УДК 371.3:004.048

В.А. Филатов, С.С. Щербак, А.А. Хайрова

Харьковский национальный университет радиоэлектроники

РАЗРАБОТКА ВЫСОКОЭФФЕКТИВНЫХ СРЕДСТВ СОЗДАНИЯ И ОБРАБОТКИ ОНТОЛОГИЧЕСКИХ БАЗ ЗНАНИЙ

Тематикой статьи является рассмотрение проблемы хранения больших и стабильных онтологий в пространственной базе данных, а так же разработка информационно-поисковой системы построения и сопровождения онтологий. Статья описывает метод хранения больших и стабильных онтологий с помощью систем обработки пространственных данных.

Ontology, RDF (Recourse Description Framework), RDFS (RDF Schema), OWL (Ontology Web Language), Oracle Database 10g, Oracle Spatial

Введение

С расширением Internet работа поисковых систем все более усложнялась: чтобы обеспечить достаточный охват, им приходилось индексировать и обрабатывать все больший объем информации. Но главная сложность заключалась даже не в увеличении количества индексируемых сайтов, а в том, чтобы обеспечить релевантные ответы на поисковые запросы пользователей, т.е. выдавать пользователям ссылки на те ресурсы, которые, по их мнению, соответствуют тому, что они искали.

Современное состояние дел в отдельных областях компьютерных наук диктует необходимость привлечения методов инженерии знаний для решения широкого класса практических задач. Ярким примером тому является инициатива Semantic Web, основная цель которой – наделить огромные массивы данных, опубликованных в сети Internet большей осмысленностью, повысить удобство работы с этой информацией. Одним из главных достижений проекта Semantic Web стала разработка стандарта описания онтологий – OWL (Ontology Web Language), благодаря чему множество инженеров по знаниям, программистов и экспертов получили возможность использовать общие правила представления, хранения и обработки онтологий [1].

Существуют различные толкования самого понятия онтологии. В данной работе под онтологией понимается спецификация некоторой предметной области, ее концептуальное описание в виде формализованного представления, которое включает словарь терминов предметной области и логические выражения, описывающее взаимосвязи этих понятий. Таким образом, онтология некоторой предметной области представляет собой тезаурус понятий этой предметной области, обеспечивающий возможность толкования терминов предметной области посредст-

вом интерпретации таких типов парадигматических отношений как «часть-целое», «класс-подкласс» и некоторых видов ассоциативных связей [2].

На волне интереса к онтологиям были созданы инструментальные средства и механизмы, специально ориентированные на широкое применение онтологий в задачах интеллектуального поиска, классификации, выявления несогласованности в данных, моделирования поведения интеллектуальных агентов и обработки данных. Однако даже наличие хорошего инструментального окружения не снимает проблем, связанных с трудностью проектирования и построения самих онтологий, а автоматизация процесса извлечения онтологий, как и в целом, задача извлечения знаний, и по настоящее время не имеют своего эффективного решения. Тем ценнее становятся уже разработанные онтологии и опыт их использования для решения широкого круга задач.

В процессе создания современных интеллектуальных информационных систем зачастую требуется интеграция знаний из разнородных источников и, как следствие, эффективное решение задач, связанных с тиражированием знаний. По-прежнему не имеет своего удовлетворительного решения проблема автоматизации процесса выбора адекватного специфике конкретной проблемной области и принятого в ней стиля рассуждения экспертов средства представления знаний. Поэтому и по сей день актуальны исследования, направленные на разработку такого подхода к представлению и тиражированию знаний, который с одной стороны позволял бы наиболее адекватно учитывать специфику проблемной области, а с другой – представлять и использовать знания в некотором унифицированном виде [3].

Актуальность и цель работы. Онтологические модели за время исследований в этой области претерпели значительное развитие. В настоящее время для создания и поддержки онтологий существует целый ряд инструментов, которые помимо общих функций редактирования и просмотра выполняют поддержку документирования онтологий, импорт и экспорт онтологий разных форматов и языков, поддержку графического редактирования, управление библиотеками онтологий и т.д.

