ИПМ им.М.В.Келдыша РАН

Абрау-2019 • Труды конференции

А.М. Гусенков, Н.Р. Бухараев,
Е.В. Биряльцев

Построение онтологии предметной области для корпоративного веб-приложения

Рекомендуемая форма библиографической ссылки

Гусенков А.М., Бухараев Н.Р., Биряльцев Е.В. Построение онтологии предметной области для корпоративного веб-приложения // Научный сервис в сети Интернет: труды XXI Всероссийской научной конференции (23-28 сентября 2019 г., г. Новороссийск). — М.: ИПМ им. М.В.Келдыша, 2019. — С. 302-317. — URL: http://keldysh.ru/abrau/2019/theses/58.pdf doi:10.20948/abrau-2019-58

Размещена также презентация к докладу

Построение онтологии предметной области для корпоративного веб-приложения

А.М. Гусенков[1], Н.Р. Бухараев[1], Е.В. Биряльцев[2]

[1] Казанский (Приволжский) федеральный университет, Казань, Россия
[2] ООО "Градиент Технолоджи", Казань, Россия

Аннотация. Представлена технология автоматизированного построения онтологии предметной области на основе информации, извлекаемой из комментариев реляционных баз данных ПАО «Татнефть». Технология основана на построении конвертора (компилятора), транслирующего логическую модель данных Epicentre Petrotechnical Open Software Corporation (POSC), представленную в виде ER-диаграмм и набора описаний на объектно-ориентированном языке EXPRESS, в язык описания онтологий OWL, рекомендованный консорциумом W3C. Описаны основные синтаксические и семантические аспекты преобразования.

Ключевые слова: онтология предметной области, реляционные базы данных, POSC, OWL

Building subject domain ontology for a corporate web application

A.M. Gusenkov[1], N.R. Bukharaev[1], E.V. Birialtsev[2]

1 Kazan Federal University, Kazan, Russia
2 Gradient Technology Ltd, Kazan, Russia

Abstract. The technology of automated construction of the subject domain ontology, based on information extracted from the comments of the TATNEFT oil company relational databases, is considered. The technology is based on building a converter (compiler) translating the logical data model of Epicenter Petrotechnical Open Software Corporation (POSC), presented in the form of ER diagrams and a set of the EXPRESS object-oriented language descriptions, into the OWL ontology description language, recommended by the W3C consortium. The basic syntactic and semantic aspects of the transformation are described.

Keywords: subject domain ontology, relational databases, POSC, OWL

1. Введение

Создание онтологии предметной области является, как правило, крайне трудоемким процессом, требующим участия высококвалифицированных специалистов, как в конкретной предметной области, так и в области компьютерной лингвистики. Одним из широко применяемых здесь методов является концептуализация понятийного аппарата [1]. При этом мы фактически строим объектную модель некой подобласти реального мира, определяя объекты, их атрибуты и взаимосвязи. Аналогичная техника выделения базовых сущностей и связей используется для построения логических моделей при проектировании реляционных баз данных [2]. Методологическая близость приемов, используемых при разработке онтологий и логических моделей баз данных, дает основание предположить возможность использования существующих логических моделей баз данных в качестве формализованного прототипа онтологии предметной области.

В статье, на примере разработки системы интеллектуального поиска для крупной нефтедобывающей компании, предложена методика автоматизированного построения онтологии предметной области на основе ее реляционной модели.

В качестве прототипа онтологии предметной области можно использовать логические модели, созданные в некоторых отраслях и имеющие статус отраслевого стандарта. Одной из них является модель данных Epicentre версии 3.0 нефтетехнической корпорации Petrotechnical Open Software Corporation (POSC [3]). Она представлена в виде ER-диаграмм [2], а также набора текстовых файлов на объектно-ориентированном языке EXPRESS (ISO 10303, part 11). Это представление ориентировано на генерацию структур баз данных по её логической модели, а также визуальное восприятие IT-специалистами.

Для описания онтологии был выбран язык OWL (Web Ontology Language) [4, 5], разработанный рабочей группой Semantic Web Activity и рекомендованный международным консорциумом W3C [6]. Реализация онтологии выполнена на диалекте языка OWL DL, соответствующем правилам дескриптивной логики, с перспективой дальнейшей разработки системы логического вывода на экземплярах понятий. В статье описана схема конвертации логической модели Epicentre в язык описания онтологий OWL, учитывающая, помимо общеотраслевых стандартов, специфику нефтегазодобывающей компании ПАО «Татнефть».

2. Модель Epicentre

В модели данных Epicentre определено более 1000 реально существующих технических и бизнес-объектов, связанных с разведкой и добычей нефти. В терминологии POSC-моделирования данных эти объекты названы сущностями (entities). В модели определены характеристики объектов, называемые атрибутами сущностей (attributes). Наиболее важными из них являются атрибуты, определяющие взаимосвязи между сущностями.

