МЕТОД СЕМАНТИЧЕСКОГО ДОСТУПА К ДАННЫМ НА ОСНОВЕ ОТОБРАЖЕНИЯ РЕЛЯЦИОННЫХ БД НА МОДЕЛЬ RDF

В.А. Вишняков, Д.С. Бородаенко

Ключевые слова: автоматизация управления обработки данных, частотный и лексический анализ, графовая структура данных, словарь идентификаторов URIref, типы данных, литералы, правила логического следования.

Введение

Для автоматизации управления обработки данных и знаний в Интернете с 1999 г. разрабатывается семантическое Вэб-пространство – это надстройка над существующей Всемирной паутиной, которая призвана сделать размещённую в ней информацию более понятной для компьютеров и интеллектуальных агентов [1]. Машинная обработка возможна в семантической паутине благодаря двум её важнейшим характеристикам [1]: использование унифицированных идентификаторов ресурсов (URI) и семантических сетей и онтологий. Современные методы автоматической обработки данных, доступных в Интернете, как правило, основаны на частотном и лексическом анализе *текстового* содержимого, которое предназначено для восприятия человеком. В семантическом Вэб-пространстве вместо этого используется стандарт RDF, описывающий семантические сети (графы), в которых узлы и дуги имеют URI.

RDF представляет собой способ описания данных в формате *субъект-отношение-объект*, в котором в качестве любого элемента этой тройки используются только идентификаторы ресурсов. Модель данных RDF опирается на следующие базовые понятия [2]: графовая структура данных, словарь идентификаторов URIref, типы данных, литералы, факты, правила логического следования. Однако большая часть накопленной информации, хранимой в Интернете, представлена в реляционных БД, доступ к которым семантическими средствами затруднен.

Структура метода

Для решения двух задач внедрения технологии RDF (интеграции с существующими реляционными БД и повышения производительности обработки данных) может быть использована система хранения RDF-данных, сочетающая подходы на основе отображения реляционных схем данных на модель RDF, на основе таблицы триплетов «субъект-предикат-объект». Для ее построения необходимо разработать модель адаптации реляционных данных для отображения на структуру RDF и обеспечить обработку RDF-запросов на доступ и обновление реляционных данных.

Модель адаптации реляционных данных представим в виде двойки: {M, N}, где M – отображения реляционной модели данных на модель RDF, позволяющего создавать утверждения RDF на основе значений полей записей реляционных таблиц, (реляционная таблица соответствует классу RDF-ресурсов, запись – RDF-ресурсу, значение первичного ключа – субъекту, имя поля – предикату, значение – объекту утверждения RDF); N – единое пространство имён (первичных ключей) для всех RDF-ресурсов, отображённых из записей реляционных таблиц, а также RDF-ресурсов, описываемых утверждениями, хранимыми в таблице триплетов, что позволит вносить в нее утверждения, использующие в позициях субъекта, предиката и объекта, любые RDF-ресурсы.

Обработку RDF-запросов определим тройкой: {Aq, An, P}, где: Aq – алгоритм преобразования запросов к данным RDF в запросы SQL; An – алгоритм обновления реляционных данных по запросу RDF; R – разбор и преобразование RDF-запросов и команд обновления данных в запросы и команды к реляционной СУБД на стандартном языке SQL.

По мере востребованности в конкретных приложениях от системы хранения RDF-данных также может потребоваться поддержка дополнительных возможностей. Набор ал-

горитмов, входящих в метод, обеспечивает поддержку следующих расширений:

– реификация (представление в виде самостоятельных ресурсов) утверждений RDF;

– применение правил логического вывода при преобразовании RDF-запросов для учёта в результатах выполнения запросов отношений подкласс-суперкласс, заданных предикатом *rdfs:sub Class Of* (вышеупомянутое создание единого пространства имён равносильно включению всех отображаемых классов ресурсов RDF на суперкласс; *rdfs.-Resource);*

– применение правил логического вывода для учёта подотношений, определённых при помощи предиката *rdfs: sub Property Of,* указывающего, что все утверждения, верные для подотношения, также верны и для базового отношения;

– применение правил логического вывода для учёта транзитивных отношений, входящих в класс предикатов *owl'TransitiveProperty* (примером практического применения транзитивного отношения может быть выборка всех комментариев к заданному сообщению вне зависимости от уровня вложенности).