Данные инструментарии построения онтологий обладают рядом существенных недостатков. Большинство инструментов хранит свои онтологии в текстовых файлах, что ограничивает размер онтологий, имеют низкую производительность, нужна дополнительная разработка алгоритмов для удобства работы с хранимыми метаданными, обладают избыточностью функций, что затрудняет работу пользователя.

Исходя из особенностей анализа недостатков существующих инструментариев, будем говорить, что тематикой статьи является рассмотрение проблемы хранения больших и стабильных онтологий в пространственной базе данных, а так же разработка информационно-поисковой системы построения и сопровождения онтологий.

В статье рассматривается способ хранения больших и стабильных онтологий с помощью систем управления пространственными базами данных. При реализации данного модуля были учтены следующие свойства онтологий: в онтологиях знания формализуются в виде описаний предметной области с помощью иерархии классов; для каждого класса задается свой набор свойств и объектов; свойства в онтологиях имеют область определения – класс, для которого задается это свойство, а также область значений.

В зависимости от областей значений свойства делятся на два типа: т-свойства (значениями заданного типа данных или набором предопределенных значений) и о-свойства (значениями которых являются объекты заданного класса).

Можно сказать, что иерархическая структура онтологий проецируется на пространственную структуру баз данных [4]. Следует понимать, что онтологии могут быть очень большими - в некоторых из них помимо сложной иерархии с множеством классов и свойств, могут храниться миллионы и миллионы объектов. В совокупности с ситуацией, когда в одной базе хранится множество онтологий, скорость выполнения запросов к базе данных сильно уменьшается. Так как рассматриваемая база данных является составной частью большого проекта, в котором обращение к базе происходит довольно часто, то от скорости выполнения запросов к базе напрямую зависит скорость работы приложения в целом. Методы приложения должны позволить помещать в пространственную базу данных информацию о предметной области в виде онтологий, а также обеспечивать высокую скорость выполнения запросов. В нем присутствуют методы работы с онтологиями, классами, объектами и свойствами объектов онтологий, а также управления сессией к базе данных.

Подводя итог, следует обратить внимание, что предлагаемый в данной статье способ хранения и сопровождения онтологий в пространственной базе данных и методы работы с ним являются универсальными, что позволяет широко использовать результаты предлагаемой работы в исследовательской и прикладной деятельности.

1. Семантическое описание данных

Онтологии разрабатываются и могут быть использованы при решении различных задач, в том числе для совместного применения людьми или программными агентами, для возможности накопления и повторного использования знаний в предметной области, для создания моделей и программ, оперирующих онтологиями, а не жестко заданными структурами данных, для анализа знаний в предметной области.

Для более интеллектуального обобщения разделов информации web-порталам необходимо определить свою онтологию, которая должна описывать терминологию, использующуюся в содержимом web-портала, и аксиомы, задающие правила исполь-

зования этих терминов в контексте других терминов. Совокупность онтологий и аксиом представляют из себя модель описания данных.

Базовый строительный блок модели данных - утверждение, представляющее собой тройку: ресурс, именованное свойство и его значение. В терминологии RDF эти три части утверждения называются соответственно: субъект, предикат и объект [5].

Ресурсом называют все, что описывается средствами RDF. Это может быть обыкновенная Web-страница или какая-то ее часть, например, отдельный элемент HTML или XML разметки, являющийся частью описываемого документа. Также ресурсом может быть целая коллекция страниц, например, отдельно взятый Web-сайт. И, наконец, в качестве ресурса может выступать нечто, не являющееся доступным непосредственно через Интернет, например, произвольный предмет из мира вещей. Одним словом, все, чему можно приписать некоторый URI (универсальный идентификатор) или URI с добавлением внутреннего имени объекта (имени якоря в HTML) может стать ресурсом и быть описано при помощи RDF.