Один из важных архитектурных принципов Epicentre основан на различии между объектами, свойствами или характеристиками объектов (properties) и видами деятельности (activities). Это разделение поддерживается требованиями практики: возможностью свойств объекта иметь многократные версии или описания, а также тем, чтобы каждое свойство было однозначно связано со своим собственным определением (описанием) или историей обработки.

Каждая сущность, представленная в модели, определяется такими параметрами, как набор атрибутов, локальные правила, цепочки супертипов. Атрибут – это перечень объектов, описывающих данную сущность. В свою очередь атрибут имеет несколько параметров, таких, как имя атрибута, список опций (ключевой, обязательный, внешний и др.), и типы связей. С сущностями связаны также правила сущности – расширенные ограничения целостности данных, определяющие возможные значения атрибутов и связей.

В модели существует три типа справочных сущностей:

- POSC Fixed – имеет фиксированное количество экземпляров, определенных POSC;
- POSC Open – имеет фиксированные экземпляры, но возможно создание дополнительных экземпляров, не определенных POSC;
- Local – нет фиксированных определенных POSC-экземпляров, возможно определение собственных экземпляров.

Все справочные сущности имеют дополнительные характеристики, позволяющие задать источник и библиографию, связанную с источником информации, содержащейся в экземпляре. Для обозначения справочных сущностей и их типов в схеме Epicentre используется отличительная приставка Ref_ к имени сущности.

Объектно-ориентированная концепция наследования класса – важная часть архитектуры Epicentre. Так как модель данных достаточно велика, концепция наследования классов обеспечивает эффективный способ организации всех сущностей в логически связанную структуру.

Другая фундаментальная часть архитектуры Epicentre – это понимание того, что многие сущности характеризуются своим пространственным представлением. Каждый из различных геометрических объектов деятельности в разных разделах модели может быть связан с местоположением через отношения с одним или несколькими общими пространственными объектами.

Модель данных Epicentre задается на языке EXPRESS и использует базовые понятия этого языка:

Сущность (entity)
Супертип (supertype) и **подтип** (subtype)
Атрибут (attribute), явный (explicit) и инверсный (inverse)
Определенный тип данных (defined data type)
Простой тип (simple type)
Агрегатный тип (aggregate type)
Ограничение согласованности, правило "где" (where rule)

Ограничение уникальности (uniqueness rule)

Схема (schema)

Описанием сущности является определение понятия или класса модели Epicentre, на основе которого создаются экземпляры или объекты.

Сущность может быть подтипом сущности-супертипа и наследовать от нее атрибуты, правила и ограничения уникальности. Спецификация супертипа – это способ задания свойств типа, общих для всех подтипов.

Супертип может быть абстрактным, в этом случае присутствует конструкция ABSTRACT, что означает, что все экземпляры должны быть специализированы. Если сущность не является абстрактной, то экземпляры могут быть как специализированными, так и нет.

Атрибут (свойство) – это конкретная характеристика сущности. Он может быть явным, инверсным или выведенным. Атрибут имеет имя и тип представления.

Явный атрибут – такой атрибут, который не выведен ни из какого другого атрибута в модели.

Инверсный атрибут служит для выражения обратного направления отношения, возникающего при задании явного атрибута. Эта сущность переопределяет область определения атрибута таким образом, чтобы тип был определением данной сущности.

Определение схемы представляет собой контейнер, в котором содержатся определения всех сущностей, типов и ограничений, видимых в определенной схеме на языке EXPRESS (рис. 1).

```
ENTITY schema_definition;
        name : STRING;
        types : SET OF named_type;
        global_rules : SET OF global_rule;
        UNIQUE
        Url : name;
END_ENTITY;
```

Рис. 1. Определение схемы на языке EXPRESS

Определения атрибутов:

name: имя схемы;

types: набор сущностей и определенных типов, объявляемых в схеме;

global rules: набор глобальных ограничений, объявленных в схеме.

Формальные утверждения:

URl: имя схемы должно быть уникально.

Ограничение уникальности указывает комбинацию атрибутов, значения которых в совокупности однозначно идентифицируют конкретный экземпляр определяемой сущности.

Ограничение согласованности определяет ограничение, накладываемое на все экземпляры данной сущности.

3. Структура OWL

Язык веб-онтологий OWL (Web Ontology Language) предоставляет возможности:

- формального определения классов и свойств этих классов;
- определения индивидов (объектов-экземпляров, представителей классов) и их свойств;
- уточнения определений классов и индивидов до степени, задаваемой формальным синтаксисом OWL.

OWL максимально совместим с языками RDF [7] и RDF Schema [8]. Форматы XML [9] и RDF – часть стандарта OWL.

