УДК 681.3

А.М. Наместников, Р.А. Субхангулов

РАЗРАБОТКА ИНСТРУМЕНТА ИНЖЕНЕРИИ ОНТОЛОГИИ В ИНТЕЛЛЕКТУАЛЬНОМ ПРОЕКТНОМ РЕПОзИТОРИИ

Наместников Алексей Михайлович, *кандидат технических наук, доцент, окончил радиотехнический факультет Ульяновского государственного технического университета. Доцент кафедры «Информационные системы» факультета информационных систем и технологий УлГТУ. Имеет статьи и монографию в области интеллектуальных систем хранения и обработки информации. [e-mail: nam@ulstu.ru].*

Субхангулов Руслан Айратович, *аспирант кафедры «Информационные системы» Ульяновского государственного технического университета, окончил факультет информационных систем и технологий УлГТУ. Имеет статьи в области интеллектуальных систем хранения и обработки информации. [e-mail: subkhangulov-ruslan@yandex.ru].*

Аннотация

Данная статья является результатом исследования, важная часть которого связана с построением онтологии предметной области проектного репозитория и среды ее разработки. В работе рассмотрено понятие онтологии с точки зрения информатики, представлено подробное описание предметно-ориентированного редактора онтологии, разработанного для решения задач интеллектуального анализ проектной документации.

Ключевые слова: онтология, RDF, индексирование, проектный документ, концепт, термин.

Alexey Mikhailovich Namestnikov, *Candidate of Engineering, Associate Professor; graduated from the Faculty of Radioengineering of Ulyanovsk State Technical University; Associate Professor at the Chair 'Information Systems' of the Faculty of Information Systems and Technology of Ulyanovsk State Technical University; author of articles and a monograph in the field of intelligent systems for storage and processing of information. e-mail: nam@ulstu.ru.*

Ruslan Ayratovich Subkhangulov, *post-graduate student at the Chair 'Information Systems' of Ulyanovsk State Technical University; graduated from the Faculty of Information Systems and Technology of Ulyanovsk State Technical University; author of articles in the field of intellectual systems for storage and processing of information. e-mail: subkhangulov-ruslan@yandex.ru.*

Abstract

The present article is a result of researches whose important part is associated with the building of ontology of application of a design repository and the environment of its development. The paper deals with a concept of ontology from the point of view of informatics, presents a detailed description of subject-oriented editor of ontology, developed to solve tasks of intellectual analy-sis of design documents.

Key words: ontology, RDF, indexing, design document, concept, term.

ВВЕДЕНИЕ

В настоящее время на каждом современном предприятии сформирован большой архив проектной документации, который позволяет достаточно быстро предоставлять по запросу пользователя необходимые документы в электронном виде. Формирование к электронному архиву запросов допускается с использованием формализованного языка (такого, например, как SQL) при заранее известных атрибутах: децимальный номер, дата создания документа, автор и т.п. При таком подходе к построению архива проектной документации у проектировщика отсутствует возможность решать слабоформализованные задачи поиска. Такими задачами могут быть полнотекстовый поиск документа, нахождение близкого по содержанию документа, кластеризация всего множества документов и другие. Для решения подобного рода задач применяются интеллектуальные системы, функционирование которых основано на предметно-ориентированных знаниях. Эти знания могут быть представлены в виде онтологии предметной области [1, 4].

При традиционном подходе анализ текстовой документации выполняется путем представления отдельного документа в виде набора «термин-частота» [6]. Решая проектные задачи и учитывая стадии жизненного цикла проектируемой системы, проектировщик с различных точек зрения обращается к содержимому проектного репозитория. Для каждой стадии или этапа жизненного цикла имеет место определенный контекст принятия проектных решений, который может быть представлен в виде онтологии, содержащей понятия (концепты) и связанные с ними термины.

В данной статье представлено описание программной системы, основной функцией которой является построение с использованием модели разметки Resource Description Framework (RDF) предметно-ориентированной онтологии, терминологический уровень которой формируется автоматически на основе текстовых описаний концептов.

