Enterprise Business Intelligence Data Preparation Using RDF Data Sources

Wajee Teswanich and Suphamit Chittayasothorn

Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand {s9060012,suphamit}@kmitl.ac.th

Abstract. Business Intelligence (BI) plays important roles in executive decision making in organizations. Up-to-date information from good data sources always gives advantages to the organization. Nowadays, information in Semantic Webs is considered important source of BI data. It provides machine readable information using the Resource Description Framework (RDF). To perform BI activities on RDF documents, many research works propose direct mining on RDF documents. This paper presents a different approach for preparing BI project information from RDF data sources. A conceptual meta schema is used to describe RDF information. The meta schema is transformed into meta tables which are used to keep both RDF schema and document information. A transformation algorithm is proposed to transform the information into 5NF relational database schemas and relations which are finally transformed into BI project data structures.

Keywords: *Resource description framework, Conceptual meta schema, Relational database, Data warehousing,*

1. INTRODUCTION

Most organizations want to gain advantages in decision making and planning of the business. The success of business needs enough up-to-date information. Since Internet comes to be one of the most popular communication channels, information over the Internet is always a good source for the organizations. Semantic Web allows information on the Internet to be shared and understood by machines. Resource Description Framework (RDF) is the data format of Semantic Web presented over the Internet nowadays. RDF Schemas (RDFS) help RDF defined properties (attributes), kinds, and relationships of resources in RDF documents.

Business Intelligence (BI) supports executives' requirements of making the decisions and planning. BI tools from many vendors are available. The tools have drivers to connect many types of data source, except RDF/RDFS documents. So there are needs to make RDF/RDFS documents ready to be used for the tool. Most relational databases (RDBs) are supported by BI tools. Furthermore, RDB has database management system (DBMS) support in order to take care of query processing and other important data management facilities such as concurrency control and recovery control. Storing a RDF/RDFS document in a relational database allows the document to be the data source for BI tool and also manageable by the DBMS.

There are attempts to store RDF/RDFS documents in relational databases [1]. However, most of them only keep the RDF documents, not the schemas so many properties including the object-oriented properties such as subclass and subproperty are lost. Some of them purpose the design of relational database schema for only a particular application using prior knowledge about the data in order to optimize the relational tables [2]. In the case where RDFS is also kept, such as in Oracle10g [3] many complex non-relational tables are required together with new methods for table manipulations.

2. RDF/RDFS

Resource Description Framework (RDF) document is a computer and human understandable format of data widely used in Semantic Webs. RDF documents use XML syntax and define relationships among XML elements within RDF documents. An RDF Schema (RDFS) [4] defines types, constraints, and relationships of the resources in RDF documents. Ontologies using OWL Web ontology language [5] can be defined for the RDF documents in order to share vocabulary extension of RDF. This paper mainly defines the transformation from RDF/RDFS document to RDB. Ontologies are not referred to in this paper.

3. SYSTEM ARCHITECTURE

The system architecture comprises two parts: data source and BI platform. In data source, each RDF/RDFS document is mapped to meta tables, which are obtained from the conceptual meta schema of RDFS. The meta tables will generate relational database tables, which can directly be data source of multidimensional cube or data to be generated as data mart or data warehouse before loading into the cube. The BI server in the BI platform part will use data from the cube to generate reports, analyzed reports, planning, or forecasting.

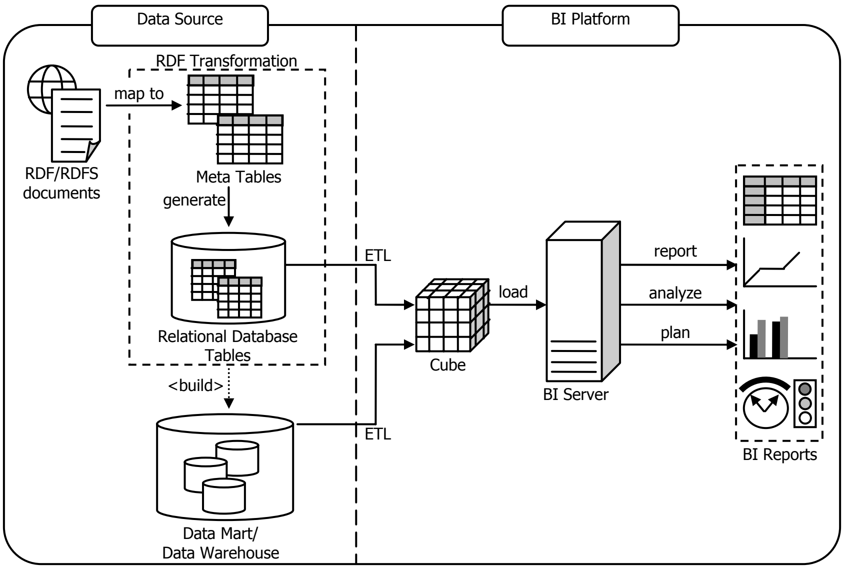


Figure 1. System Architecture

4. CONCEPTUAL META SCHEMA OF RDFS

From the data source in the system architecture in Figure 1, RDF transformation needs complete meta tables describing RDF/RDFS document before generates relational database tables. Since RDFS defines relationships among RDF elements, the corresponding complete conceptual meta schema of RDFS need to be defined.