Основными элементами онтологии OWL являются классы, свойства, представители классов (индивиды или экземпляры) и отношения между этими представителями.

Наиболее фундаментальные понятия в предметной области должны соответствовать классам, которые находятся в корне различных таксономических деревьев. Каждый индивид в OWL является членом класса owl:Thing. Таким образом, каждый класс, определенный пользователем, автоматически является подклассом owl:Thing. Корневые классы, специфичные для данной предметной области, определяются простым объявлением именованного класса. Фундаментальным таксономическим конструктором для классов является отношение “быть подклассом” rdfs:subClassOf, описывающее связь частного класса с более общим.

Свойства позволяют утверждать общие факты о членах классов и особые факты об индивидах. Свойство – это бинарное отношение. Различают два типа свойств:

- свойства-значения – отношения между представителями классов и RDF-литералами или типами данных, определяемых XML Schema;
- свойства-объекты – отношения между представителями двух классов.

При определении свойства существует множество способов ограничить это отношение. Можно определить домен и диапазон. Свойство может быть определено как специализация (подсвойство) существующего свойства. Возможны и более сложные ограничения. Свойства, как и классы, могут быть организованы в иерархию.

Существует возможность определить характеристики свойства, что обеспечивает мощный механизм для усовершенствованного рассуждения о свойстве. Приведем некоторые характеристики свойств, имеющие важное значение для описания методики конвертации логической модели Epicentre в язык описания онтологий OWL.

TransitiveProperty

Если свойство P определено как транзитивное, то для любых x, y и z: P(x,y) и P(y,z) предполагают P(x,z).

SymmetricProperty

Если свойство P помечено как симметрическое, то для любых x и y: P(x,y), если P(y,x).

FunctionalProperty

Если свойство P помечено как функциональное, то для любых x, y и z: P(x,y) и P(x,z) предполагают y=z.

inverseOf

Если свойство P1 помечено как owl:inverseOf P2, то для всех x и y: P1(x,y), если P2(y,x).

Заметим, что синтаксис для owl:inverseOf берет в качестве аргумента название свойства. A, если B означает (A предполагает B) и (B предполагает A).

InverseFunctionalProperty

Если свойство P помечено как обратно функциональное, то для всех x, y и z: P(y,x) и P(z,x) предполагает y=z.

В дополнение к обозначению характеристик свойств можно разными способами еще больше ограничить диапазон свойства в определенных контекстах. Это делается с помощью следующих ограничений свойств.

allValuesFrom

Данные ограничения являются локальными по отношению к содержащему их классу. Ограничение owl:allValuesFrom требует, чтобы для каждого представителя данного класса, который имеет данное свойство, все значения этого свойства являлись представителями класса, заданного в пункте owl:allValuesFrom.

someValuesFrom

Ограничение owl:someValuesFrom требует, чтобы для каждого представителя данного класса, который имеет данное свойство, хотя бы одно значение этого свойства являлось представителем класса, заданного в пункте owl:someValuesFrom.

Кардинальность

Параметр owl:cardinality позволяет указать количество элементов в связи. Например, свойство UNIQUE класса Name_entity определяет единственную связь (рис. 2).

В частности, в OWL DL owl:maxCardinality может использоваться для указания верхнего предела, а owl:minCardinality – для нижнего предела. В комбинации друг с другом они могут использоваться для ограничения кардинальности свойства в пределах числового интервала.

```
<owl:Class rdf:ID=" Name_entity">
        <rdfs:subClassOf>
                <owl:Restriction>
                        <owl:onProperty rdf:resource="#UNIQUE"/>
                        <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1
                        </owl:cardinality>
                </owl:Restriction>
        </rdfs:subClassOf>
</owl:Class>
```

Рис. 2. Пример определения кардинальности на языке OWL

4. Конвертация модели Epicentre в OWL DL

При построении OWL-онтологии были использованы следующие основные подходы:

- любой сущности Epicentre соответствует простой именованный класс OWL-онтологии с сохранением в именах классов приставок, позволяющих идентифицировать сущности-свойства и сущности-справочники; все эти классы располагаются в корне таксономического дерева онтологии;
- степени связности сущностей (один-к-одному, один-ко-многим, многие-ко-многим) в OWL соответствует определение простых свойств-атрибутов, если связанная сущность не является типом данных, и свойств-значений в противном случае; указание степени связи между классами реализовано с помощью понятия кардинальности в OWL.