1 МОДЕЛЬ ОНТОЛОГИИ В ИНТЕЛЛЕКТУАЛЬНОМ РЕПОЗИТОРИИ

Одно из определений онтологии, сформулированное известным исследователем в данной области Грубером, - это спецификация концептуализации. Данное определение наиболее часто используется в информационных технологиях. Концептуализация - это структура реальности, рассматриваемая независимо от словаря предметной области и конкретной ситуации [1].

Для реализации таких функций интеллектуального проектного репозитория (ИПР), как кластеризация, классификация и информационный поиск, необходимо привлекать априорные знания о предметной области. Поскольку в качестве основного информационного ресурса в ИПР рассматривается проектный документ, содержащий структурированный текст [1], онтологическая модель предметной области проектной организации должна включать как структурированные предметные знания, так и лингвистические знания.

Структуру онтологии можно разделить на две части. В первую часть, которая определяет структурированные предметные знания, входят следующие уровни:

- корневой уровень;
- уровень типов;
- структурный уровень;
- уровень проектов.

Во вторую часть онтологии, которая представляет лингвистические знания, входят уровни:

- уровень понятий;
- уровень терминов.

Формально онтологию представим следующим образом:

$$O = <r,\ T,\ S,\ C,\ W,\ R>, \quad (1)$$

где r - корневая вершина онтологии, соответствующая классу проектных документов;

$T = \{t_1, t_2, \ldots, t\}$ - множество типов проектных документов ИПР, t_i - ***i-й*** тип проектного документа;

$S = S^1 \cup S^2 \cup \ldots \cup S^n$ - множество структур документов;

$C = \{c_1, c_2, \ldots, c_k\}$ - множество понятий предметной области ИПР;

$W = \{w_1, w_2, \ldots, w_l\}$ - множество терминов предметной области ИПР;

R - множество отношений, определяемое следующим образом:

$$R = R_G \cup R_C \cup R_A$$

где R_g - антисимметричное, транзитивное, нерефлексивное бинарное отношение обобщения;

R_C - бинарное транзитивное отношение композиции «часть-целое»;

R_A – конечное множество ассоциативных отношений.

Для структур документов справедливо соотношение:

$\forall t_t 3 S^i : i = j.$

Другими словами, для каждого типа документа в онтологии определена его структура.

Множество S^1 содержит разделы и подразделы проектного документа типа t_j. В общем случае имеет место следующее неравенство:

$$S^i \cap S^j \neq \emptyset,$$

что означает допустимость пересечения структурных элементов между различными типами документов.

Существуют специализированные инструменты, поддерживающие различные форматы описания онтологий. Так в работе [3] представлен обзор широко известных редакторов онтологий. Выделим некоторые функции, которые имеют значение при использовании редактора онтологий как компоненты интеллектуального репозитория:

- функция автоматического формирования терминологического уровня онтологии на основе текстовых описаний понятий;
- функция вычисления весовых коэффициентов при отношениях ассоциаций между терминами и понятиями онтологии.

Далее описывается архитектура редактора предметно-ориентированной онтологии, позволяющего не только сформировать структуру онтологии, но и реализовать вышеприведенные функции.

2 СТРУКТУРНО-ФУНКЦИОНАЛЬНОЕ РЕШЕНИЕ РЕДАКТОРА ОНТОЛОГИИ

В качестве хранилища онтологий используется Java фреймворк Sesame с веб-сервером Apache Tomcat. Хранилище Sesame - открытая (open source) база данных RDF с поддержкой логического вывода по RDF-тройкам и запросам. Данный программный продукт предлагает большой набор инструментов для разработчиков с использованием RDF и RDF Schema. На рисунке 1 представлена архитектура разработанного редактора онтологии.

В подсистеме редактирования онтологии (рис. 1) представлены основные модули системы. Модуль формирования схемы онтологии и модуль формирования содержания онтологии предоставляют пользователю интерфейс для создания схемы онтологии и набора экземпляров классов онтологии. функция формирования терминологического окружения понятий онтологии реализована в модуле онтологической индексации. В модуле взаимодействия с RDF-хранилищем Sesame реализованы функции, позволяющие сохранять онтологию в хранилище Sesame и загружать онтологию из данного хранилища. Программный интерфейс взаимодействия с базами данных является компонентом Sesame и предлагает разработчикам средства для организации взаимодействия со сторонними программными системами. Модуль визуализации онтологии позволяет производить редактирование создаваемой онтологии, представляя ее в виде графа,

Рис. 1. Архитектура редактора онтологии

внешний вид которого можно настраивать (форму узлов, дуг и цветовую гамму).

