IEC/IEEE 29148:2011. *Systems and software engineering. Life cycle processes. Requirements engineering*. IEEE, 2011. 83 p.

3. Leffingwell D., Widrig D. *Managing software requirements: a use case approach*. Boston, Addison-Wesley, 2003. 544 p.

4. Wiegers K., Beatty J. *Software requirements*. Redmond, WA, Microsoft Press, 2013. 637 p.

5. Bahill A.T., Henderson S.J. Requirements development, verification, and validation exhibited in famous failures. *Systems Engineering*, 2005, vol. 8, no. 1, pp. 1–14. doi: 10.1002/sys.20017.

6. Kott A., Peasant J. Representation and management of requirements: the RAPID-WS project. *Concurrent Engineering*, 2005, vol. 3, no 2, pp. 93–106. doi: 10.1177/1063293X9500300203.

7. Alexander I., Stevens R. *Writing better requirements*. Boston, Addison-Wesley, 2002. 159 p.

8. Gruber T.R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 1993, vol. 5, pp. 199–220. doi: 10.1006/knac.1993.1008.

9. Kossmann M., Wong R., Odeh M., Gillies A. Ontology-driven requirements engineering: building the OntoREM meta model. *Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications*, Damascus, Syria, 2008, pp. 1–6. doi: 10.1109/ICTTA.2008.4530315.

10. Castañeda V., Ballejos L., Caliusco M.L., Galli M.R. The use of ontologies in requirements engineering. *Global Journal of Research in Engineering*, 2010, vol. 10, iss. 6, pp. 2–8.

11. Goknil A., Kurtev I., Berg K. van den. A metamodeling approach for reasoning about requirements. *Model Driven Architecture – Foundations and Applications*: 4th European Conference, ECMDA-FA 2008: proceedings, Berlin, 2008, pp. 310–325.

12. Siegemund K. *Contributions to ontology-driven requirements engineering.* Dr.-Ing sci. diss. Dresden, Technischen Universität Dresden, 2014. 236 p.

13. Avdeenko T.V., Pustovalova N.V. Postroenie soglasovannoi modeli trebovanii dlya protsessa programmnoi inzhenerii [Building a harmonized model of requirements for software development process]. *Trudy SPIIRAN – SPIIRAS Proceedings*, 2016, iss. 1 (44), pp. 31–49. doi: 10.15622/sp.44.3. (In Russian).

14. Avdeenko T.V., Pustovalova N.V. The ontology-based approach to support the requirements engineering process. *13th International Scientific-Technical Conference on Actual problems of*

Electronic Instrument Engineering (APEIE-2016): proceedings, Novosibirsk, 2016, vol. 1, no 2, pp. 513–518. doi; 10.1109/APEIE.2016.7806406.

15. Murtazina M.Sh., Avdeenko T.V. [The ontology-driven approach to support the requirements engineering process in scrum framework]. *Sbornik trudov ITNT-2018: IV Mezhdunarodnaya konferentsiya i molodezhnaya shkola "Informatsionnye tekhnologii i nanotekhnologii"* [Proceedings of ITNT-2018. IV International Conference on Information Technology and Nanotechnology]. Samara, New technology Publ., 2018, pp. 2610–2620. (In Russian).

16. Anureev I.S., Batura T.V., Borovikova O.I., Zagorul'ko Yu.A., Kononenko I.S., Marchuk A.G., Marchuk P.A., Murzin F.A., Sidorova E.A., Shilov N.V. *Modeli i metody postroeniya informatsionnykh sistem, osnovannykh na formal'nykh, logicheskikh i lingvisticheskikh podkhodakh* [Models and methods of building information systems based on formal, logical and linguistic approaches]. Novosibirsk, SB RAS Publ., 2009. 330 p.

17. *Robeer M., Lucassen G., Werf J.M.E.M. van der, Dalpiaz F., Brinkkemper S.* Automated extraction of conceptual models from user stories via NLP. *2016 IEEE 24th International Requirements Engineering (RE) Conference: proceedings*, Beijing, 2016, pp. 196–205. doi: 10.1109/RE.2016.40.

18. Assawamekin N., Sunetnanta T., Pluempitiwiriyawej C. Ontology-based multiperspective requirements traceability framework. *Knowledge and Information Systems*, 2010, vol. 25, no. 3, pp. 493–522. doi: 10.1007/s10115-009-0259-2.