Под свойством следует понимать некий аспект, характеристику, атрибут или отношение, используемое для описания ресурса. Каждое свойство имеет свой специфический смысл, допустимые значения, тип ресурсов, к которым оно может быть применено, а также отношения с другими свойствами.

Согласно спецификации, значение свойства может иметь один из двух типов. Первый – это ресурс, задаваемый некоторым URI. Второй тип – литерал – есть некоторое текстовое значение характеристики. Впрочем, литерал может выражать собой значение любого примитивного типа данных, присутствующего в XML. Его тест также может содержать в себе некую разметку, например, XML, но отличительной особенностью такой разметки является то, что она не обрабатывается RDF-процессором и воспринимается как обычная строка [6].

Реальное значение RDF невозможно оценить, пока он используется для внутренних целей отдельно взятого приложения. Польза от внедрения RDF будет тогда, когда он станет средством межпрограммного взаимодействия, обмена данными, когда машины получат способность комбинировать информацию, полученную из различных источников, тем самым, получая какую-то новую информацию. Чем больше приложений в Интернете смогут работать с данными, тем выше станет их ценность.

2. Описание технологий для разработки и сопровождения онтологий

Статус рекомендации W3C и наличие готовых интероперабельных программных решений делают технологии Semantic Web более привлекательными, чем другие технологические решения инженерии знаний [7]. Ключевым моментом данного подхода является то, что кроме ресурсов в нашей базе данных хранятся так же метаданные для описания объектов хранилища и управления ими, метаданные описаны с помощь языка RDFS (рис. 1).

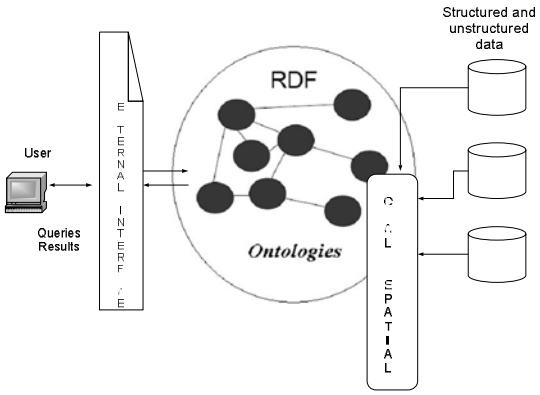

Рис. 1. Модель хранения метаданных

Данный подход основан на Oracle Spatial - технологии СУБД Oracle Database 10g, включающие дополнительные возможности по обработке пространственных данных для поддержки пространственных сервисов, различного рода ГИС-приложений, предназначенных для обработки или предоставления информации о местонахождении объектов и других информационных систем.

СУБД Oracle 10g включает поддержку RDF/RDFS, давая возможность разработчикам приложений использовать преимущества платформы семантической организации данных. Прикладные разработчики могут дополнять значение к данным и метаданным, определяя новые наборы термов и отношений между ними. Эти наборы термов ("онтологии") более приспособлены для осуществления запросов и анализа, основанного на семантическом подходе, чем обычные наборы данных. Онтологические наборы данных, часто содержащие миллионы элементов данных и отношений между ними, которые могут быть сгруппированы в триплеты, используя новую RDF модель данных. Oracle допускает расширение биллионам триплетов для удовлетворения требований большинства приложений [8]. Какие же собственно принципы хранения RDF в Oracle Spatial 10g?

– RDF данные хранятся как направленный, логический граф;

– Субъекты и объекты отображаются как узлы, а предикаты как связи, у которых субъект является начальным узлом, а объект конечным;

– Связи представляют из себя полный RDF триплет.

– Oracle Spatial RDF Модель данных

– RDF Модель данных поддерживает три типа объектов базы данных:

– Модель (RDF граф, состоящий из набора триплетов);

– База правил (набор правил);

– Индекс правила (направленный RDF граф).