Детально рассмотрим процесс формирования связей между понятийным и терминологическим уровнями. Понятийный уровень формируется пользователем посредством ввода концептов. Терминологический уровень формируется автоматически, на основе документов, называемых текстовыми входами понятия, и метода вычисления весов терминов. Терминологический уровень формируется из текстового входа, выполняя следующие операции:

1. Поступление на вход документов, характеризующих концепты.
2. Формирование терминологического уровня.

2.1.Сканирование введенного контекста.

2.2.Удаление стоп-слов.

2.3.Выделение основ терминов путем удаления окончания слов.

3. Вычисление частоты встречаемости терма.

В программе реализованы два варианта вычисления нормированной частоты встречаемости терма f:

1) Нормированная частота термина t в документе d вычисляется по формуле [6]:

$$ntf_{t,d} = a + \frac{tf}{tfmatW} \quad (2)$$

где ntf_{td} - вес термина t в документе d;

$tf\hat{}(d) = max\hat{}tf\hat{}_{d}$ - максимальная величина tf в документе d;

a - сглаживающий коэффициент, принимающий значение между нулем и единицей (как правило, устанавливают равным 0,4). Роль данного параметра заключается в уменьшении вклада второго члена выражения (2). Нормировка частоты термина по максимуму предназначена для того, чтобы избежать следующей аномалии [6]: в крупных документах наблюдаются более высокие частоты терминов, так как в таких документах чаще содержатся повторяющиеся слова.

2) Нормированная частота термина t в документе d вычисляется по формуле [6]:

$$ntf_{t4} = 1 + \log(tf_{t4}) \Big/ \sqrt{w_1^2 + w_2^2 + \ldots + w_m^2}, \quad (3)$$

где w^2 - евклидова нормировка.

В качестве формата описания онтологии используется RDF - разработанная консорциумом Всемирной паутины (W3C) модель для представления данных и метаданных. RDF представляет собой утверждения о ресурсах в виде, пригодном для машинной обработки, и является частью концепции семантической паутины.

Ресурсом в RDF может быть любая сущность как информационная, так и неинформационная. Утверждение, высказываемое о ресурсе, имеет вид «субъект — предикат — объект» и называется триплетом.

Предикат (в контексте RDF его обычно называют свойством) может пониматься либо как атрибут, либо как бинарное отношение между двумя ресурсами. Но RDF сама по себе не предоставляет никаких механизмов ни для описания атрибутов ресурсов, ни для определения отношений между ними.

Для этого предназначена RDFS (RDF Schema) - модель описания словарей для RDF. RDF Schema определяет классы, свойства и другие ресурсы. RDFS является семантическим расширением RDF. Она предоставляет механизмы для описания групп связанных ресурсов и отношений между этими ресурсами. Все определения RDFS выражены с помощью RDF (поэтому RDF и называется «самоописывающейся» моделью). Новые термины, вводимые RDFS, такие как «домен», «диапазон» свойства, являются ресурсами RDF.

При разработке онтологии необходимо выполнить следующие этапы:

1. Определение классов в онтологии.
2. Организация классов в некоторую иерархию (базовый класс -> подкласс).
3. Определение свойств и допустимых значений.
4. Создание экземпляров классов и заполнение значениями свойств.

При работе в рассматриваемой программной системе необходимо выполнить все этапы с учетом предлагаемой функциональности программного продукта. С первого по третий этап в программе формируется схема будущей онтологии, в которой определяются классы и свойства онтологии, а также отношения между классами. На четвертом этапе происходит заполнение онтологии на основе созданного ранее описания в виде RDF Schema, т.е. создание экземпляров классов. Результатами работы программы являются разработанная схема RDF и онтология, представленная в формате RDF.