19. Bol'shakova E.I., Vorontsov K.V., Efremova N.E., Klyshinskii E.S., Lukashevich N.V., Sapin A.S. *Avtomaticheskaya obrabotka tekstov na estestvennom yazyke i analiz dannykh* [Automatic processing of natural language texts and data analysis]. Moscow, NIU VShE Publ., 2017. 269 p.

20. Boguslavskii I.M., Iomdin L.L., Krysin L.P., ed. *Slovo i yazyk: sbornik statei k vos'midesyatiletiyu akademika Yu.D. Apresyana* [Word and language. Collection of articles on the eightieth anniversary of Academician Yu.D. Apresyan]. Moscow, Yazyki slavyanskikh kul'tur Publ., 2011. 735 p.

21. Galaktionov V.A., Musatov A.M., Mansurova O.Yu., Elkin S.V., Klyshinskii E.S., Maksimov V.Yu., Amineva S.N., Zhirnov R.V., Igashov S.Yu., Musaeva T.N. *Sistema mashinnogo perevoda "Krosslyator 2.0" i analiz ee funktsional'nosti dlya zadachi translyatsii znanii* [The machine-translation system "Krossljator 2.0" and analysis of its functionality for the knowledge translation problem]. Moscow, 2007. 27 p.

22. *Mnogotselevoi lingvisticheskii protsessor ETAP-3* [The multifunctional ETAP-3 linguistic processor]. Available at: http://iitp.ru/ru/ru/researchlabs/922.htm (accessed 10.12.2018).

23. Boguslavsky I., Chardin I., Grigorieva S., Grigoriev N., Iomdin L., Kreidlin L., Frid N. Development of a dependency treebank for Russian and its possible applications in NLP. *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, 2002, vol. 3, pp. 852–856.

24. *ABBYY Intelligent Search SDK*. (In Russian). Available at: https://www.abbyy.com/ru-ru/isearch/compreno/ (accessed 10.12.2018).

25. Turdakov D.Yu., Astrakhantsev N.A., Nedumov Ya.R., Sysoev A.A., Andrianov I.A., Mayorov V.D., Fedorenko D.G., Korshunov A.V., Kuznetsov S.D. Texterra: a framework for text analysis. *Programming and Computer Software*, 2014, vol. 40, iss. 5, pp. 288–295.

26. *Texterra i analiz tekstov* [Texterra and text analysis]. Available at: https://morphs.ru/posts/2017/03/18/texterra (accessed 10.12.2018).

27. Turdakov D. *Analiz sotsial'nykh setei: okhota na botov i trollei* [Analysis of social networks: hunting for bots and trolls]. Available at: https://www.osp.ru/netcat_files/userfiles/TBD_1_2017/Turdakov_TBD.pdf (accessed 10.12.2018).

28. *Tomita-parser* [Tomita parser]. Available at: https://tech.yandex.ru/tomita/ (accessed 10.12.2018).

29. *Seman*. Available at: https://sourceforge.net/projects/seman/ (accessed 10.12.2018).

30. Sokirko A.V. *Semanticheskie slovari v avtomaticheskoi obrabotke teksta (po materialam sistemy DIALING)*. Diss. kand. tekhn. nauk [Semantic dictionaries in automatic text processing (based on the DIALING system). PhD eng. sci. diss.]. Moscow, 2001. 88 p.

31. *Avtomaticheskaya obrabotka teksta* [Text processing]. Available at: http://aot.ru/ (accessed 10.12.2018).

32. *Otkrytyi korpus* [Open Corpora]. Available at: http://opencorpora.org/ (accessed 10.12.2018).

33. Bocharov V., Bichineva S., Granovsky D., Ostapuk N., Stepanova M. Quality assurance tools in the OpenCorpora project. *Komp'yuternaya lingvistika i intellektual'nye tekhnologii*: materialy ezhegodnoi Mezhdunarodnoi konferentsii "Dialog" [Proceedings of the 17th International conference on computational linguistics and intellectual technologies]. Moscow, 2011, pp. 101–109.

Для цитирования:

Авдеенко Т.В., Муртазина М.Ш., Гриф М.Г. О возможностях автоматической обработки текста в инженерии требований // Научный вестник НГТУ. – 2018. – № 4 (73). – С. 27–46. – doi: 10.17212/1814-1196-2018-4-27-46.

For citation:

Avdeenko T.V., Murtazina M.Sh., Grif M.G. O vozmozhnostyakh avtomaticheskoi obrabotki teksta v inzhenerii trebovanii [On the possibilities of automatic text processing for constructing the ontology of requirements]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2018, no. 4 (73), pp. 27–46. doi: 10.17212/1814-1196-2018-4-27-46.