Использование предложенной технологии позволяет разработчикам портала создавать единое унифицированное представление данных во всех приложениях, что позволит точно находить необходимую информацию, упростит корпоративную интеграцию и интеграцию данных, сократит избыточность данных и обеспечит единство семантических значений во всех приложениях. Все это, в свою очередь, облегчает разработку, поддержку и обновление приложений в пределах корпорации.

Основными преимуществами использования Oracle Spatial 10g [8]:

– Интегрирование данных из разных источников без применения программирования;

– Поддержка децентрализованного управления данными;

– Поддержка всех RDF типов данных;

– SQL поиск и восстановление RDF моделей;

– Осуществление запросов к RDF моделям, с использованием схемы графа;

– Сочетание запросов RDF с другими SQL операторами;

– Логический вывод, основанный на RDFS (RDF схемы) правилах;

– Логический вывод, основанный на правилах, определяемых пользователем в приложении.

3. Разработка элементов программного обеспечения системы

В основе предлагаемого подхода – активное использование метазнаний не только для описания синтаксиса и семантики языка представления знаний, но и для формирования онтологий, описывающих основные виды взаимосвязей между понятиями проблемной области, а также модель пользователя системы. Однако, как уже отмечалось во введении, проектирование и разработка онтологий, то есть онтологический инжиниринг, не является тривиальной задачей. Он требует от разработчиков профессионального владения технологиями инженерии знаний — от методов извлечения знаний до их структурирования и формализации [9].

На сегодняшний день для большинства инструментальных средств построения онтологий характерно следующее: во-первых, большая часть подобных систем хоть и имеет визуальную составляющую, однако некоторые конструкции приходится набирать вручную, что повышает уровень требований к разработчику онтологий – перед тем, как приступать непосредственно к своей работе, эксперт вынужден тратить время на изучение языка представления знаний; во-вторых, часть инструментальных средств реализуют определенную функциональность для выполнения запросов к онтологиям, но, к сожалению, не имеют унифицированного интерфейса для формирования и выполнения запросов из внешних приложений. В-третьих, практически нет свободно распространяемых и ориентированных на конечного пользователя редакторов онтологий, что, естественно, замедляет развитие всего направления онтологического инжиниринга [10].

При разработке инструментальной среды разработчика онтологий мы постарались нивелировать минусы аналогов и перенять их достоинства. Любой объект онтологии имеет графическое представление (не только классы и индивиды, но и свойства, связи и др.). Система является независимым приложением, которое способно выступать в качестве сервера онтологий. В настоящее время редактор полностью поддерживает конструкции языка описания онтологий RDF, ведется работа по расширению ее возможностей до языка OWL. На рис. 2 приведена архитектура приложения.

В данной статье мы уделили особое внимание разработке эффективного приложения обработки, создания и управления онтологиями. Заметим, что каждый экземпляр в мире онтологий является членом класса THING, т.о. каждый определенный нами класс автоматически является подклассом THING.

Работа пользователя в системе начинается с создания новой модели данных. В данной модели пользователь может создавать свою собственную онтологию предметной области и в дальнейшем редактировать ее не затрагивая другие модели. В тоже время он имеет возможность в любой момент изменить текущую модель данных и приступить к сопровождению другой онтологии или вообще удалить модель из базы (рис. 3).

