

ИНФОРМАТИКА,  
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА  
И УПРАВЛЕНИЕ

INFORMATICS,  
COMPPUTER ENGINEERING  
AND MANAGEMENT

УДК 004.82

DOI: 10.17212/1814-1196-2019-4-99-120

## Контролируемый естественный язык для работы с онтологиями\*

В.И. ХАБАРОВ<sup>1,a</sup>, И.С. СТЕПАНОВ<sup>2,b</sup>, А.А. СЕРЕНКО<sup>3,c</sup>

<sup>1</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет

<sup>2</sup> 630049, РФ, г. Новосибирск, ул. Дуси Ковальчук, 191, Сибирский государственный университет путей сообщения

<sup>3</sup> 630090, РФ, г. Новосибирск, пр. академика Лаврентьева, 6, Институт вычислительной математики и математической геофизики Сибирского отделения РАН

<sup>a</sup> [khabarov51@mail.ru](mailto:khabarov51@mail.ru)   <sup>b</sup> [step\\_il@mail.ru](mailto:step_il@mail.ru)   <sup>c</sup> [fafnur@yandex.ru](mailto:fafnur@yandex.ru)

Работа содержит описание контролируемого естественного языка, предназначенного для работы с онтологиями. Благодаря формализованному определению понятийной структуры предметной сферы в онтологии становится возможной машинная обработка онтологических сведений, применяемая в новых поисковиках, системах информационной интеграции, экспертных системах.

Приводится краткий анализ аналогов, а также обоснование необходимости разработки специализированного языка. Систематизированы требования, предъявляемые к языку с учетом области применения. Описано позиционирование разрабатываемого языка по отношению к другим представителям группы онтологических языков. В настоящей работе предлагается один из возможных вариантов контролируемого языка, изначально ориентированный на русскоязычного пользователя. Однако, как показала практика, мультиязычный вариант представления оказался возможным и весьма полезным для целей профессионального обучения в связи с тем, что структура многих языков может быть представлена достаточно единообразно. Поскольку контролируемый естественный язык является подмножеством естественного языка, он не противоречит синтаксическим правилам русского языка, а также его лексемы семантически соответствуют. В свою очередь, естественный язык является более выразительным и сложным. При этом контролируемый естественный язык должен однозначно отображаться в стандартный язык представления онтологий OWL. Язык OWL базируется на дескриптивной логике (DL), ввиду этого достигается трехстороннее соответствие языков.

Дано описание структуры, синтаксиса и грамматики языка, представление дескриптивной логики и правил в конструкциях языка. Приведены примеры представления знаний в описываемом формате в сравнении с аналогами. Описано использование контролируемого естественного языка для составления повествования с поддержкой онтологического формата. В рамках разработки языка некоторые конструкции были упрощены относительно англоязычного аналога, а также адаптированы к специфике русского языка.

---

\* Статья получена 03 сентября 2019 г.

Эффективность процесса управления знаниями напрямую зависит от формы их представления. Широкое распространение онтологий в целях организации контента открывает новые возможности для работы с семантикой, заключенной в текстах. Контролируемый естественный язык является двунаправленным, воспринимаемым как человеком, так и машиной логического вывода. Его использование позволит найти более широкое применение онтологиям, а также снизить порог входа для работы с ними.

**Ключевые слова:** контролируемый естественный язык, язык описания онтологий, онтологии, инженерия знаний, база знаний, представление знаний, механизм логического вывода, редактор онтологий, мультязычность, семантика, трансляторы

## ВВЕДЕНИЕ

Ограниченный естественный язык обычно используется для лаконичного и точного представления знания в конкретной профессиональной предметной области. Например, ограниченный язык профессиональной прозы используется при написании научных статей, технической документации, справочников, учебников и пр. Однако, имея определенные ограничения на стиль и на используемый словарь понятий, такой язык не обладает достаточной для однозначного машинного представления точностью выражения смысла. В связи с этим возникает потребность в разработке контролируемого естественного языка (КЕЯ) с более жесткими ограничениями на синтаксис и семантику. Это дает возможность, с одной стороны, компилировать тексты на КЕЯ в машинные формы представления знаний, а с другой стороны, естественно воспринимать их человеком без какой-либо специальной подготовки [1].

КЕЯ открывает новые возможности привлечения специалистов из различных предметных областей для создания современных систем, основанных на технологиях обработки знаний. Контролируемый естественный язык по существу является формой представления онтологий, которые в настоящее время широко используются как способ профессиональных коммуникаций, инструмент для поиска информации, средство представления образовательного контента, форма представления знаний для систем искусственного интеллекта. КЕЯ способствует не только повышению уровня машинного восприятия информации, но и снижению порога входа для людей, работающих со знаниями.

Следует отметить, что КЕЯ зависит от используемого естественного языка. В англоязычных странах контролируемый английский язык широко применяется для создания контента, семантика которого однозначно воспринимается машинами. Наиболее ярким примером является ACEWIKI. Реализация, применение и решение типовых задач с использованием контролируемых языков на примере данного проекта подробно представлены в [2]. Наиболее популярные на сегодняшний день редакторы онтологий, такие как, например, Protege и Top Braid Composer, требуют определенной подготовки пользователя с точки зрения работы с интерфейсами данных программ и, что более важно, компетенции в инженерии знаний. Использование КЕЯ для работы пользователей со знаниями, в частности с такой формой их представления, как онтологии, нашли свою реализацию в продукте Fluent Editor 2015. Интерфейс данной программы позволяет использовать привычные англоязычные конструкции для описания предметной области [3].

В русскоязычном сегменте на сегодняшний день ситуация в области разработки КЕЯ выглядит достаточно скромно. Можно, пожалуй, указать всего один ресурс, позволяющий создавать базы знаний на контролируемом русском языке [4]. Но предлагаемый язык не предназначен для неподготовленных пользователей. Для сравнения, существует более ста версий контролируемого английского языка, которые нашли прикладное применение и развитие.

На данный момент разработка в области инструментария по работе с онтологиями ведется в направлении графических редакторов и текстовых [5]. Таким образом, задача разработки КЕЯ для русскоязычного ресурса с контентом в формате онтологий является актуальной. С акцентом на образование онтологический формат представления знаний является своевременным в связи с нарастающей тенденцией цифровой трансформации данной отрасли [6].