3 РЕАЛИЗАЦИЯ ОНТОЛОГИИ В ИНТЕЛЛЕКТУАЛЬНОМ ПРОЕКТНОМ РЕПОЗИТОРИИ

В разрабатываемой предметно-ориентированной онтологии интеллектуального проектного репозитория RDF Schema определяет классы и свойства следующих видов:

• понятия описываются с помощью класса <rdfs:Class rdf:ID="Concept"/>,

• термы описываются с помощью класса <rdfs:Class rdf:ID="Term"/>,

• концепт-термы (объекты, представляющие отношения между понятиями и термами) описываются с помощью класса <rdfs:Class rdf:ID="ConceptTerm"/>.

При разработке онтологии необходимо соблюдать ряд правил, которые в дальнейшем позволят использовать сформированную онтологию в процессе онтологически-ориентированной индексации [5].

Отношение «Обобщение» реализовано посредством описания свойства «Generalization»:

```
<rdf:Property rdf:ID="Generalization">
<rdfs:domain rdf:resource="#Concept"/>
<rdfs:range rdf:resource="#Concept"/>
</rdf:Property>.
```

Пример экземпляров классов, находящихся в отношении «Обобщение»:

```
<Concept ^:Ю="Субъект" />
<Concept ^:Ю="Эксперт">
Generalization rdf:resource="#Субъект" />
</Concept>
<Concept ^:Ю="Проектировщик">
Generalization rdf:resource="#Субъект" />
</Concept>.
```

Отношение «Включение» реализовано посредством описания свойства «Inclusion»:

```
<rdf:Property rdf:ID="Inclusion">
<rdfs:domain rdf:resource="#Concept" />
<rdfs:range rdf:resource="#Concept" />
</rdf:Property>.
```

Пример экземпляров классов, находящихся в отношении «Включение»:

```
<Concept rdf:ID="Rational_Unified_Process_RUP">
<Concept rdf:Ю="Рабочие_процессы_RUP">
<Inclusion rdf:resource="#Rational_Unified_Process_RUP" />
</Concept>
<Concept ^:Ю="Основные_процессы">
<Inclusion rdf:resource="#Рабочие_процессы_RUP" />
</Concept>.
```

Отношения между концепт-термами и понятиями реализуются с помощью свойства:

```
<rdf:Property rdf:ID="AssociatedWithConcept">
<rdfs:domain rdf:resource="#ConceptTerm"/>
<rdfs:range rdf:resource="#Concept"/>
</rdf:Property>.
```

Отношения между концепт-термами и термами реализуются с помощью свойства:

```
<rdf:Property rdf:ID="AssociatedWithTerm">
<rdfs:domain rdf:resource="#ConceptTerm"/>
<rdfs:range rdf:resource="#Term"/>
</rdf:Property>.
```

Относительная частота встречаемости терма в понятии описывается с помощью свойства:

```
<rdf:Property rdf:ID="HasAFreq">
<rdfs:domain rdf:resource="#ConceptTerm"/>
<rdfs:range rdf:resource="&xsd;float"/>
</rdf:Property>.
```

В ходе разработки онтологии на основе представленной схемы была создана онтология предметной области «Проектирование информационных систем» интеллектуального проектного репозитория, фрагмент которой имеет следующий вид:

```
<Concept ^:Ю="Субъект"/>
<Concept ^:Ю="Эксперт">
Generalization rdf:resource="#Субъект"/>
</Concept>
<Concept ^:Ю="Проектировщик">
Generalization rdf:resource="#Субъект"/>
</Concept>
<Concept ^:Ю="Тестировщик">
Generalization rdf:resource="#Субъект"/>
</Concept>
<Concept ^:Ю="Программист">
Generalization rdf:resource="#Субъект"/>
</Concept>
<Concept ^:Ю="Объект"/>.
```