На рисунке 1 представлена структура разработанного метода семантического доступа к данным на основе отображения реляционных БД на модель данных RDF.

Рисунок 1 – Структура метода семантического доступа к данным на основе отображения РБД на модель RDF

Суть метода заключается в интеграции новой модели адаптации реляционных данных для отображения на модель данных RDF; процедур логического вывода на основе известных алгоритмов; обработке RDF-запросов на доступ и обновление данных. Для реализации обработки разработаны новые алгоритмы преобразования запросов к данным RDF в запросы SQL и обновления реляционных данных по запросу RDF.

Следует отметить, что существующие системы хранения RDF-данных ограничиваются применением правил логического вывода на уровне приложения, что упрощает реализацию таких систем, но существенно снижает производительность обработки запросов. Например, выполнение запроса с учётом правил для подклассов и подотношений в подобной системе подразумевает перебор всех возможных комбинаций подклассов и подотношений, используемых в RDF-запросе, и выполнение отдельного запроса SQL для каждого варианта.

Разработанный метод семантического доступа, в отличие от аналогов, полагается на реализацию логического вывода на уровне хранимых процедур реляционной СУБД. Механизм хранимых процедур позволяет в процессе обновления данных создавать и поддерживать вспомогательные структуры

данных, обеспечивающие выборку данных с учётом всех заданных правил логического вывода посредством одного запроса SQL. Наиболее известный пример такой структуры – транзитивное замыкание, сводящее проверку истинности транзитивного отношения до одной операции выборки.

Модель адаптации реляционных данных не накладывает дополнительных ограничений на используемую схему реляционной базы данных сверх ограничений стандарта SQL. Любая таблица *T* в первой нормальной форме может быть отображена для доступа при помощи RDF-запросов. Таким образом, любая существующая база данных может быть адаптирована для доступа через RDF, не теряя при этом обратной совместимости с существующими SQL-запросами [3]. Процесс адаптации включает добавление в базу данных атрибутов, внешних ключей, таблиц и хранимых процедур, необходимых для преобразования запросов RDF и поддержки дополнительных возможностей, в частности, реификация утверждений и логический вывод на правилах для *rdfs:sub Class Of rdfs: sub Property Of u owhTransitiveProperty*. Разработанная модель может быть представлена в виде следующей последовательности шагов:

1. Ввести n множеств кортежей T_i, представляющих таблицы реляционной базы данных:

$$T_1 = \{\langle a_{t_1 1}, \ldots, a_{t_1 m_1}\rangle\}$$
$$T_2 = \{\langle a_{t_2 1}, \ldots, a_{t_2 m_2}\rangle\}$$
$$T_3 = \{\langle a_{t_n 1}, \ldots, a_{t_n m_n}\rangle\}$$

2. Для каждого реляционного атрибута выбрать соответствующее RDF-отношение $p_k \in P$, где P – множество отношений RDF, построить отображение отношений RDF на реляционные таблицы.

3. Создать единое пространство имён для ресурсов RDF, отображённых из записей таблиц и ресурсов, описываемых в таблице триплетов:

3.1. создать таблицу ресурсов *R,* отображённую на суперкласс *rdfs.'Resource,* с автоматически генерируемым первичным ключом *id(R),* так что для любого определённого на *rdfs: Resource* RDF-отношении *рщ:*

$$M\{p_{Rj}) = (R, a_{Rj});$$

3.2. заменить первичные ключи id(T_j) таблиц T_j, отображённых на подклассы класса *rdfs.'Resource,* на внешние ключи, ссылающиеся на таблицу ресурсов R.

3.3. обновить существующие внешние ключи с учётом замененных значений первичных ключей.