В настоящей работе предлагается один из возможных вариантов КЕЯ, изначально ориентированный на русскоязычного пользователя. Однако, как показала практика, мультиязычный вариант представления оказался возможным и весьма полезным для целей профессионального обучения в связи с тем, что структура многих языков может быть представлена достаточно единообразно.

## 1. ТРЕБОВАНИЯ, ПРЕДЪЯВЛЯЕМЫЕ К КЕЯ

Контролируемый естественный язык (КЕЯ) в данной работе относится к классу онтологических языков (ОЯ). К числу основных требований, предъявляемых к онтологическим языкам, например в [7], отнесены следующие требования:

- четко определенный синтаксис;
- эпистемологическая адекватность, то есть способность выразить все понятия и отношения в данной предметной области;
- использование логики как основы для языка (логика первого порядка, дескриптивная логика, модальная логика);
- достаточная выразительная сила, способствующая восприятию текста человеком;
- строгость кодирования для однозначной машинной интерпретации понятий, связей, аксиом;
- возможность выполнять логический вывод, основываясь на структурных компонентах языка;
- функциональная полнота, необходимая для решения различных задач при операциях с онтологиями: построение и совместное использование онтологий, обмен знаниями, взаимодействие с онтологиями;
- однозначная интерпретация смысла, выраженного в знаниевых сущностях, не только создателями онтологии, но и внешними агентами (пользователями, программами и т. п.).

Сформулируем дополнительные требования, которым должен удовлетворять КЕЯ:

- КЕЯ является подмножеством естественного языка;
- КЕЯ должен быть однозначно отображаться в OWL [8].

Поскольку КЕЯ является подмножеством естественного языка, он не противоречит синтаксическим правилам ЕЯ, а также его лексемы семантически соответствуют ЕЯ. В свою очередь, естественный язык является более выразительным и сложным. При этом КЕЯ должен однозначно отображаться в стандартный язык представления онтологий OWL. Язык OWL базируется на дескриптивной логике (DL), ввиду этого достигается трехстороннее соответствие языков (рис. 1).

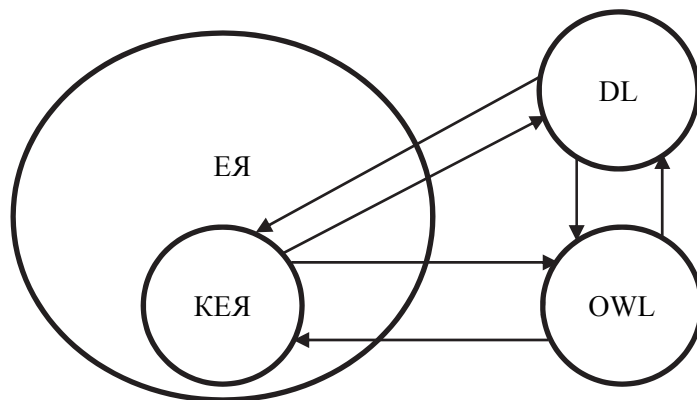


Рис. 1. Отношение между языками OWL, DL, ЕЯ и КЕЯ

Fig. 1. Relations between the OWL, DL, ЕЯ and КЕЯ languages

## 2. СТРУКТУРА ЯЗЫКА

Любой язык состоит из набора понятий и связей между ними. Для удобства восприятия, а также в целях соответствия современным стандартам представления знаний [9] все понятия были разделены на три базовых класса:

- объект – сущность, содержащая информацию о понятии, как минимум название и идентификатор;
- отношение – сущность, отвечающая за связь между понятиями. В OWL данный класс представлен как *ObjectProperty*;
- свойство – сущность, отвечающая за связь между понятиями и их характеристиками. В OWL данный класс представлен как *DataProperty*.

Стоит отметить, что отношение и свойство являются подклассами объекта, но, чтобы не усложнять структуру онтологии, они были вынесены в отдельные иерархии. В общем случае каждый из классов имеет свои отношения и свойства, как минимум родовидовые отношения и лингвистические свойства.

В основе языка лежит триплетная запись. Важно учитывать, что, несмотря на изначальную простоту данной структуры, с помощью нее можно выстраивать довольно сложные языковые конструкции благодаря тому, что в качестве субъектов и объектов могут выступать не только понятия, но и сами триплеты. И вложенность эта ограничивается только возможностями человеческого восприятия и вычислительными затратами. Наиболее распространенным применением данной конструкции является запись правил, где в каче-

стве отношения выступает причинно-следственная связь, а в качестве субъекта и объекта – триплет, содержащий условие выполнения правила, и триплет, содержащий следствие его выполнения соответственно.

Структура данной конструкции представлена на рис. 2.

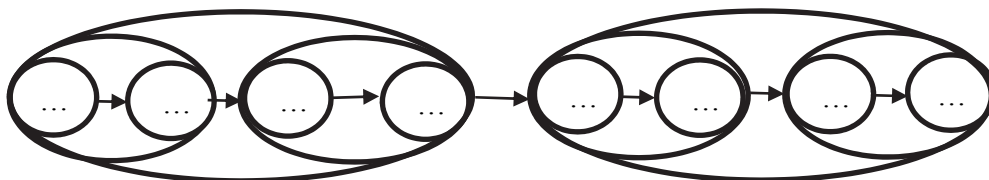


Рис. 2. Структура вложенности триплетных записей

Fig. 2. A structure of triplet record nesting

Кроме того, данная структура соответствует фреймовому формату представления знаний [10].

### 3. БАЗОВЫЕ КЛАССЫ ОТНОШЕНИЙ

При реализации онтологий ключевую роль играют родовидовые отношения между понятиями. В большинстве языков описания онтологий для данного отношения выделяют специальные языковые конструкции. В КАЯ Every <C> is a <D>, в OWL используется стандартное отношение SubClassOf. Но ввод ограничений на формат записи родовидового отношения существенно снижает выразительность языка описания предметной области. Кроме того, данный подход требует ввода этого отношения для каждого из языков представления (русского, китайского, английского).

Для рассматриваемого языка выработан следующий подход: для выделенных отношений вводится не служебная конструкция языка, а отдельный класс, который содержит все отношения, содержащие в себе родовидовую семантику или семантику отношения часть–целое, а также обратные им.

Таким образом, при работе с данным языком нет ограничений в выразительных средствах при вводе базовых отношений. Возможно использование любых синонимов при условии, что они унаследуют семантику заданного родительского класса.