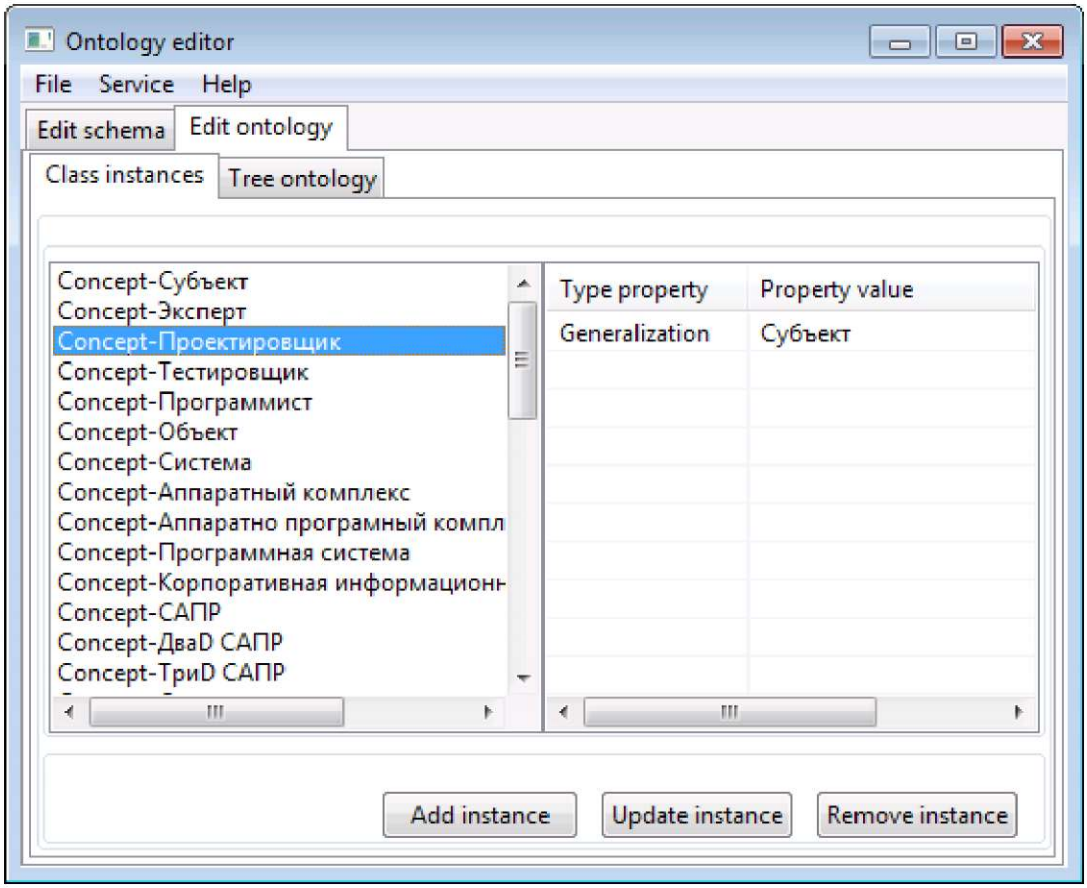

Рис. 2. Фрагмент списка экземпляров классов онтологии

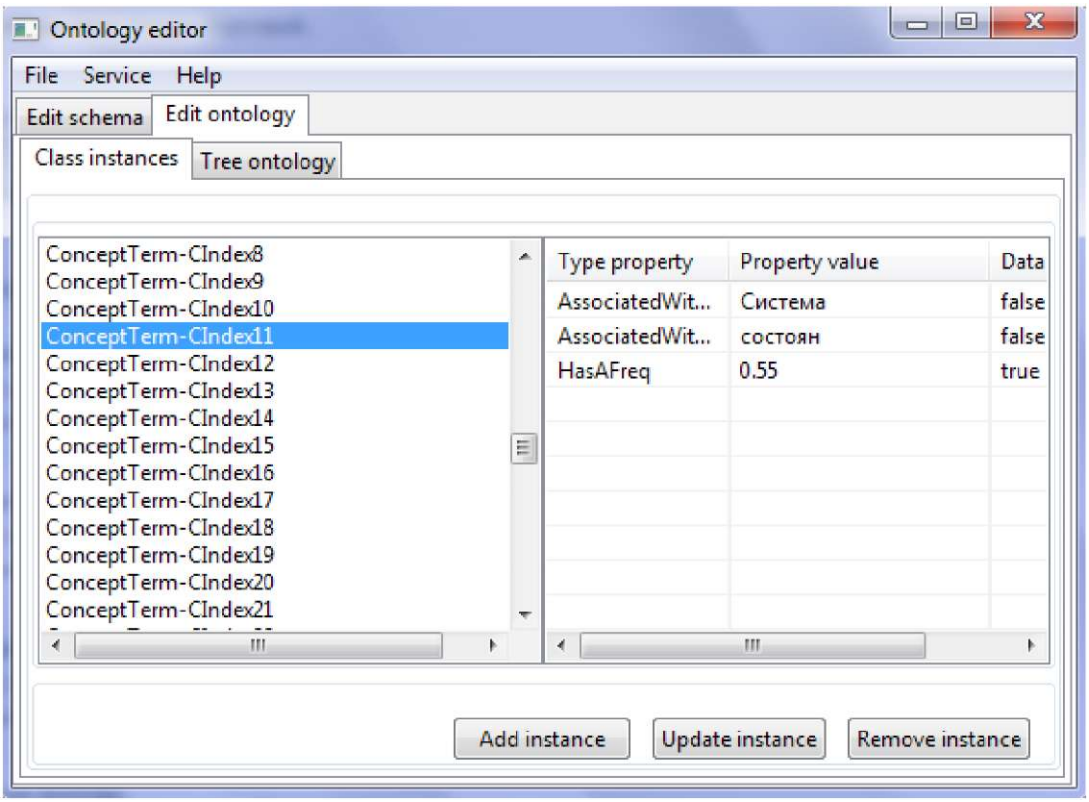

Рис. 3. Результат выполнения индексации

Пользовательский интерфейс редактора онтологии предоставляет пользователю доступ к инструментам для создания классов и свойств, которые впоследствии будут использованы при формировании экземпляров классов. На рисунке 2 представлен фрагмент списка экземпляров классов сформированной онтологии с указанием свойств классов.

Для создания связей между понятийным и терминологическим уровнями необходимо выполнить процедуру индексирования, при которой программа открывает диалоговое окно для отбора классов верхнего уровня (понятия), которые будут индексироваться, и классов нижнего уровня (терминов), формирующихся после выполнения индексирования.

Программа автоматически создает экземпляры классов ConceptTerm, которые описывают отношения между экземплярами классов Concept и Term в разрабатываемой онтологии с автоматическим расчетом частоты встречаемости термов.

Результаты выполнения индексации представлены на рисунке 3.

ЗАКЛЮЧЕНИЕ

Представленное в статье описание структурно-функционального решения редактора онтологий и соответствующий прототип программной системы выполнены в виде, позволяющем использовать данную систему в качестве компонента интеллектуального проектного репозитория. Программа реализует функцию сохранения онтологии в хранилище