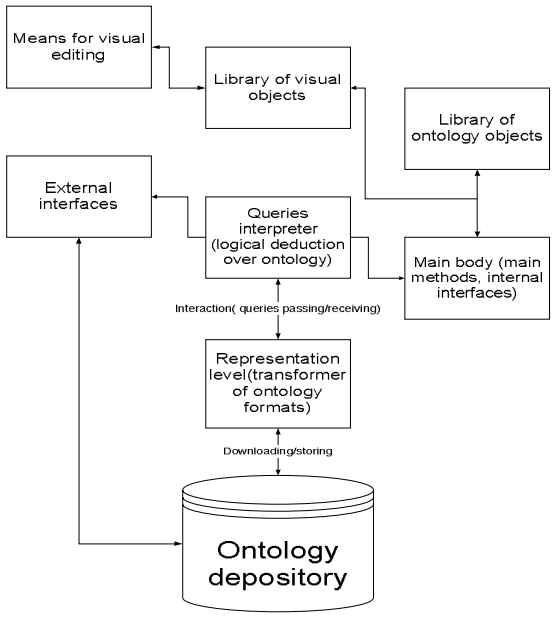

Рис. 2. Архитектура разработанной системы

Для создания онтологии после создания модели пользователь переходит на следующую вкладку «Классы». В левой части страницы расположена иерархия классов в виде дерева реализованного посредством технологии ajax.

При выборе любого класса в правой части окна отображаются свойства, принадлежащие данному

классу, а так же экземпляры относящиеся к нему (рис. 3). Здесь имеется возможность добавления и удаления класса. Так же на этой странице пользователь может привязывать свойства к выбранному классу и добавлять или удалять экземпляры класса. Все изменения заносятся в базу в качестве RDF – триплетов.

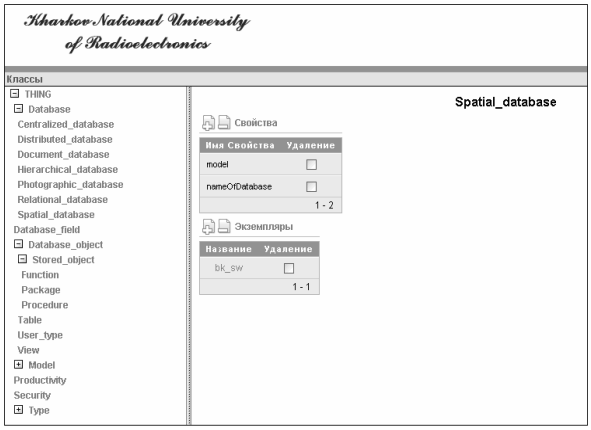

Рис. 3. Представление классов, их экземпляров и свойств

На закладке «Свойства» отображается дерево свойств, опять же таки реализация основана на технологии ajax. В данном дереве отображены родительские отношения свойств друг другу, т.е. иерархия свойств. При выборе любого свойства отображается дерево экземпляров, показывающее их отношения между собой по данному.

На данной странице имеется возможность добавления нового свойства, удаления свойства и его редактирование. В редактировании свойства пользователь может изменить тип свойства (float, string, int, class), классы к которым данное свойство может быть привязано, а также родителей свойства (рис. 4).

Реализована возможность пользователя самостоятельно разрабатывать запросы к базе данных. Здесь мы можем совмещать обычные sql запросы с sparql запросами. Рассмотрим следующий пример: узнать все базы данных относящиеся к классу пространственным базам данных и получить дату создания базы из другой таблицы. Данный запрос будет выглядеть следующим образом:

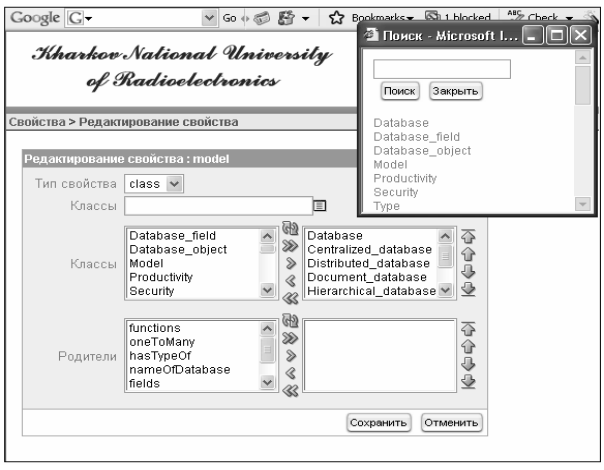

Рис. 4. Окно редактирования свойства

```
SELECT database_inf.createdate  FROM database_inf,
TABLE(SDO_RDF_MATCH(
        '(?m rdf:type :Spatial_database) (?m
:nameOfDatabase ?name)',
         SDO_RDF_Models('bk_sw_model'),
         null,

SDO_RDF_Aliases(SDO_RDF_Alias(",sm_pkg.user_alias)),
         null)) t WHERE
sm_pkg.user_alias||database_inf.name=t.NAME
```

С помощью функции SDO_RDF_MATCH мы получаем доступ к нашей базе данных RDF триплетов. Первым параметром является основа sparql запроса, в котором мы конкретно указываем, что хотим получить, второй параметр – это модель к которой мы обращаемся. Третьим параметром является база правил, с помощью которой существует возможность производить логический вывод. Четвертый параметр – это пространство имен модели. Пятый параметр – это фильтр, который является одним из параметров sparql запроса. Рекомендуется для начала использовать null, т.к. для использования фильтра его нужно добавить в базу правил, которая будет реализована в следующей версии приложения.

Вся онтология определенной области, как упоминалось ранее, хранится в базе данных в виде RDF триплетов. Но триплеты, хранящиеся в какой-то определенной базе, никакой пользы не несут, т.к. основой Semantic Web являются поисковые агенты, на запросы которых они получают своего рода описание определенного ресурса в виде текстового файла, описанного на owl либо, как в нашем случае на RDF языке. В нашей системе реализован модуль, который выдает часть RDF документа, описывающего запрашиваемый объект, по запросу интеллектуального агента или другой системы (запрос производится посредством http протокола). Модуль так же предоставляет список всех объектов, информация о которых хранится в системе

Адрес, по которому агент из базы запрашивает rdf документ, выглядит следующим образом:

http://<ИМЯ_ХОСТА>/<DatabaseAccessDescription>/<НАЗВАНИЕ_ПРОЦЕДУРЫ>< ПАРАМЕТРЫ ЗАПРОСА>

В результате запроса http://localhost/apex/sw-agent?p=bk_sw возвращается следующий rdf-документ:

```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-
rdf-syntax-ns#
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:model="http://swhost.kture/swstore/dbl#">
<rdf:Description
rdf:about="http://swhost.kture/swstore/dbl#bk_sw">
<rdf:type
rdf:resource="http://swhost.kture/swstore/dbl#Spatial_databas
e"/>
<model:model
rdf:resource="http://swhost.kture/swstore/dbl#relational_mode
"/>
<model:nameOfDatabase>bk_sw</model:nameOfData
base>
</rdf:Description>
</rdf:RDF>
```

Выводы

В данной статье был рассмотрен способ хранения больших и стабильных онтологий, с использованием технологии Spatial СУБД Oracle 10. Использование Oracle Database 10g для управления данными, которые размечены с помощью языка семантической разметки, позволяет выделить ряд преимуществ по сравнению с подходами управления основанными на файлах или на специализированных базах данных. Прежде всего это низкий риск, высокое качество, производительность и безопасность.

В результате анализа существующих систем хранения и разработки онтологических баз знаний были выявлены следующие недостатки [11]:

- хранение данных в файлах;
- низкая производительность;
- разработка дополнительных алгоритмов для удобства хранения метаданных;
- избыточность.

Для устранения перечисленных выше недостатков была разработана поисково-информационная система разработки и сопровождения онтологий.

В результате полного описания объектов и их свойств предметная область представлена как сложная иерархическая база знаний над которой можно осуществлять «интеллектуальные» операции, такие как семантический поиск и определение целостности и достоверности данных.