Список наиболее значимых с точки зрения построения онтологий отношений представлен в приложении 3 к книге [11].

На данный момент среди базовых выделены следующие отношения:

- класс – подкласс;
- подкласс – класс;
- часть – целое;
- целое – часть;
- причина – следствие;
- следствие – причина.

Данный перечень является расширяемым в зависимости от предметной области, возможно введение новых классов отношений. Для этого необходимо задать класс и описать правила обработки его элементов.

#### 4. СТРУКТУРЫ И СЦЕНАРИИ

Согласно [12] люди используют в своем мышлении сценарии – структуры с заданными параметрами объектов и последовательностью действий.

Так, например, ситуация обучения в общем случае будет включать два объекта: учителя и ученика, процесс передачи знаний, а также среду, в которой это происходит, и инструменты.

Структура типа «конструкция» будет иметь множество вложенных объектов, связанных между собой отношением «часть–целое».

В рамках реализации языка представляется возможным выделить счетное количество таких шаблонов, благодаря которым пользователь сможет в полной мере описать рассматриваемую предметную область.

Данные шаблоны служат еще одним уровнем стандартизации знаний, регламентирующим не только форму представления, но и его порядок.

#### 5. СЛУЖЕБНЫЕ КОНСТРУКЦИИ ЯЗЫКА

Для реализации поддержки дескриптивной логики необходимо ввести в язык служебные конструкции, влияющие на семантику понятий. Данные конструкции должны быть строго формализованы в языке и учтены в алгоритмах трансляции.

Данные конструкции можно разделить на две группы. Первая группа отвечает за реализацию дескриптивной логики (OWL-DL и OWL-EL). Конструкции дескриптивной логики приведены в табл. 1.

Таблица 1

Table 1

Название	Синтаксис	Реализация в языке
Понятие	A	Запись, стоящая на первом месте или третьем месте в триплете
Всеобщее, универсальное понятие	T	<b>Что-либо (кто-либо)</b> , или синонимы. Массив является расширяемым
Пустое понятие	$\perp$	<b>Ничто (никто)</b> , или синонимы. Массив является расширяемым
Пересечение	$C \sqcap D$	<b>и</b> , оператор ставится между понятиями на одном уровне табуляции с субъектами
Объединение	$C \sqcup D$	<b>или</b> , оператор ставится между понятиями на одном уровне табуляции с субъектами
Симметричная разность	$C \Delta D$	<b>либо</b> , оператор ставится между понятиями на одном уровне табуляции с субъектами
Отрицание	$\neg C$	<b>не</b> , оператор ставится перед триплетом на строку выше, после него табуляция триплета смещается на один отступ
Ограничение на значение	$\forall R.C$	Принимается по умолчанию для классов
Квантификатор существования	$\exists R.C$	Принимается по умолчанию для индивидов

Вторая группа отвечает за реализацию безопасных правил (Safe rules) [13]. Конструкции правил приведены в табл. 2.

Таблица 2

Table 2

Название	Реализация в языке
Неявное задание классов	<p>&lt;Понятие&gt;&lt;Служебная конструкция&gt;&lt;Условие&gt;&lt;Следствие&gt;</p> <p>Пример: Человек     <b>который</b>         имеет:             сына;             дочь         является:             родителем</p>
Правила (могут использовать несколько переменных при записи выражений)	<p><b>Если</b> &lt;Условие&gt;, <b>то</b> &lt;Следствие выполнения&gt;</p> <p>Пример: <b>Если</b>     температура воздуха         Имеет значение, C°=         20, <b>то</b>     Ира         должна надеть:         футболку</p>

## 6. СИНТАКСИС И ГРАММАТИКА ЯЗЫКА

В рамках разработки синтаксиса онтологического языка основными задачами является идентификация объектов, отношений и субъектов, а также четкое структурное разграничение записей.

Определенную сложность представляют объекты, имена которых являются составными, поскольку в этом случае мы не можем использовать пробел в качестве разделителя между понятиями.

В продукте Fluent Editor 2015, упомянутом выше, задача решена путем введения дополнительного разделителя «-» (тире), которое служит для разделения слов в рамках одного понятия, а пробелы используются для разделения понятий между собой.

Пример записи триплета с составными понятиями в Fluent Editor 2015:

*Company-I has-common-nameequal-to 'Tommy's-Pizza-and-Subs'.*

В разрабатываемом КЕЯ также введены дополнительные разделители, принцип их организации взят из языков программирования. Для разделения

понятий используется табуляция, каждое следующее понятие начинается с новой строки и имеет больший отступ от начала строки чем следующее, за исключением перечислений субъектов.

Пример записи триплета с составными понятиями в разрабатываемом языке:

*Company 1*  
*has common name=*  
*Tommy's Pizza and Subs.*

Порядок записи строго определен: <объект> |<отношение><тип> | <субъект>.

Для определения типа используются следующие символы:

« : » – для обозначения отношения между объектами (object property), в этом случае субъектом выступает понятие;

« = » – для задания значений свойствам объекта (data property), в этом случае субъектом выступают числовые или текстовые данные.

В представлении онтологии, которое используется для чтения, символы, идентифицирующие тип отношения, не отображаются, для того чтобы не затруднять чтение конструкциями, не типичными для естественного языка.

Каждый новый триплет записывается с новой строки, в конце его ставится точка. Так, каждый триплет является отдельным предложением.

Структура языка должна позволять не только описывать знания, но и строить запросы к ним на естественном языке [14].

Ниже представлены примеры использования данного языка.

Особенности разрабатываемого языка будут рассматриваться в сравнении с OWL, а также с англоязычным аналогом.

- **OWL** – Web Ontology Language;
- **FE (controlled English)** – Fluent Editor 2015;
- **Onto (controlled Russian)** – Onto.plus.

В табл. 3–14 представлено сравнение аналогичных представлений на трех языках.