В OWL отсутствуют структурные элементы, которые в полном объеме описывают определение уникальности объектов модели Epicentre. Поэтому в определение каждого класса на языке OWL добавлено новое предопределенное свойство, в котором перечислены все атрибуты, образующие уникальный ключ. Аналогичным же образом решена проблема сохранения условий ограничений. Для каждой категории данных Epicentre в OWL-онтологии построены отдельные классы, в свойствах которых использовались встроенные типы данных языка OWL. Построена формальная LR(1)-грамматика [10] модели Epicentre, на основе которой реализовано семантическое преобразование модели Epicentre версии 3.0, описанной на языке EXPRESS, в онтологию на языке OWL DL. Выполнена русификация описания сущностей и атрибутов модели Epicentre, а также соответствующих им классов и свойств на OWL. Построение онтологии реализовано на диалекте языка OWL DL, соответствующем правилам дескриптивной логики, что в дальнейшем позволит использовать системы логического вывода на экземплярах понятий.

Программная реализация конвертора модели Epicentre в язык OWL DL выполнена на языке Java с использованием генератора лексических анализаторов flex [11] и генератора синтаксических анализаторов CUP [12].

Для апробации интеграции баз данных на основе онтологий и проверки корректности построенных онтологий разработан программный интерфейс OakOwlProject 1.0, обеспечивающий навигацию и манипуляции со всей совокупностью построенных онтологий.

В качестве прототипа для реализации некоторых интерфейсных и функциональных возможностей среды OakOwlProject был выбран известный Java-проект с открытым исходным кодом Protégé [13], функционал которого был существенно расширен. Внешний вид интерфейса OakOwlProject в части его работы с онтологиями предметной области показан на рис. 3.

Epicentre на языке EXPRESS представляет собой последовательность описаний сущностей, каждая из которых имеет синтаксическую структуру, представленную на рис. 4.

В данной синтаксической конструкции используются следующие обозначения:

name_entity – имя сущности, может иметь приставку Pty_ или Ref_ **(**которые соответственно указывают на сущность–свойство или справочную сущность**);**

name_entity1, ... , name_entityN – список сущностей name_entity**;**

name_attribute – имя прямого атрибута, может иметь приставку Ref_;

