UDK 519.7:007.52

S.S. Shcherbak, V.V. Zavgorodniy

Kremenchuk Mykhailo Ostrohradskyi National University

PATTERN IMPLEMENTATION PRINCIPLES IN SEMANTIC WEB APPLICATION FOR MEDICAL INSTITUTION

Popularity of RDF data has recently increased. It assumes information description using RDF-triplets. Users' interaction with such data requires studying principles of working with triple stores. The complexity of dynamic querying is the main difficulty faced by contemporary scientists. A decision support system for medical institution has been created to deal with such a difficulty. It is a semantic web application that eases querying by pattern implementation. Visual queries are automatically converted into SPARQL query language for further accessing triple stores. Users obtain context with triplets according to constraints they set. A unified search interface has been developed for visual querying the triple stores.

Keywords: *decision support system, RDF triplet, query pattern, semantic web.*

Introduction

Global network is a huge data and knowledge storage, which provides means for accessing structured data through specialized search interfaces [1]. RDF data are defined using Resource Description Framework (RDF) [2]. This definition takes place in a form of triplets (subject – predicate – object) or quads (named graph – subject – predicate – object). For simplicity we shall use notation of triplets to define triplets as well as quads, if it does not lead to contradictions.

RDF data search is characterized by determination of objects that belong to some application domain. In case of using such an approach, the content of documents is presented as a collection of objects grouped together by certain context [3]. To determine object context we shall use a term "graph" that belongs to a quad. Information search assumes identification of named graph triplets and context identification that match user specific criterions or restrictions.

RDF model assumes distributed storage of objects along with their schemas (if such schemas exist) on different web-servers. Such servers include integrated or external triple stores. SPARQL (recursive acronym for SPARQL Protocol and RDF Query Language) is used for accessing triple stores. If we draw analogy with relational database, SPARQL is alike to SQL in a way [4]. To use such a language of querying effectively, one should possess some special knowledge and skills. It neither encourages SPARQL popularization, nor increases triple store expansion.

Our goal is to increase the efficiency of search based on triple stores by means of decrease in complexity of SPARQL dynamic querying and due to implementation of query patterns for further development of intellectual decision support system. Current approaches and possibilities of RDF data in the context of search and modern user interface development techniques for RDF data storages have been analysed. In the next section, search procedures are provided.

Pattern implementation for medical decision support system

Implementation patterns should be designed for efficient search implementation and friendly user interfaces [5]. The patterns should provide visual querying to named graphs of triple stores without requiring any knowledge or skills in SPARQL from users.

User's query on SPARQL is transmitted from web-application (with front end on HTML and CSS) to the triple store, where queries are processed using interfaces PDO and ODBC. Processing results are sent back to users in XML or JSON format. SPARQL queries are traditionally written by users. This requires some specialized knowledge about SPARQL and structure of objects residing in triple store. One of applications, that realize such a functionality, is ISQL web-interface from OpenLink used in Dbpedia. This application possesses universal data accessing techniques, and it is characterized by operation stability. A user gains access to named graphs of triple store through visual SPARQL query builder. It gives an opportunity to achieve information about graph topology automatically. This information includes predicates and object data types. It can be used to create search request through determination of user restrictions (filters) on data returned from a triple store.

The developed system assumes several actions to be performed on SPARQL for dynamic return of named graph topology. Such actions include: returning a list of named graphs from a triple store, returning a list of named graph predicates from a triple store, returning predicate data type. Fig. 1 shows query patterns for the actions mentioned above. These SPARQL queries may be more complicated, for instance queries that return some named graph or group, which name equals (or partially corresponds) to a predefined criterion (see fig. 2).

Language tags can also be used as user restrictions. In this case, a filter can be set to return objects from a graph in Russian (see fig. 3).

```
SELECT distinct ?g WHERE
{
GRAPH ?g { ?s ?p ?o}
}
```

```
SELECT distinct ?p
FROM <http://shcherbak.net/User>
{
?s ?p ?o
}
```

```
SELECT datatype(?o)
FROM <http://shcherbak.net/User>
WHERE
{
?s ns:date_of_ birthday ?o
}
```

Fig. 1. Query patterns for returning a list of all the named graphs, named graph predicates, predicate data type

```
SELECT distinct ?g  WHERE
{
GRAPH ?g
  { ?s ?p ?o . Filter(regex(?g, <http://shcherbak.net/>))
  }
}
```

Fig. 2. Query pattern for returning a graph which name corresponds to a predefined criterion

```
FILTER ( lang(?name) = "ru" && lang(?second_name) =
"ru" && lang(?third_name) = "ru"
        )
}
```

Fig. 3. Language filter pattern

Several restrictions can be used simultaneously to search for an object [4]. Fig. 4 shows query pattern for returning patient's data.

```
PREFIX ns:<http://shcherbak.net/>
SELECT  ?num ?name ?second_name
?third_name
WHERE
{ ?pat ns:name ?name;
     ns:second_name ?second_name;
     ns:third_name ?third_name;
     ns:belongs ?card.
```