Таблица 3

Table 3

Классы		
OWL	FE	Onto
ИгрокХоккейнойКоманды	Игрок–хоккейной–команды	игрок хоккейной команды
Целостность составного объекта достигается за счет исключения пробелов Разделение слов происходит за счет заглавных букв	Целостность составного объекта достигается за счет замены пробелов на тире Тире служит разделителем слов	Целостность составного объекта достигается за счет принципа «одна строка – один объект» Слова разделяются пробелом



Таблица 4

Table 4

Экземпляр		
OWL	FE	Onto
<pre>&lt;ClassAssertion&gt; &lt;Class IRI=" ИгрокиХоккей- нойКоманды "/&gt; &lt;NamedIndividual IRI="Василий"/&gt; &lt;/ClassAssertion&gt;</pre>	Василий is a игрок– хоккейной–команды	<b>Василий</b> это: игрок хоккей- ной команды
Экземпляр задается явно	Экземпляр определяется по признаку заглавной буквы Классы прописываются нижним регистром, экземпляры начинаются с заглавной	Целостность составного объекта достигается за счет принципа «одна строка – один объект» Слова разделяются пробелом

Классы и экземпляры отличаются за счет регистров и спецсимволов.

Таблица 5

Table 5

Свойства		
OWL	FE	Onto
<pre>&lt;DataPropertyAssertion&gt; &lt;DataProperty IRI="hasAge" /&gt; &lt;NamedIndividual IRI="Tom" /&gt; &lt;Literal&gt;23&lt;/Literal&gt; &lt;/DataPropertyAssertion&gt;</pre>	Tom has-age equal-to 23	<b>Василий</b> возраст= 23
Свойство задается явно	Свойство определяется явным заданием его значения, через оператор: equal-to	Свойство определяется явным заданием его значения через оператор «=»

Таблица 6

Table 6

Включения классов		
OWL	FE	Onto
<pre>&lt;SubClassOf&gt;   &lt;Class IRI="Tree" /&gt;   &lt;Class IRI="Plant" /&gt; &lt;/SubClassOf&gt;</pre>	Every tree is a plant	деревья это: растения
<b>Отношение включения задается явно</b>	Отношение иерархии характеризуется обобщением в начале предложения	Отношение иерархии. Обобщение принимается по умолчанию
<pre>&lt;SubClassOf&gt;   &lt;Class IRI="Branch" /&gt;   &lt;ObjectSomeValuesFrom&gt;     &lt;ObjectProperty IRI="isPartOf"/&gt;     &lt;Class IRI="Tree" /&gt;   &lt;/ObjectSomeValuesFrom&gt; &lt;/SubClassOf&gt;</pre>	Every branch is-part-of a tree	ветки являются частью: деревьев
Экзистенциальное ограничение роли. <b>Отношение агрегации задается явно, между классом, заданным явно и классом, заданным неявно (части деревьев)</b>	Отношение часть–целое характеризуется обобщением в начале предложения	Отношение часть–целое. Обобщение принимается по умолчанию
<pre>&lt;SubClassOf&gt;   &lt;Class IRI="Lion" /&gt;   &lt;ObjectAllValuesFrom&gt;     &lt;ObjectProperty IRI="eats"/&gt;     &lt;Class IRI="Herbivore" /&gt;   &lt;/ObjectAllValuesFrom&gt; &lt;/SubClassOf&gt;</pre>	Every lion eats nothing-but herbivore	львы едят: только растения
Универсальное ограничение роли. <b>Отношение включения задается явно</b> <b>Между львами и всеми, кто ест только травоядных</b>	Отношение агрегации характеризуется обобщением в начале предложения. Родительский класс формируется по значению признака «едят». Для этого вводится ограничительная конструкция	Отношение агрегации. За счет вводной конструкции только накладывается ограничение за значение отношения «едят»

Оба типа ограничения (экзистенциальные и универсальные) дополняют друг друга. Отличаются они введением ключевого слова «только».

Окончание табл. 6

End of Tab. 6

Включения классов		
OWL	FE	Onto
<pre> &lt;SubClassOf&gt;   &lt;Class IRI="Giraffe" /&gt;   &lt;ObjectAllValuesFrom&gt;     &lt;ObjectProperty IRI="eats"/&gt;     &lt;ObjectUnionOf&gt;       &lt;Class IRI="Leaf" /&gt;       &lt;Class IRI="Twig" /&gt;     &lt;/ObjectUnionOf&gt;   &lt;/ObjectAllValuesFrom&gt; &lt;/SubClassOf&gt; </pre>	Every giraffe eats nothing-but thing that is a leaf and-or is a twig	<p>Жирафы</p> <p>едят только:</p> <p>листки</p> <p>или</p> <p>едят только:</p> <p>ветки.</p> <p><i>Короткая запись:</i></p> <p>жирафы</p> <p>едят только:</p> <p>листки</p> <p>или</p> <p>ветки</p>
Отношение включения задается явно. Пример, в котором область определения значений отношения задана объединением и пересечением двух классов	Для неявного задания используется универсальный концепт с ограничениями	Ограничения задаются явно. Для удобства сформулирована короткая форма записи без дублирования отношений
<pre> &lt;SubClassOf&gt;   &lt;Class IRI="TastyPlant" /&gt;   &lt;ObjectIntersectionOf&gt;     &lt;ObjectSomeValuesFrom&gt;       &lt;ObjectInverseOf&gt;         &lt;ObjectProperty IRI="eats"/&gt;       &lt;/ObjectInverseOf&gt;     &lt;Class IRI="Carnivore" /&gt;   &lt;/ObjectSomeValuesFrom&gt;   &lt;ObjectSomeValuesFrom&gt;     &lt;ObjectInverseOf&gt;       &lt;ObjectProperty IRI="eats"/&gt;     &lt;/ObjectInverseOf&gt;     &lt;Class IRI="Herbivore" /&gt;   &lt;/ObjectSomeValuesFrom&gt; &lt;/ObjectIntersectionOf&gt; &lt;/SubClassOf&gt; </pre>	Every tasty-plant is eaten by a carnivore and is eaten by a herbivore	<p>Вкусные растения</p> <p>Служат пищей:</p> <p>травоядным</p> <p>и</p> <p>Служат пищей:</p> <p>хищникам</p>
Пример, в котором область определения задана пересечением. <b>Пересекаются классы хищников и травоядных</b>	Отношение включения характеризуется обобщением в начале предложения	Пересечение идет на уровне отношений, поскольку они могут отличаться, и объединение на уровне субъектов не работает
<pre> &lt;SubClassOf&gt;   &lt;Class IRI="PalmTree" /&gt;   &lt;ObjectSomeValuesFrom&gt;     &lt;ObjectProperty IRI="hasPart"/&gt;     &lt;ObjectComplementOf&gt;       &lt;Class IRI="Branch" /&gt;     &lt;/ObjectComplementOf&gt;   &lt;/ObjectSomeValuesFrom&gt; &lt;/SubClassOf&gt; </pre>	Every palm-tree has-part something that is not a branch	<p>пальмы</p> <p>не состоят из:</p> <p>веток</p>
Пример, в котором область определения задана дополнением (исключением из множества)	Отношение включения характеризуется обобщением в начале предложения	За счет вводной конструкции «не». Происходит исключение из множества