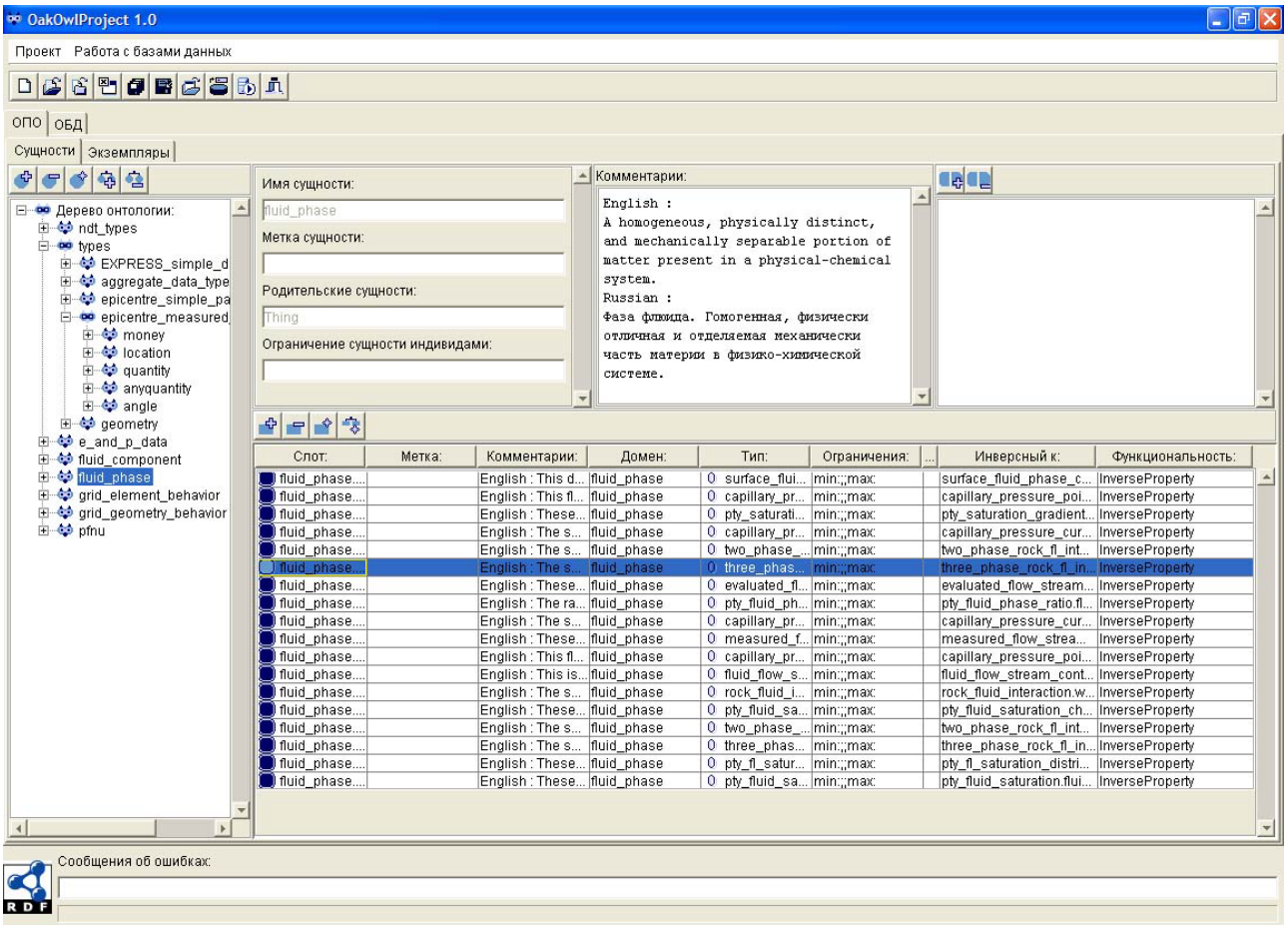

Рис. 3. Общий вид интерфейса пользователя системы OakOwlProject

```
ENTITY name_entity
        ABSTRACT SUPERTYPE OF (
                ONEOF(name_entity1, ..., name_entityN)
        )
        SUBTYPE OF (sub_name1, ..., sub_nameN);
        name_attribute: OPTIONAL SET[0 :?] OF type_name
        ……….
        INVERSE
                invers_attribute_name: SET[0 :?] OF invers_name_entity
                                                FOR entity_invers_type
        ……….
        UNIQUE
        si: name_attribute1, ..., name_attributeN
        WHERE условие
        ……..
END_ENTITY;
```

Рис. 4. Описаний сущностей на языке EXPRESS

sub_name – имя родительской сущности, может быть обычной сущностью (приставка отсутствует) или сущностью-свойством (имеется приставка Pty_);

sub_name1, ... , sub_nameN – список сущностей sub_name**;**

type_name – имя типа прямого атрибута, может быть справочной сущностью (Ref_), обычной сущностью (приставка отсутствует) или именованным типом данных (приставка Ndt_);

invers_attribute_name – имя инверсного атрибута, может быть обычной сущностью (приставка отсутствует) или сущностью-свойством (имеется приставка Pty_);

invers_name_entity – имя сущности, которая связана обратным отношением с заданной сущностью, может быть справочной сущностью (Ref_), обычной сущностью (приставка отсутствует) или сущностью-свойством (Pty_);

entity_invers_type – имя сущности, которая является типом прямого атрибута, соответствующего данному инверсному атрибуту, может быть обычной или справочной сущностью;

name_attribute1, ... , name_attributeN – список атрибутов name_attribute.

Смысл ключевых слов, набранных прописными буквами, будет описан ниже.

На рис. 5 представлен способ конвертации сущностей в классы OWL.

```
ENTITY name_entity          <owl:Class rdf:ID=" name_entity ">
...                   =>    ...
END_ENTITY;                 </owl:Class
```

Рис. 5. Конвертация сущностей в классы OWL

В Epicentre базовыми классами являются: E_AND_P_DATA, FLUID_COMPONENT, FLUID_PHASE, GRID_ELEMENT_BEHAVIOR, GRID_GEOMETRY_BEHAVIOR, PFNU. Автоматически данные классы будут являться подклассами класса owl:Thing и располагаться в корне таксономического дерева. В оболочке OakOwlProject_1.0, например, базовые классы находятся в корне иерархии, показанной на рис. 6.

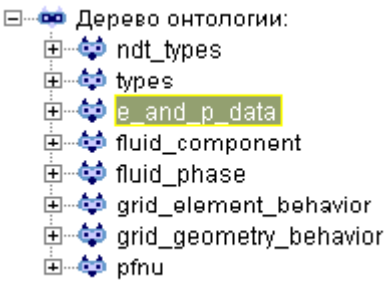

Рис. 6. Базовые классы модели

Конвертация отношений наследования показана на рис. 7. Здесь в конструкции SUBTYPE OF перечисляется список родительских сущностей для данной сущности. Данному отношению в OWL однозначно соответствует синтаксическая конструкция subclass.

```
SUBTYPE OF (
    sub_name1, ..., sub_nameN =>
);
```

```
<owl:Class rdf:ID="name_entity">
<rdfs:subClassOf rdf:resource="# sub_name1" />
        ...
        <rdfs:subClassOf rdf:resource="# sub_nameN" />
        ...
</owl:Class>
```

Рис. 7. Конвертация отношений наследования классов

После определения названия сущности и ее места в иерархии в модели Epicentre приводится список атрибутов, как прямых, так и инверсных. Определению прямого атрибута сущности соответствует синтаксическая конструкция на рис. 8.

```
name_attribute: OPTIONAL SET[0 :?] OF type_name
```