The conceptual meta schema of RDFS can be described by a diagram of a semantic data model or a conceptual schema model e.g., an Entity-relationship (ER) diagram [6], a class diagram [7], a Nijssen's Information Analysis Method (NIAM) diagram [8,9], etc. The diagram should be able to present subclass relationships among classes and then be transformed to set of meta tables in RDB. In this research paper, the NIAM diagram [8,9] is used to explain RDFS and clarify RDF/RDFS graphs or documents.

4.1 NIAM Conceptual Meta Schema of RDFS

The RDFS specification [4] is described by the NIAM conceptual meta schema in Figure 2. The schema defines relationships among resources in RDF documents. From the schema, meta tables are generated as shown in Figure 3.

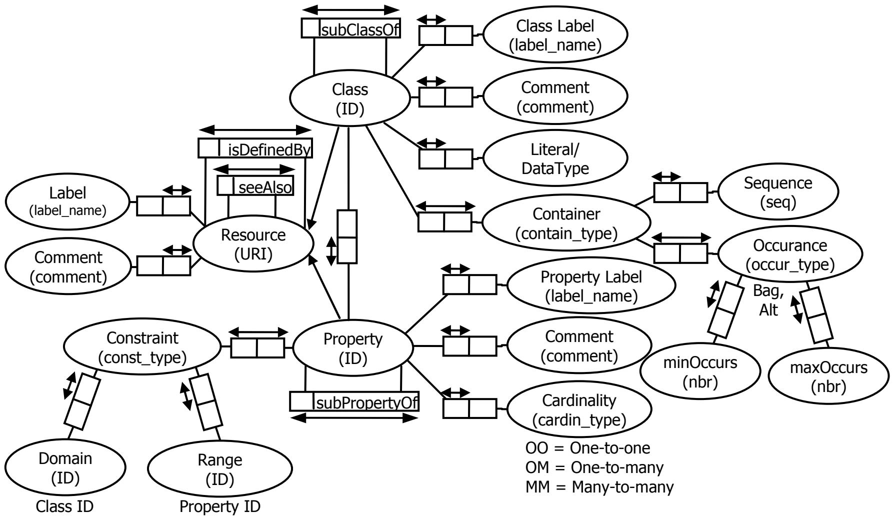


Figure 2. A Conceptual Meta Schema of RDFS

RESOURCE	URI	ResourceLabel	Comment		
SEE_ALSO	URI1	URI2			
IS_DEFINED_BY	URI1	URI2			
CLASS	CLASS_ID	ClassLabel	Comment	Literal	
PROPERTY	PROPERTY_ID	Property Label	Comment	Cardinality	CLASS_ID
CONTAINER	CONTAIN_TYPE	CLASS_ID	Seq	OCCUR_TYPE	
OCCURANCE	OCCUR_TYPE	CONTAIN_TYPE	CLASS_ID	nbr	
CONSTRAINT	CONST_TYPE	PROPERTY_ID	ID		
SUB_CLASS_OF	SUPER_ID	SUB_ID			

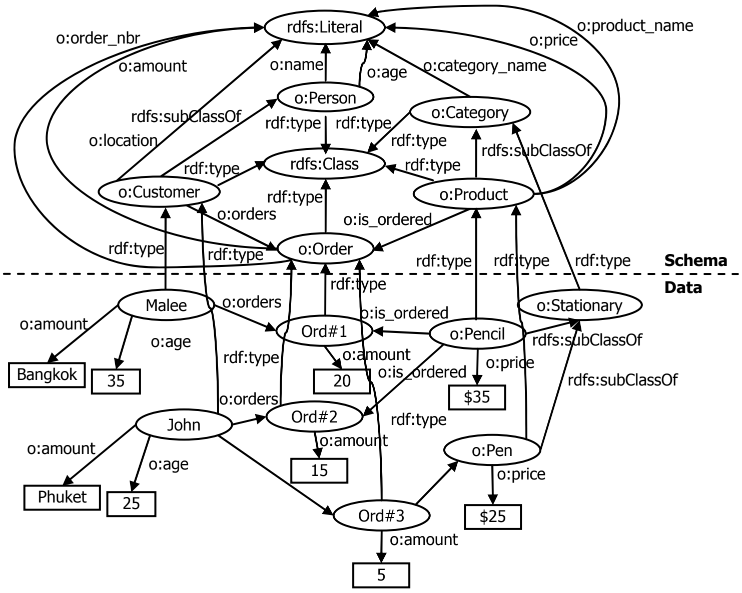
SUB_PROPERTY_OF	SUPER_ID	SUB_ID

Figure 3. Meta Tables of RDFS

4.2 Transformation of RDF/RDFS Document to RDB

In the RDF transformation in the system architecture in Figure 1, the RDF document and its schema are first mapped into meta tables in Figure 3. Then, the meta tables will derive relational database tables. However, there is some information that needs to be obtained from the RDF document such as the type of relationships between classes. If the ontology [5] is given along with RDF document, some information is also gathered from the ontology such as minCardinality, maxCardinality, allValueForm, etc.