Таблица 7

Table 7

Эквивалентность классов		
OWL	FE	Onto
<code>&lt;EquivalentClasses&gt;</code> <code>&lt;Class IRI="Boy" /&gt;</code> <code>&lt;Class IRI="YoungMaleMan" /&gt;</code> <code>&lt;/EquivalentClasses&gt;</code>	Something is a boy if-and-only-if-it is a young-male-man	мальчик эквивалентен: юноша
Отношение эквивалентности задается явно	Отношение определяется через универсальный концепт, который может относиться к одному классу, только в случае если относится к другому. Это условие характеризуется вводной конструкцией	Явное задание отношения между классами
<code>&lt;EquivalentClasses&gt;</code> <code>&lt;Class IRI="Man" /&gt;</code> <code>&lt;ObjectIntersectionOf&gt;</code> <code>&lt;Class IRI="Adult" /&gt;</code> <code>&lt;Class IRI="Male" /&gt;</code> <code>&lt;Class IRI="Person" /&gt;</code> <code>&lt;/ObjectIntersectionOf&gt;</code> <code>&lt;/EquivalentClasses&gt;</code>	Something is a man if-and-only-if-it is an adult that is a male and is a person	Кто-то эквивалентен: мужчине только если является: взрослым который является: человек мужского пола...
Пример, в котором класс эквивалентен пересечению трех других классов	Класс задан не явно через пересечение	Класс задан не явно через пересечение

Эквивалентность – частный случай включения, при котором каждый из классов целиком включает друг друга.

Таблица 8

Table 8

Непересекающиеся классы.		
OWL	FE	Onto
<code>&lt;SubClassOf&gt;</code> <code>&lt;Class IRI="Herbivore" /&gt;</code> <code>&lt;ObjectComplementOf&gt;</code> <code>&lt;Class IRI="Carnivore" /&gt;</code> <code>&lt;/ObjectComplementOf&gt;</code> <code>&lt;/SubClassOf&gt;</code>	Every herbivore is not a carnivore	травоядные не являются: хищниками
Запись определяет непересекающийся класс через эквивалент его дополнения	Все элементы одного класса не являются элементами другого	Все элементы одного класса не являются элементами другого

Дополнительный атрибут отношения, который накладывает ограничение на количество связей между классами.

Таблица 9

Table 9

Ограничение множеств		
OWL	FE	Onto
<pre>&lt;SubClassOf&gt;   &lt;Class IRI="Person" /&gt;   &lt;ObjectMaxCardinality cardinality="2" &gt;     &lt;ObjectProperty IRI="isChildOf"/&gt;   &lt;Class IRI="Parent"&gt;   &lt;/ObjectMaxCardinality&gt; &lt;/SubClassOf&gt;</pre>	Every person is-a-child-of at-most two parents	человек является ребенком: максимум 2 родителей
Атрибут ограничения сверху. Задается максимальное количество связей между экземплярами классов	Атрибут задается вводной конструкцией, а также числом либо словом, обозначающим число (числа до десяти имеют в языке словесные эквиваленты)	Атрибут задается вводной конструкцией «максимум» и числом
<pre>&lt;SubClassOf&gt;   &lt;Class IRI="Person" /&gt;   &lt;ObjectMinCardinality cardinality="2" &gt;     &lt;ObjectProperty IRI="isChildOf"/&gt;   &lt;Class IRI="Parent"&gt;   &lt;/ObjectMinCardinality&gt; &lt;/SubClassOf&gt;</pre>	Every person is-a-child-of at-least two parents	человек является ребенком: минимум 2 родителей
Атрибут ограничения снизу. Задается минимальное количество связей между экземплярами классов	Атрибут задается вводной конструкцией, а также числом либо словом, обозначающим число	Атрибут задается вводной конструкцией «минимум» и числом
<pre>&lt;SubClassOf&gt;   &lt;Class IRI="Person" /&gt;   &lt;ObjectExactCardinality cardinality="2" &gt;     &lt;ObjectProperty IRI="isChildOf"/&gt;   &lt;Class IRI="Parent"&gt;   &lt;/ObjectExactCardinality&gt; &lt;/SubClassOf&gt;</pre>	Every person is-a-child-of two parents	человек является ребенком: ровно 2 родителей

Таблица 10

Table 10

Значение	FE	Onto	OWL
<2	less-than two	меньше 2	<ObjectMaxCardinality cardinality="1">
>2	more-than two	больше 2	<ObjectMinCardinality cardinality="3">
≤2	at-most two	максимум 2	<ObjectMaxCardinality cardinality="2">
≥2	at-least two	минимум 2	<ObjectMinCardinality cardinality="2">
≠2	different-than two	не ровно 2	<ObjectMinCardinality cardinality="3"><ObjectMaxCardinality cardinality="1">
=2	two	ровно 2	<ObjectExactCardinality cardinality="2" >

В контролируемом языке есть несколько вводных конструкций, но в OWL они транслируются таким образом, что выражаются только через максимум, минимум и «равно».

Для классов с несколькими экземплярами иногда приходится выделять ключевое свойство, которое является параметром идентификации уникальной записи.

Таблица 11

Table 11

Ключевые свойства		
OWL	FE	Onto
<pre>&lt;HasKey&gt;   &lt;Class IRI="Man" /&gt;   &lt;ObjectProperty IRI="hasId"/&gt; &lt;/HasKey&gt;</pre>	Every X that is a man is-unique- if X has-id something	Все X которые являются: людьми уникальны если X имеет id: что-либо
Явное указание ключевого параметра	Задание параметра через вводную конструкцию	Задание параметра через вводную конструкцию