Sesame, а также в текстовом файле в виде RDF-троек (субъект - объект - предикат). Реализован модуль онтологической индексации, устанавливающий связь между понятийным и терминологическим уровнями онтологии. Рассмотренный редактор онтологий был применен для построения модели предметной области «Проектирование информационных систем», содержащей 43 концепта (понятия) и более 800 терминов. При использовании данной онтологии экспериментальным путем доказано улучшение качества кластеризации тестового множества проектных документов.

СПИСОК ЛИТЕРАТУРЫ

1. Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д. Онтологии и тезаурусы: модели, инструменты, приложения. - Режим доступа: www.intuit.ru/department/expert/ontoth/7/1.htm.

2. Бениаминов Е.М. Некоторые проблемы широкого внедрения онтологий в IT и направления их решений. - Режим доступа: www.beniaminov.rsuh.ru/BeniaminovOntoNew.pdf.

3. Овдей О.М., Проскудина Г.Ю. Обзор инструментов инженерии онтологий. - Режим доступа: www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part4/op.

4. Ермаков А.Е. Автоматизация онтологического инжиниринга в системах извлечения знаний из текста. - Режим доступа: www.rco.ru/attach/news/5790/onthologies.pdf.

5. Наместников А.М., Филиппов А.А. Концептуальная индексация проектных документов //Автоматизация процессов управления. - 2010. - №2(20). - С. 34-39.

6. Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск. - М.: Вильямс, 2011.