Рис. 8. Определение прямого атрибута сущности

Синтаксические конструкции OPTIONAL и SET[0:?] OF могут отсутствовать. Ключевое слово OPTIONAL определяет необязательность наличия значения атрибута, а конструкция SET[0:?] OF определяет степень связности (один-к-одному, один-ко-многим, многие-ко-многим), если атрибут является агрегатным. Такому блоку в OWL соответствует определение простых

свойств-атрибутов, если связанная сущность не является типом данных, и свойств-значений в противном случае. Таким образом, такой блок может быть представлен в OWL синтаксической конструкцией, показанной на рис. 9.

```
<owl:ObjectProperty rdf:ID="name_attribute">
        <rdfs:domain rdf:resource="#name_entity"/>
        <rdfs:range rdf:resource="#type_name"/>
</owl:ObjectProperty>
```

Рис. 9. Определение в OWL свойств-атрибутов и свойств-значений

Здесь имя свойства будет соответствовать имени атрибута, а отношение будет связывать исходную сущность и некоторый класс. Проблема повторения имен (например, названия атрибута-свойства и сущности-класса, с которой существует связь), решается введением точечной нотации, то есть свойство будет называться в онтологии name_entity.name_attribute. Если атрибут имеет значения некоторого типа данных, то, по смыслу, должно быть свойство-значение, но, так как в Epicentre типы имеют более сложную структуру, фактически получаем те же свойства-объекты.

Указание степени связи между классами, а также необязательность (OPTIONAL) можно реализовать с помощью понятия кардинальности в OWL.

```
ENTITY well_test_open_period_recovery
        SUBTYPE OF ( well_test_recovery );
        time_to_surface: OPTIONAL ndt_time;
END_ENTITY;
```

Рис. 10. Ограничение на свойство на языке EXPRESS

```
<owl:Class rdf:ID="well_test_open_period_recovery">
        <rdfs:subClassOf rdf:resource="#well_test_recovery"/>
        <rdfs:subClassOf>
                <owl:Restriction>
                        <owl:onProperty>
                                <owl:ObjectProperty rdf:resource=
                                "#well_test_open_period_recovery.time_to_surface"/>
                        </owl:onProperty>
                        <owl:minCardinality>0</owl:minCardinality>
                        <owl:maxCardinality>1</owl:maxCardinality>
                        <owl:cardinality>0</owl:cardinality>
                </owl:Restriction>
        </rdfs:subClassOf>
</owl:Class>
```

Рис. 11. Реализация с помощью понятия кардинальности на OWL

Так, OPTIONAL означает, что связь может быть и 0, поэтому в описании класса добавляется ограничение на свойство. Например, для сущности well_test_open_period_recovery (рис. 10) описание в OWL будет следующим (см. рис. 11). Определение самого свойства показано на рис. 12.

```
<owl:ObjectProperty>
        <rdf:about>#well_test_open_period_recovery.time_to_surface
        </rdf:about>
        <rdfs:domain>
                <owl:Class>
                        <rdf:about>#well_test_open_period_recovery</rdf:about>
                </owl:Class>
        </rdfs:domain>
        <rdfs:range>
                <owl:Class>
                        <rdf:about>#ndt_time</rdf:about>
                </owl:Class>
        </rdfs:range>
</owl:ObjectProperty>
```

Рис. 12. Определение свойства на OWL

Определению инверсного атрибута сущности соответствует следующая синтаксическая конструкция (рис. 13):

```
INVERSE
invers_attribute_name: SET[0 :?] OF invers_name_entity
        FOR entity_invers_type;
```

Рис. 13. Определение инверсного атрибута

Данной синтаксической конструкции в OWL можно сопоставить схему, как и в отношении прямого атрибута, только с некоторыми модификациями. Аналогичным образом реализуются и ограничения кардинальности (рис. 14).

```
<owl:ObjectProperty rdf:ID="invers_attribute_name">
        <rdfs:domain rdf:resource="#invers_name_entity"/>
        <rdfs:range rdf:resource="#entity_invers_type"/>
</owl:ObjectProperty>
```

Рис. 14. Ограничения кардинальности

Так, например, у сущности activity есть атрибут activity_alias, который является инверсным к aliased_object, поэтому необходимо обозначить

инверсность путем добавления в определение свойства Type: InverseProperty, а также указать к какому свойству данное свойство является обратным (рис. 15).

```
<owl:ObjectProperty>
        <rdf:about>#activity.activity_alias</rdf:about>
        <owl:type>
                <rdf:about>#InverseProperty</rdf:about>
        </owl:type>
        <owl:inverseOf>
                <owl:ObjectProperty>
                        <rdf:about>#activity_alias.aliased_object</rdf:about>
                </owl:ObjectProperty>
        </owl:inverseOf>
</owl:ObjectProperty>
```

Рис. 15. Определение инверсного атрибута на OWL

Определению уникальности и ограничений на атрибуты сущности соответствует следующая синтаксическая конструкция (рис. 16):

```
UNIQUE
si: name_attribute1, ..., name_attributeN;
WHERE условие
```

Рис. 16. Определение уникальности и ограничений на атрибуты сущности.