Таблица 12

Table 12

Задание сложных понятий		
OWL	FE	Onto
<pre>&lt;DLSafeRule&gt; &lt;Body&gt;   &lt;ClassAtom&gt;   &lt;Class IRI="Cat" /&gt;   &lt;Variable IRI="CatX" /&gt;   &lt;/ClassAtom&gt;   &lt;ClassAtom&gt;   &lt;ObjectIntersectionOf&gt;   &lt;Class IRI="Animal" /&gt;   &lt;ObjectMinCardinality cardinality="5"&gt;   &lt;ObjectProperty IRI="has" /&gt;   &lt;Class IRI="Leg" /&gt;   &lt;/ObjectMinCardinality&gt;   &lt;/ObjectIntersectionOf&gt;   &lt;Variable IRI="CatX" /&gt;   &lt;/ClassAtom&gt; &lt;/Body&gt; &lt;Head&gt;   &lt;ClassAtom&gt;   &lt;ObjectComplementOf&gt;   &lt;Class IRI="Bird" /&gt;   &lt;/ObjectComplementOf&gt;   &lt;Variable IRI="CatX" /&gt;   &lt;/ClassAtom&gt; &lt;/Head&gt; &lt;/DLSafeRule&gt;</pre>	If a cat is an animal that has at- least five legs then the cat is something that is not a bird	Если кошка является: животным которое имеет: минимум 5 ног то эта кошка является: чем-то, что не является: птицей

Окончание табл. 12

End of Tab. 12

Задание сложных понятий		
OWL	FE	Onto
<b>Задание класса через набор условий.</b> <b>"Cat" – исходный класс.</b> <b>"CatX" – определяемый класс</b>	Через условную конструкцию описывается правило, по которому неявно заданный подкласс является не пересекаемым с другим классом	Через условную конструкцию описывается правило, по которому не явно заданный подкласс является не пересекаемым с другим классом

Таблица 13

Table 13

Задание экземпляров		
OWL	FE	Onto
<pre>&lt;ClassAssertion&gt;   &lt;Class IRI="Giraffe" /&gt;   &lt;NamedIndividual IRI="Sophie" /&gt; &lt;/ClassAssertion&gt;</pre>	Sophie is a giraffe	София это: жираф
<b>Экземпляр задается явно</b>	Экземпляр распознается за счет написания его с заглавной буквы	Экземпляр распознается за счет написания его с заглавной буквы

Таблица 14

Table 14

Подклассы отношений		
OWL	FE	Onto
<pre>&lt;SubObjectPropertyOf&gt;   &lt;ObjectProperty IRI="isProperPartOf" /&gt;   &lt;ObjectProperty IRI="isPartOf" /&gt; &lt;/SubObjectPropertyOf&gt;</pre>	If X is-proper-part-of Y then X is-part-of Y	Отношение собственная часть это: быть частью
<b>Явное задание подмножества</b>	Задание через условия и переменные	Явное задание через вводную конструкцию «Отношение»

Отдельно стоит рассмотреть задачу задания аксиом, потому как в OWL часть из них является стандартной характеристикой отношений. Люди же, как правило, не оперируют в данных категориях. Для человека привычнее рассматривать то или иное утверждение на примере объектов. Сравнение задания аксиом рассмотрено в табл. 15.

Таблица 15

Table 15

Транзитивность		
OWL	FE	Onto
<pre>&lt;TransitiveObjectProperty&gt;   &lt;ObjectProperty IRI="hasPart" /&gt; &lt;/TransitiveObjectProperty&gt;</pre>	<p>If X has-part something that has-part Y then X has-part Y</p>	<p>Если X состоит из: чего-либо, что состоит из: Y, то X состоит из: Y</p>
<b>Явное задание транзитивности отношения</b>	Задание через условия и переменные	Задание через условия и переменные
Рефлексивность		
OWL	FE	Onto
<pre>&lt;ReflexiveObjectProperty&gt;   &lt;ObjectProperty IRI="isPartOf" /&gt; &lt;/ReflexiveObjectProperty&gt;</pre>	<p>Every-single-thing is- part-of itself</p>	<p>Все X состоит из: X</p>
<b>Явное задание рефлексивности отношения</b>	Задание через условия и вводные конструкции	Задание через условия и переменные
Иррефлексивность		
OWL	FE	Onto
<pre>&lt;SubClassOf&gt;   &lt;Class abbreviatedIRI="owl:Thing" /&gt; &gt;   &lt;ObjectComplementOf&gt;     &lt;ObjectHasSelf&gt;       &lt;ObjectProperty IRI="isProperPartOf" /&gt;     &lt;/ObjectHasSelf&gt;   &lt;/ObjectComplementOf&gt; &lt;/SubClassOf&gt;</pre>	<p>Every-single-thing is not a thing that is-proper- part-of itself</p>	<p>Все X не состоит из: чего-либо, что состоит из: X</p>
<b>Явное задание рефлексивности отношения</b>	Задание через условия и вводные конструкции	Задание через условия и переменные
Симметричность		
OWL	FE	Onto
<pre>&lt;EquivalentObjectProperties&gt;   &lt;ObjectProperty IRI="isFriendOf" /&gt;   &lt;ObjectInverseOf&gt;     &lt;ObjectProperty IRI="isFriendOf" /&gt;   &lt;/ObjectInverseOf&gt; &lt;/EquivalentObjectProperties&gt;</pre>	<p>X is-friend-of Y if-and- only-if Y is-friend-of X</p>	<p>X друг: Y, только если Y друг: X</p>



Окончание табл. 15

End of Tab. 15

Симметричность		
OWL	OWL	OWL
Задание через эквивалентность и инверсию отношений.	Задание через условия, вводные конструкции и переменные	Задание через условия и переменные
Эквивалентность		
OWL	FE	Onto
<code>&lt;EquivalentObjectProperties&gt;</code> <code>&lt;ObjectProperty IRI="isCloseTo" /&gt;</code> <code>&lt;ObjectProperty IRI="isNearTo" /&gt;</code> <code>&lt;/EquivalentObjectProperties&gt;</code>	X is-close-to Y if-and-only-if X is-near-to Y	X рядом с: Y, только если X не далеко: Y
Явное задание эквивалентности отношений	Задание через условия и переменные	Задание через условия и переменные
Непересекаемость		
OWL	FE	Onto
<code>&lt;DisjointObjectProperties&gt;</code> <code>&lt;ObjectProperty IRI="hates" /&gt;</code> <code>&lt;ObjectProperty IRI="loves" /&gt;</code> <code>&lt;/DisjointObjectProperties&gt;</code>	If X loves Y then X does-not hate Y	X ненавидит: Y, только если X не любит: Y
Явное задание непересекаемости отношений	Задание через условия и переменные	Задание через условия и переменные

Обработка отрицаний отличается от английского языка. Поскольку в русском языке нет определенных правил, согласно которым частица «не» используется в определенном месте предложения, в КЕЯ она учитывается в составе отношения на любой позиции.