4. Зарегистрировать хранимые процедуры логического вывода по правилам *rdfs:sub Class Of* для обновления таблицы ресурсов и поддержки целостности внешних ключей при выполнении операций над таблицами подклассов T_j.

5. Создать хранимые процедуры и вспомогательные структуры данных, необходимые для поддержки дополнительных возможностей алгоритма преобразования запросов:

5.1. зарегистрировать хранимые процедуры для прочих случаев логического вывода по правилам *rdfs:sub Class Of;*

5.2. для представления RDF-данных, не отображённых на реляционную схему, и реификации утверждений RDF создать таблицу триплетов S, отображённую на *R*;

5.3. для поддержки логического вывода по правилам *rdfs: sub Property Of* добавить атрибуты a_s различения подотношений, ссылающиеся на записи в таблице ресурсов *R,* хранящие идентификаторы URIref соответствующих отношений, для каждого атрибута, отображённого на отношение, для которого определены подотношения;

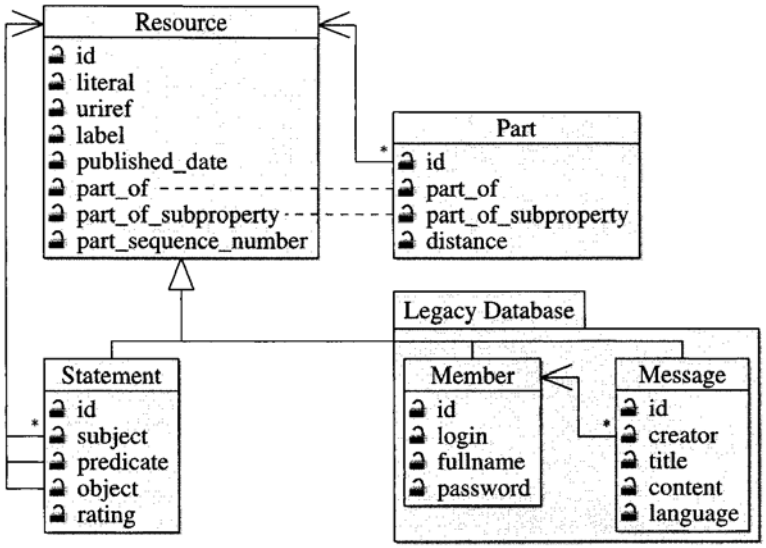
5.4. создать таблицы транзитивных замыканий $T_{a_t}^+$ и зарегистрировать хранимые процедуры логического вывода по правилам *owl'TransitiveProperty* для каждого атрибута $a_t(T_j)$, отображённого на транзитивное отношение p_t.

Пример БД

На рисунке 2 приведен пример схемы базы данных, полученной в результате применения всех вышеперечисленных изменений к схеме БД системы обмена сообщениями.

Таблицы Member и Message, отображённые на RDF таблицы исходной реляционной схемы, соответствующие сущностям «пользователь» и «сообщение». Таблица Resource представляет суперкласс *rdfs.'Resource,* её атрибуты literal, uriref и label используются в преобразовании запросов для различения типов ресурсов, а атрибуты publishecLdate и part_of отображаются на отношения *dcdate* и *dct:isPartOf*, действительные для всех ресурсов. В таблице Statement хранятся стандартные триплеты *(rdf.'subject, rdf'.predicate, rdf-.object),* расширенные специфичным для системы Samizdat атрибутом rating, содержащим определяемый пользователями рейтинг истинности реифицированных утверждений RDF. Таблица

Part содержит транзитивное замыкание отношения *dctiisPartOf.* Поскольку для данного отношения определена возможность различения подотношений через атрибут part_of _subproperty таблицы Resource**,** этот атрибут также отражён и в таблице Part; атрибут distance – стандартный атрибут, используемый при вычислении транзитивных замыканий.