Основными преимуществами разработанной системы являются:

- удобное хранение онтологий в пространственной базе данных;
- предоставление сервером доступа к онтологии через web-сервис, сохранение и извлечение онтологии из хранилища;
- отсутствие необходимости реализации конвертера из формата RDF в реляционную схему и наоборот;
- использование объектной модели онтологий, которая представляет понятия и отношения из онтологии в удобном для пользователя объектно-ориентированном виде;
- удобный пользовательский интерфейс;
- предоставление описания объекта в текстовом виде на запрос пользователя.

Одним из недостатков системы является сложность внедрения. Формат RDF обладает высокой сложностью и не рассчитан на применение рядовыми пользователями Internet. Так же данный формат не позволяет описать предметную область в полном объеме, поэтому в будущем будет предусмотрена поддержка описания онтологий посредством языка OWL [5].

Многим web-разработчикам и программистам бывает сложно освоить RDF и OWL. Кроме того, сам смысл концепции ещё не доведён до широких кругов пользователей. Работа по популяризации Semantic Web ещё не доведена до конца, не хватает практических примеров.

Использование предложенной технологии позволит разработчикам онтологий создавать единое унифицированное представление данных во всех приложениях, что позволит точно находить необходимую информацию, упростит корпоративную интеграцию данных, сократит избыточность данных и обеспечит единство семантических значений во всех приложениях. Все это, в свою очередь, облегчает разработку, поддержку и обновление приложений.

Список литературы

1. Филатов В.А., Хайрова А.А. Технология организации образовательных web-сервисов на основе XMLDB, HTMLDB, ORACLE SPATIAL // Інформаційні технології та комп'ютерна інженерія. – Вінниця: ВНТУ. – 2007. – Вип. 1(18). – С. 240-247.

2. Collins H. Enterprise knowledge portals: next generation portal solutions for dynamic information access, better decision making and maximum results. – N.Y.: AMACOM, 2003. – С. 403.

3. Щербак С.С., Хайрова А.А. Розроблення освітніх web-сервісів як ефективна стратегія розвитку мережевого навчання // НПУ «Інформатизація бізнесу очима молодих: прогресивні технології, наука, підприємництво» (17 – 18 травня 2007р.). – Х.: ХНЕУ, 2007. – С. 73-74.

4. Филатов В.А., Хайрова А.А. Исследование методов и инструментальных средств для разработки образовательных web – сервисов // 11-й Международный молодежный форум «Радиоэлектроника и молодежь в XXI веке». – Х.: ХНУРЭ, 2007. – С. 387.

5. Berners-Lee, T., Hendler, J., Lassilla O. The Semantic web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities // Scientific American, May 2001.

6. L. Stojanovic, J. Schneider, A. Maedche, S. Libischer, R. Studer, Th. Lumpp, A. Abecker, G. Breiter и J. Dinger. The role of ontologies in autonomic computing systems. – IBM Research Journal. 2004.

7. Xavier Lopez, Susie Stephens, Jeam Ihm, Jayant Sharma, Melliyal Annamalai, Omar Olonso. Semantic Data integration for the Enterprise. – March 2006.

8. Сайт ORACLE [Электрон. ресурс]. – Режим доступа: www.oracle.com/technology/tech/semantic_technologies.

9. Ternier S., Duval E., Vandepitte P. LOMster: Peer-to-peer Learning Object Metadata. In: P.Barker and S. Rebelsky (eds.) // Proc. of ED-MEDIA'2002 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, Denver, CO, June 24-29, 2002, AACE. – P. 1942-1943.

10. Хайрова А.А. Использование XML-ориентированных технологий для разработки и поддержки образовательного WEB-портала // XIII Межд. студ. НК «Высшая школа Украины перед вызовами XXI столетия» (8 апреля 2006г.). – Х.. НУА, 2006. – С. 134.

11. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб: Питер, 2000.

Поступила в редколлегию 10.05.2007

Рецензент: д-р техн. наук, проф. Е.И. Кучеренко, Харьковский национальный университет радиоэлектроники, Харьков.