## 7. ПОВЕСТВОВАНИЕ

Задание онтологии на естественном языке позволяет сохранять порядок повествования и тем самым вводить в онтологии дополнительную семантику. Такой подход позволяет строить описание предметной области в порядке, предусмотренном автором онтологии (например, хронологическом или иерархическом).

В случае изложения материалов предметной области в виде повествования важную роль играет беспрепятственное восприятие текста информации –

гладкочтение. Данный термин включает в себя отсутствие символов, слов и конструкций, не предназначенных для естественного языка.

Разрабатываемый КЕЯ соответствует данным требованиям, а значит, будет понятен человеку без каких-либо специализированных навыков.

## ЗАКЛЮЧЕНИЕ

В рамках разработки КЕЯ получилось ограничить естественный язык, при этом сохранив его свойства гладкочтения, для обычного пользователя. Таким образом был сформирован язык, который, с одной стороны, беспрепятственно воспринимается человеком, а с другой – может быть интерпретирован как онтология машиной логического вывода. Данный язык поддерживает все необходимые стандарты для трансляции его в OWL и дальнейшей работы в других программных средах. КЕЯ базируется на дескриптивной логике и поддерживает все ее базовые конструкции, а также отвечает требованиям, предъявляемым к онтологическим языкам.

В рамках разработки языка некоторые конструкции были упрощены относительно англоязычного аналога, а также адаптированы к специфике русского языка.

Эффективность процесса управления знаниями напрямую зависит от формы их представления. Широкое распространение онтологий в целях организации контента открывает новые возможности для работы с семантикой, заключенной в текстах. Данная работа позволяет снизить порог входа для использования онтологий людьми без специальных навыков.

## СПИСОК ЛИТЕРАТУРЫ

1. *Safwat H., Davis B., Zarrouk M.* Engineering an aligned gold-standard corpus of human to machine oriented Controlled Natural Language // 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI). – Santiago, Chile, 2018. – P. 421–427. – DOI: 10.1109/WI.2018.00-58.
2. *Kuhn T.* Controlled English for knowledge representation: Doctoral thesis / Faculty of Economics, Business Administration and Information Technology of the University of Zurich. – Zurich, 2010. – 242 p.
3. *Weichbroth P.* Fluent editor and Controlled Natural Language in ontology development // International Journal on Artificial Intelligence Tools. – 2019. – Vol. 28, N 4. – P. 243.
4. *Милов Ю.П.* О стандарте сетевого русского языка, или Да здравствует КРЯ! // Информационные ресурсы России. – 2016. – № 3. – С. 40–41.
5. *Toti D., Longhi A.* SEMANTO: a graphical ontology management system for knowledge discovery // Journal of Ambient Intelligence and Humanized Computing. – 2018. – Vol. 9, N 4. – P. 1075–1084.
6. *Хабаров В.И., Волегжанина И.С.* Цифровые трансформации в профессиональном образовании (на примере подготовки кадров транспорта): монография. – М.: Русайнс, 2018. – 210 с.
7. *Тузовский А.Ф., Чуриков С.В., Ямпольский В.З.* Системы управления знаниями (методы и технологии) / под общ. ред. В.З. Ямпольского. – Томск: Изд-во НТЛ, 2005. – 258 с.
8. *W3C OWL Working Group.* Web Ontology Language [Electronic resource]. – URL: <https://www.w3.org/TR/owl2-overview/> (accessed: 13.12.2019).
9. *Гаврилова Т.А., Хорошевский В.Ф.* Базы знаний интеллектуальных систем: учебное пособие. – СПб.: Питер, 2000. – 382 с.

10. Минский М. Фреймы для представления знаний. – М.: Мир, 1979. – 151 с.
11. Методы представления знаний: методические указания / сост. И.Л. Коробова. – Тамбов: Изд-во Тамбов. гос. технол. ун-та, 2003. – 24 с.
12. Люгер Дж.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. – 4-е изд. – М.: Вильямс, 2005. – 283 с.
13. Hitzler P., Parsia B. Ontologies and rules // Handbook on ontologies / S. Staab, R. Studer (eds.). – Berlin: Springer, 2009. – P. 111–132.
14. Hu X., Dang D. Natural language aggregate query over RDF data / X. Hu, D. Dang, Y. Yao, L. Ye // Information Sciences. – 2018. – Vol. 454–455. – P. 363–381.

*Хабаров Валерий Иванович*, доктор технических наук, профессор теоретической и прикладной информатики факультета прикладной математики и информатики Новосибирского государственного технического университета. Основные направления научных исследований: искусственный интеллект, планирование эксперимента, статистические методы анализа данных, математическое моделирование транспортных потоков. Имеет более 150 публикаций. E-mail: khabarov51@mail.ru

*Степанов Илья Сергеевич*, аспирант кафедры информационных технологий на транспорте Сибирского государственного университета путей сообщения. Основное направление научных исследований – искусственный интеллект и интеллектуальные системы, основанные на знаниях. Имеет 6 публикаций. E-mail: step\_il@mail.ru

*Серенко Александр Андреевич*, младший научный сотрудник Института вычислительной математики и математической геофизики Сибирского отделения РАН. Основное направление научных исследований – искусственный интеллект и интеллектуальные системы, основанные на знаниях. Имеет 8 публикации. E-mail: fafnur@yandex.ru

*Khabarov Valery Ivanovich*, D.Sc. (Eng.), professor at the department of theoretical and applied informatics at the faculty of applied mathematics, Novosibirsk State Technical University. His main research fields include artificial intelligence, experiment design, statistical methods of data analysis and mathematical modeling of traffic flows. He is the author of more than 150 publications. E-mail: khabarov51@mail.ru

*Stepanov Ilia Sergeevich*, a postgraduate student at the department of information technologies, State Siberian Transport University. His research interests are focused on artificial intelligence and intelligent knowledge-based systems. He has published 6 research papers. E-mail: step\_il@mail.ru

*Serenko Alexander Andreevich*, a junior research fellow at the Institute of Computational Mathematics and Mathematical Geophysics, SB RAS. His main field of research covers artificial intelligence and intelligent knowledge-based systems. He is the author of 8 publications. E-mail: fafnur@yandex.ru