Здесь после ключевого слова UNIQUE приведен список уникальных атрибутов, которые являются ключами для данной сущности.

В OWL отсутствуют структурные элементы, которые в полном объеме описывают определение уникальности Epicentre. Поэтому в определение каждого класса на языке OWL добавлено новое свойство с именем name_entity.UNIQUE, в котором перечислены все атрибуты, образующие уникальный ключ. Аналогичным же образом поступили и с WHERE: в определение каждого класса на языке OWL добавлено новое свойство с именем name_entity.WHERE, в котором записаны условия ограничений.

Типы данных в модели Epicentre классифицированы на следующие категории:

- **Simple Data Types** – простые;
- **Simple Pattern Types** – простые структурные;
- **Measured Quantity Types** – метрические численные;
- **Geometry Types** – геометрические.

Для каждой категории данных в онтологии OWL построены отдельные классы, в свойствах которых использовались встроенные типы данных языка OWL.

5. Заключение

Для синтаксического описания модели Epicentre 3.0 корпорации POSC построена формальная LR(1) грамматика, которая использовалась в качестве входного файла генератора синтаксических анализаторов Java Cup.

На основе этой грамматики и схемы конвертации модели Epicentre средствами языка Java, описанной выше, построена онтология предметной области природно-технических объектов на языке OWL. Не все конструкции модели Epicentre могут быть синтаксически выражены средствами языка OWL. Для конвертации такой информации были определены специальные классы и зарезервированные свойства OWL, которые могут быть использованы семантически адекватно модели Epicentre.

Общий объем LR(1) грамматики с встроенной в нее реализацией семантики конвертации составил порядка 30 страниц. Файл с описанием модели Epicentre на языке EXPRESS содержит около 500 страниц. Полученная модель на языке OWL имеет объем около 3500 страниц текста. Таким образом, в данной работе полностью, без потери информации, осуществлен перевод модели Epicentre в OWL-онтологию. Визуализация построенной онтологии предметной области позволяет убедиться в ее корректности.

Корпоративное веб-приложения ПАО «Татнефть», основанное на интеграции реляционных баз данных в виде: универсальной онтологии реляционных баз данных, онтологии предметной области, лингвистического тезауруса, и генерирующего SQL запросы, задаваемые на русском языке, описано в работах [14–18]. Способ реализации извлечения информации из комментариев реляционных баз данных существенно использует методы компьютерной лингвистики; подробно соответствующий алгоритм извлечения знаний изложен в этих же работах.

Методы оптимизации алгоритма построения SQL-запросов, наиболее ресурсоемкая часть которого связана с необходимостью перебора соединений таблиц, рассмотрены в работе [19].

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, проект 18-07-00964.

Литература

1. Гаврилова Т.А., Хорошевский Т.А. Базы знаний интеллектуальных систем // СПб.: Питер, 2001. – 384 с.
2. Дейт К.Д. Введение в системы баз данных // М.: Изд. Дом «Вильямс», 2001. – 72 с.
3. Epicentre v3.0. – URL: http://www.energistics.org/energistics-standards-directory/epicentre-archive .
4. OWL Web Ontology Language. – URL: https://www.w3.org/TR/2004/REC-owl-features-20040210/ .
5. Towards the Semantic Web: Ontology-Driven Knowledge Management. – Chicester, UK: John Wiley & Sons, 2003. – 312 p.