Fig. 4. Query pattern for returning patient's data

These data are selected from the following two graphs (see fig. 5). The number of patient's medical card can be used as a parameter. (see fig. 6)

```
GRAPH <http://shcherbak.net/Patient>
    {
     ?numPat ?s1 ?o.
    }
GRAPH <http://shcherbak.net/Med_Card>
       {
        ?s2 ?p  ?num.
        FILTER (?s2=?o && ?p=ns:belongs || ?s2=?o &&
?p=ns:num && ?num=1  )
       }
```

Fig. 5. Graph pattern

```
FILTER (?card = (
            SELECT ?cardX
            FROM <http://shcherbak.net/Med_Card>
            WHERE
            {
              ?cardX ns:num ?num.
            }
          )
```

Fig. 6. Pattern for parameter setting

A unified search interface based on the above SPARQL queries is described in the next section, information control schema for medical institution with distributed data processing (that we have been dealing with) is also provided.

Experimental results

The proposed models and technologies are highly efficient in a sense of enterprise market appeal from the point of basic principles suggested in [6]. They are also interesting from the point of modern effective solutions for semantic applications. Fig. 7 shows information control schema for medical institution with distributed data processing that we have been dealing with.

The proposed example of user interface is interesting as ontology integration on the user interface layer is quite a novel field of research [7]. Fig. 8 shows search interface for the queries mentioned in the previous section.

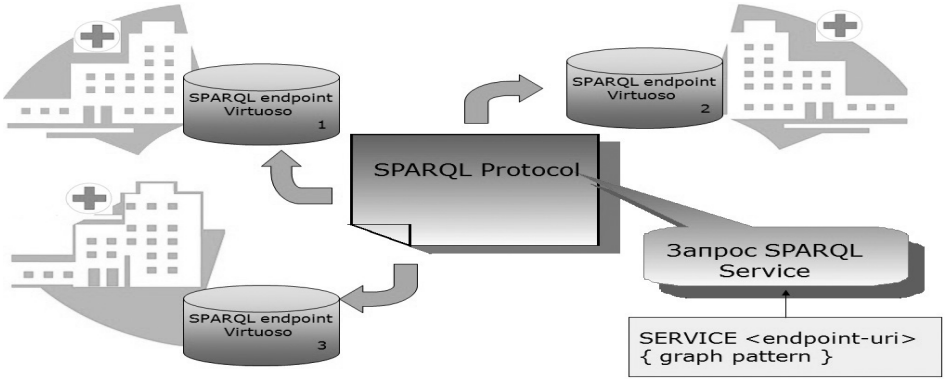

Fig. 7. Information control schema for medical institution

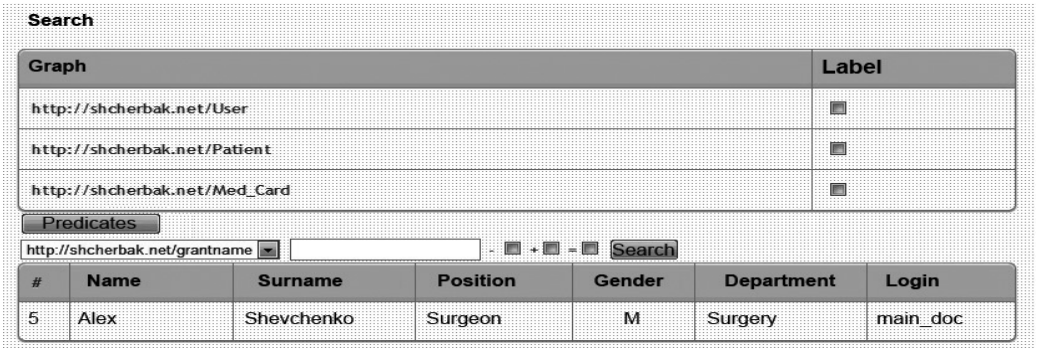

Fig. 8. Decision support system interface

The interface is performed on PHP and tested on multi-model data server OpenLink Virtuoso. This server is selected because of several reasons. First of all, it is cross-platform and can be used for relational data management as well as RDF and XML data management, free text content management and full text indexing. Secondly, it supports lots of programming languages and semantic web technologies, and thirdly, it is available for free and commercial use [8, 9]. OpenLink Virtuoso is chosen as a triple store, reasoner, RDF generator and SPARQL endpoint. The proposed search interface provides users with additional information that can ease query building and decrease time required for it. Query building procedure reduces to selection of named graphs (which are interesting for users) and setting restrictions on predicates. Search results are grouped according to the subject in a tabular style. Consequently, data are presented at a clients' front end in a friendly way, and users may know nothing about SPARQL query existence.