DOI: 10.17212/1814-1196-2019-4-99-120

***Controlled natural language for ontology editing\****V.I. KHABAROV<sup>1,a</sup>, I.S. STEPANOV<sup>2,b</sup>, A.A. SERENKO<sup>3,c</sup><sup>1</sup>Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation<sup>2</sup>Siberian State Transport University, 191 Dusi Kovalchuk Street, Novosibirsk, 630049, Russian Federation<sup>3</sup>Institute of Computational Mathematics and Mathematical Geophysics, Siberian Branch of the Russian Academy of Sciences, 6 Akademika Lavrenteva Prospekt, Novosibirsk, 630090, Russian Federation<sup>a</sup> khabarov51@mail.ru   <sup>b</sup> step\_il@mail.ru   <sup>c</sup> fafnur@yandex.ru**Abstract**

The work contains a description of a controlled natural language intended for working with ontologies. Thanks to a formalized definition of the conceptual structure of the subject area in ontology, it is possible to automatically process ontological information which is used in new search engines, information integration systems, and expert systems.

A brief analysis of analogues is given, as well as the rationale for the need to develop a specialized language. The requirements for the language are systematized taking into account the scope. The positioning of the developed language in relation to other representatives of the group of ontological languages is described. In this paper, we propose one of the possible versions of a controlled language, initially oriented to a Russian-speaking user. However, as practice has shown, a multilingual version of the presentation turned out to be possible and very useful for vocational training purposes due to the fact that the structure of many languages can be represented quite uniformly. Since a controlled natural language is a subset of the natural language, it does not contradict the syntactic rules of the Russian language, and its lexemes are semantically consistent. In turn, a natural language is more expressive and complex. At the same time, a controlled natural language must be unambiguously mapped into the standard OWL ontology presentation language. OWL is based on a descriptive logic (DL) and because of this, trilateral language matching is achieved.

The description of the structure, syntax and grammar of the language is also given. Representation of a descriptive logic and rules in language constructs are described. Examples of knowledge representation in the described format in comparison with analogues are given. The use of a controlled natural language for compiling a narrative with support of the ontological format is described. As part of the development of the language, a number of designs were simplified with respect to the English-language analogue, and also adapted to the specifics of the Russian language.

The effectiveness of the knowledge management process directly depends on the form of their presentation. The widespread use of ontologies in order to organize the content opens up new possibilities for working with semantics contained in texts. A controlled natural language is bi-directional, perceived by both man and an inference machine. Its use will make it possible to find a wider application of ontologies, as well as to lower an input threshold to work with them.

**Keywords:** controlled natural language, ontological language, ontologies, knowledge engineering, knowledge base, knowledge representation, inference mechanism, ontology editor, multilingualism, semantics, translators

---

\* Received 03 September 2019.

## REFERENCES

1. Safwat H., Davis B., Zarrouk M. Engineering an aligned gold-standard corpus of human to machine oriented Controlled Natural Language. *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Santiago, Chile, 2018, pp. 421–427. DOI: 10.1109/WI.2018.00-58.
2. Kuhn T. *Controlled English for knowledge representation*. Doctoral thesis. Zurich, 2010. 242 p.
3. Weichbroth P. Fluent editor and Controlled Natural Language in ontology development. *International Journal on Artificial Intelligence Tools*, 2019, vol. 28, no. 4, p. 243.
4. Milov Yu.P. O standarte setevogo russkogo yazyka, ili Da zdravstvuet KRYa! [On the standards of the internet's Russian language, or Long Live Controlled Russian Language!]. *Informatsionnye resursy Rossii – Information Resources of Russia*, 2016, no. 3, pp. 40–41. (In Russian).
5. Toti D., Longhi A. SEMANTO: a graphical ontology management system for knowledge discovery. *Journal of Ambient Intelligence and Humanized Computing*, 2018, vol. 9, no. 4, pp. 1075–1084.
6. Khabarov V.I., Volegzhnina I.S. *Tsifrovye transformatsii v professional'nom obrazovanii (na primere podgotovki kadrov transporta)* [Digital transformations in vocational education (on the example of training transport personnel)]. Moscow, Rusains Publ., 2018. 210 p.
7. Tuzovskii A.F., Chirikov S.V., Yampol'skii V.Z. *Sistemy upravleniya znaniyami (metody i tekhnologii)* [Knowledge management systems]. Tomsk, NTL Publ., 2005. 258 p.
8. W3C OWL Working Group. *Web Ontology Language*. Available at: <https://www.w3.org/TR/owl2-overview/> (accessed 13.12.2019).
9. Gavrilova T.A., Khoroshevskii V.F. *Bazy znaniy intellektual'nykh sistem* [Knowledge bases of intelligent systems]. St. Petersburg, Piter Publ., 2000. 382 p.
10. Minsky M. *A framework for representing knowledge*. Cambridge, 1974 (Russ. ed.: Minskii M. *Freimy dlya predstavleniya znaniy*. Moscow, Mir Publ., 1979. 151 p.).
11. Korobova I.L., comp. *Metody predstavleniya znaniy* [Methods of knowledge representation]. Tambov, Tambov State Technological University Publ., 2003. 24 p.
12. Luger G.F. *Artificial intelligence: structures and strategies for complex problem solving*. 4th ed. Harlow, England, New York, Pearson Education, 2002 (Russ. ed.: Lyuger Dzh.F. *Iskusstvennyi intellekt: strategii i metody resheniya slozhnykh problem*. 4th ed. Moscow, Williams Publ., 2005. 283 p.).
13. Hitzler P., Parsia B. Ontologies and rules. *Handbook on ontologies*. Ed. by S. Staab, R. Studer. Berlin, Springer, 2009, pp. 111–132.
14. Hu X., Dang D., Yao Y., Ye L. Natural language aggregate query over RDF data. *Information Sciences*, 2018, vol. 454–455, pp. 363–381.

Для цитирования:

Хабаров В.И., Степанов И.С., Серенко А.А. Контролируемый естественный язык для работы с онтологиями // Научный вестник НГТУ. – 2019. – № 4 (77). – С. 99–120. – DOI: 10.17212/1814-1196-2019-4-99-120.

For citation:

Khabarov V.I., Stepanov I.S., Serenko A.A. Kontroliruemiyi estestvennyi yazyk dlya ra-boty s ontologiyami [Controlled natural language for ontology editing]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2019, no. 4 (77), pp. 99–120. DOI: 10.17212/1814-1196-2019-4-99-120.