Member, Message – отображённые на RDF таблицы *Ti,* Resource – таблица ресурсов; Statement – таблица триплетов; Part – таблица транзитивного замыкания для отношения *dct'AsPartOf* part_of_subproperty – атрибут различения подотношений отношения *dct:isPartOf*

Рисунок 2 – Схема модели адаптации реляционных данных для отображения на модель данных RDF на примере системы открытой публикации Samizdat

Разработанная система хранения RDF-данных полагается на ручное отображение отношений RDF на таблицы и атрибуты. Конфигурация отображения загружается из файла в формате YAML, пример такой конфигурации для схемы БД, приведенной на рисунке 2, представлен на рисунке 3.

```
ns:
s: 'http: //www.nongnu.org/samizdat/rdf/schema#'
rdf: 'http: //www.w3.org/1999/02/22-rdf-syntax-ns#'
dc: 'http: //purl.org/dc/elements/1.1/'
dct: 'http: // purl. org/dc/terms/'

map:
' dc::date': {Resource: published_date}
 'dct::isPartOf': {Resource: part_of}

 'rdf:: subject: {Statement: subject}
 'rdf:: predicate':{Statement:predicate}
 'rdf::object': {Statement: object}
 's::rating': {Statement: rating}

 's::login': {Member: login}
 's::fullName': {Member: full_name}

 'dc::creator': {Message: creator}
 'dc:-.title': {Message: title}
 's::content': {Message: content} 'dc::language': {Message: language}
```

Рисунок 3 – Элемент модели адаптации реляционных данных (отображение отношений RDF на реляционные таблицы и атрибуты)

В первой части конфигурации, выделенной ключевым словом **ns,** перечислены пространства имён, используемые далее для сокращённой записи идентификаторов URIref. В основной части конфигурации, выделенной ключевым словом **map,** дано отображение идентификаторов предикатов (сокращённых с помощью заданных выше пространств имён) на поля таблиц реляционной схемы БД. Например, заданный в последней строке примера предикат *dc.'language* отображён на поле **language** таблицы **Message.**

Заключение

В результате выполненной разработки получен новый метод для обработки знаний. Метод найдет применение для автоматизации обмена данными предприятия, вуза [4] с другими учреждениями и повышения эффективности поиска релевантной информации в Интернете.

ЛИТЕРАТУРА

1. Berners-Lee T. The Semantic Web / T. Berners-Lee, J. Hendler, O. Lassila // Scientific American. – May, 2001. – P. 28–37. – 240 p.

2. Klyne, G. Resource Description Framework (RDF): Concepts and Abstract Syntax [Electronic resource] / Klyne, G., Carroll, J.J. – W3C, December 2003. – Mode of access: http://www.w3.org/TR/rdf-concepts/. – Date of access: 04.07.2010.

3. Бородаенко, Д.С. Отображение реляционных данных на семантическую модель RDF при помощи динамического преобразования запросов / Д.С. Бородаенко // Доклады БГУИР. – 2010. – № 2(48). – С. 84–89.

4. Бородаенко, Д.С. Перспективы использования технологии RDF в сети вуза / Д.С. Бородаенко, В.А. Вишняков // Информатизация образования. – 2010. – № 4. – С. 35–42.

РЕЗЮМЕ

Для автоматизации управления обработки данных и знаний в Интернете средствами семантики языка RDF разработан новый метод отображения реляционных БД на модель данных RDF. Суть метода заключается в интеграции модели адаптации реляционных данных для отображения на модель данных RDF; процедур логического вывода на основе известных алгоритмов; обработке RDF-запросов на доступ и обновление данных. Метод найдет применение при автоматизации обмена данными предприятия, вуза с другими учреждениями и повышения эффективности поиска релевантной информации в Интернете.

SUMMARY

To automate the management of data and knowledge in Internet by means of RDF-semantics it was developed a new method of mapping relational databases to RDF data model. The method is to integrate adaptation model of relational data for display on a data model of RDF; logical inference based on well-known algorithms of RDF-processing requests for access and update data. Method will find application in the automation of data exchange of companies and universities with other agencies and in improvement of the search of relevant information in Internet.

*Статья поступила в редакцию 25 января 2011 г.