Conclusions

Although RDF language gives an opportunity to create graph structures of arbitrary level of complexity, there are several restrictions for communication with a triple store through ODBC (Open Database Connectivity) that have been revealed. Data in a triple store are structured according to object-oriented principles, i.e. named graph is a container (class) for a set of objects belonging to it. The objects are uniquely identified by a triplet subject. This restriction provides more possibilities for software developers that use relational databases. If we do not take these restrictions into account, the proposed method will work, but data generation sense will be changed. To consider object restrictions, named graphs based on typical SPARQL queries have been created.

Technological recommendations have been provided for implementation of user interfaces to triple stores with their automatic generation. After a query is executed, users obtain context with triplets that match to restrictions on predicates and objects. The proposed visual SPARQL querying implementation is oriented on fast input of queries to data stores and search quality perfection. It permits searching in conditions of partially defined object schemas. The proposed decision support system gives an opportunity to decrease time needed for information search.

Bibliography

1. Ma, Z. The Semantic Web for Knowledge and Data Management: Technologies and Practices / Z. Ma, H. Wang. – Hershey: IGI Global, 2009. – 367 p.

2. Bizer, C. The RDF Book Mashup: From Web APIs to a Web of Data / C. Bizer, R. Cyganiak, T. Gauß // 3rd Workshop on Scripting for the Semantic Web. – 2007. – Vol. 248.

3. Allemang, D. Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL / D. Allemang, J. Hendler. – Burlington: Morgan Kaufmann, 2009. – 352 p.

4. DuCharme, B. Learning SPARQL / B. DuCharme. – Sebastopol: O'ReillyMedia, 2011. – 258 p.

5. Beck, K. Implementation Patterns / K. Beck. – NJ: Addison-Wesley, 2008. – 157 p.

6. Wood, D. Linking Enterprise Data / D. Wood. – NY: Springer, 2010. – 291 p.

7. Paulheim, H. Ontology-Based Application Integration / H. Paulheim. – NY: Springer, 2011. – 270 p.

8. Segaran, T. Programming the Semantic Web / T. Segaran, C. Evans, J. Taylor. – Sebastopol: O'Reilly Media, 2009. – 302 p.

9. Hitzler, P. Foundations of Semantic Web Technologies / P. Hitzler, M. Krötzsch, S. Rudolph. – Boka Raton: Chapman and Hall/CRC, 2009. – 456 p.

Поступила в редколлегию 10.01.2013

Рецензент: д-р. техн. наук, проф. А.П. Оксанич, Кременчугский национальный университет им. М. Остроградского, Кременчуг.

ПРИНЦИПЫ РЕАЛИЗАЦИИ ШАБЛОНОВ В СЕМАНТИЧЕСКОМ ВЕБ-ПРИЛОЖЕНИИ ДЛЯ МЕДИЦИНСКОГО УЧРЕЖДЕНИЯ

С.С. Щербак, В.В. Завгородний

В последнее время наблюдается рост популярности использования RDF-данных. Их использование предполагает описание информации с помощью RDF-триплетов. Взаимодействие пользователей с такими данными требует от них изучения принципов работы с хранилищами триплетов. Сложность динамических запросов является основным препятствием, с которым сталкиваются современные ученые. Для преодоления этого препятствия, разработана система поддержки принятия решений для медицинского учреждения. Система представляет собой семантическое веб-приложение, которое облегчает формирование запросов путем реализации шаблонов. Визуальные запросы автоматически конвертируются в язык запросов SPARQL для дальнейшего доступа к хранилищам триплетов. Пользователи получают контекст с триплетами с учетом заданных ими ограничений. Для формирования визуальных запросов к хранилищу триплетов создан унифицированный поисковый интерфейс.

Ключевые слова: *система поддержки принятия решений, RDF-триплет, шаблон запроса, семантик-веб.*

ПРИНЦИПИ РЕАЛІЗАЦІЇ ШАБЛОНІВ У СЕМАНТИЧНОМУ ВЕБ-ДОДАТКУ ДЛЯ МЕДИЧНОГО ЗАКЛАДУ

С.С. Щербак, В.В. Завгородній

Останнім часом спостерігається зріст популярності використання RDF-даних. Використання цих даних передбачає опис інформації за допомогою RDF-триплетів. Взаємодія користувачів із такими даними потребує від них вивчення принципів роботи зі сховищами триплетів. Складність динамічних запитів є основною перешкодою, з якою зіштовхуються сучасні вчені. Для подолання цієї перешкоди розроблено систему підтримки прийняття рішень для медичного закладу. Система представляє собою семантичний веб-додаток, котрий полегшує формування запитів за допомогою реалізації шаблонів. Візуальні запити автоматично конвертуються до мови запитів SPARQL для подальшого доступу до сховищ триплетів. Користувачі отримують контекст з триплетами з урахуванням заданих ними обмежень. Для формування візуальних запитів до сховища триплетів створено уніфікований пошуковий інтерфейс.

Ключові слова: *система підтримки прийняття рішень, RDF-триплет, шаблон запиту, семантік-веб.*