6. The World Wide Web Consortium (W3C). – URL: http://www.w3c.org .
7. RDF 1.1 Concepts and Abstract Syntax. – URL: https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/ .
8. RDF Schema 1.1. – URL: https://www.w3.org/TR/rdf-schema/ .
9. Extensible Markup Language (XML). – URL: https://www.w3.org/XML/ .
10. Льюис Ф., Розенкранц Д., Стирнз Р. Теоретические основы проектирования компиляторов // М.: Мир, 1979. – 654 с.
11. Allmon B.J., Anderson J. Flex on Java // Manning Publications Co. Greenwich, CT, USA, ISBN: 1933988797, 2010. – 264 p.
12. CUP Parser Generator for Java. – URL: https://www.cs.princeton.edu/~appel/modern/java/CUP/ .
13. Protégé. – URL: http://protege.stanford.edu/ .
14. Birialtsev E., Bukharaev N., Gusenkov A. Intelligent search in Big Data // Journal of Physics: Conference Series, Volume 913, conference 1. – Published online: 25 October 2017.
15. Гусенков А.М. Интеллектуальный поиск сложных объектов в массивах больших данных // Электронные библиотеки. – 2016. – Т. 19, № 1. – С. 3–39.
16. Гусенков А., Биряльцев Е., Жибрик О. Интеллектуальный поиск в структурированных массивах информации // LAP LAMBERT Academic Publishing. – Deutschland: OmniScriptum Marceting DEU GmbH, ISBN 978-3-659-76919-1, 2015. – 129 с.
17. Гусенков А.М., Биряльцев Е.В. Интеграция реляционных баз данных на основе онтологий // Ученые записки Казанского государственного университета. Серия Физико-математические науки. – 2007. – Т. 149, кн. 2. – С. 13–34.
18. Gusenkov A., Bukharaev N., Birialtsev E. On ontology based data integration: problems and solutions // Journal of Physics: Conference Series, Volume 1203, conference 1. – 012059. – URL: https://iopscience.iop.org/article/10.1088/1742-6596/1203/1/012059 /meta .
19. Gusenkov A., Bukharaev N. On Semantic Search Algorithm Optimization // New Knowledge in Information Systems and Technologies. WorldCIST'19. Advances in Intelligent Systems and Computing, Volume 930. – Springer, Cham, 2019. – URL: https://link.springer.com/chapter/10.1007/978-3-030-16181-1_45 .

References

1. Gavrilova T.A., Khoroshevskii T.A. Bazy znanii intellektualnykh sistem // SPb.: Piter, 2001. – 384 s.
2. Date C.J. Deit K.D. Vvedenie v sistemy baz dannykh // M.: Izd. Dom «Viliams», 2001. – 72 s.
3. Epicentre v3.0. – URL: http://www.energistics.org/energistics-standards-directory/epicentre-archive .
4. OWL Web Ontology Language. – URL: https://www.w3.org/TR/2004/REC-owl-features-20040210/ .

5. Towards the Semantic Web: Ontology-Driven Knowledge Management. – Chicester, UK: John Wiley & Sons, 2003. – 312 p.
6. The World Wide Web Consortium (W3C). – URL: http://www.w3c.org .
7. RDF 1.1 Concepts and Abstract Syntax. – URL: https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/ .
8. RDF Schema 1.1. – URL: https://www.w3.org/TR/rdf-schema/ .
9. Extensible Markup Language (XML). – URL: https://www.w3.org/XML/ .
10. Lewis P., Rosenkrantz D., Stearns R. Teoreticheskie osnovy proektirovaniia kompiliatorov // M.: Mir, 1979. – 654 s.
11. Allmon B.J., Anderson J. Flex on Java // Manning Publications Co. Greenwich, CT, USA, ISBN: 1933988797, 2010. – 264 p.
12. CUP Parser Generator for Java. – URL: https://www.cs.princeton.edu/~appel/modern/java/CUP/ .
13. Protégé. – URL: http://protege.stanford.edu/ .
14. Birialtsev E., Bukharaev N., Gusenkov A. Intelligent search in Big Data // Journal of Physics: Conference Series, Volume 913, conference 1. – Published online: 25 October 2017.
15. Gusenkov A.M. Intellektualnyi poisk slozhnykh obieektov v massivakh bolshikh dannykh // Elektronnye biblioteki. – 2016. – T. 19, № 1. – S. 3–39.
16. Gusenkov A., Birialtsev E., Zhibrik O. Intellektualnyi poisk v strukturirovannykh massivakh informatsii // LAP LAMBERT Academic Publishing. – Deutschland: OmniScriptum Marceting DEU GmbH, ISBN 978-3-659-76919-1, 2015. – 129 c.
17. Gusenkov A.M., Birialtsev E.V. Integratsiia reliatsionnykh baz dannykh na osnove ontologii // Uchenye zapiski Kazanskogo gosudarstvennogo universiteta. Seriia Fiziko-matematicheskie nauki. – 2007. – T. 149, kn. 2. – S. 13–34.
18. Gusenkov A., Bukharaev N., Birialtsev E. On ontology based data integration: problems and solutions // Journal of Physics: Conference Series, Volume 1203, conference 1. – 012059. – URL: https://iopscience.iop.org/article/10.1088/1742-6596/1203/1/012059 /meta .
19. Gusenkov A., Bukharaev N. On Semantic Search Algorithm Optimization // New Knowledge in Information Systems and Technologies. WorldCIST'19. Advances in Intelligent Systems and Computing, Volume 930. – Springer, Cham, 2019. – URL: https://link.springer.com/chapter/10.1007/978-3-030-16181-1_45 .