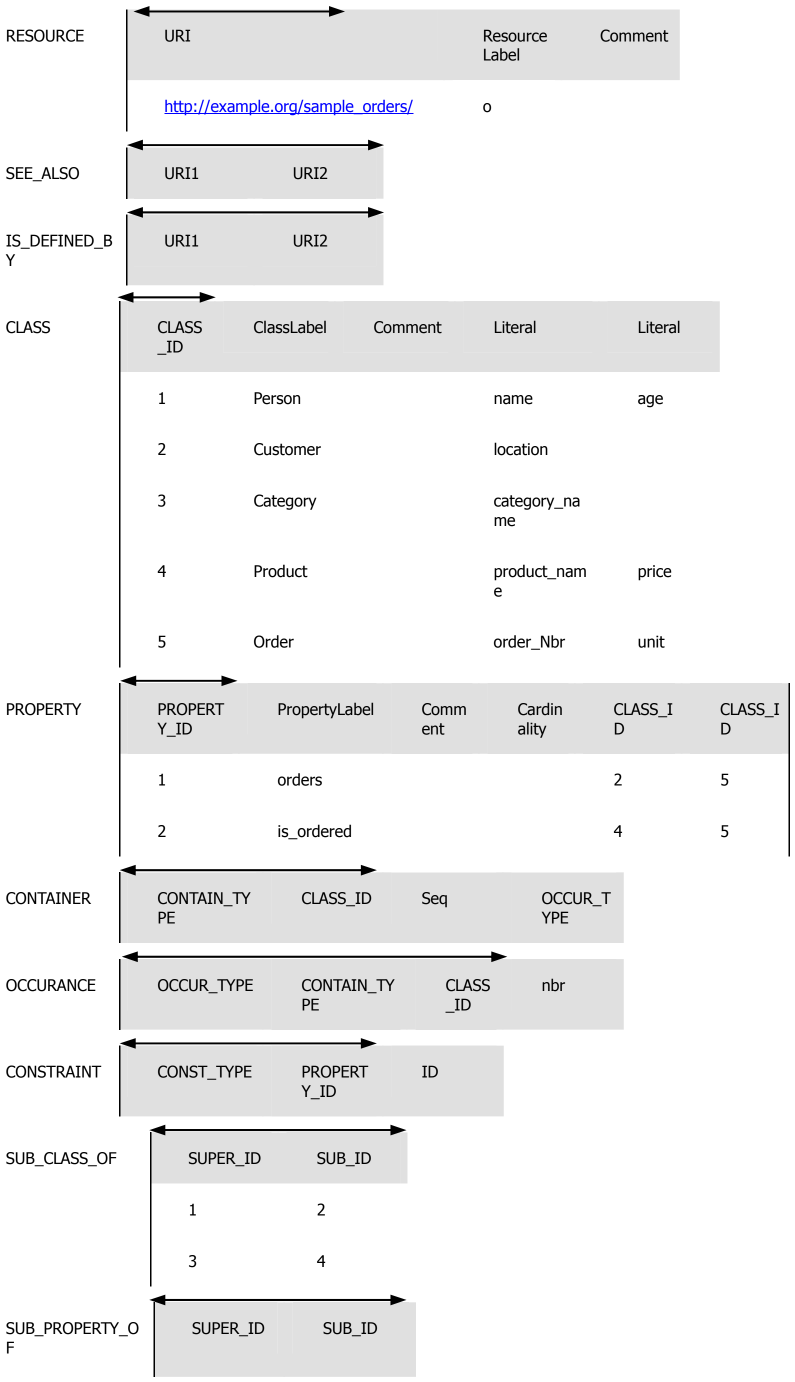
Figure 4-7 illustrate an example of RDF transformation process.



PREFIX o: http://example.org/sample_orders/

Figure 4. A Sample RDF/RDFS Graph

From the sample RDF/RDFS document in Figure 4, meta tables are populated as shown in Figure 5.



RESOURCE

URI	Resource Label	Comment
http://example.org/sample_orders/	o	

SEE_ALSO

URI1	URI2

IS_DEFINED_BY

URI1	URI2

CLASS

CLASS_ID	ClassLabel	Comment	Literal	Literal
1	Person		name	age
2	Customer		location	
3	Category		category_name	
4	Product		product_name	price
5	Order		order_Nbr	unit

PROPERTY

PROPERTY_ID	PropertyLabel	Comment	Cardinality	CLASS_ID	CLASS_ID
1	orders			2	5
2	is_ordered			4	5

CONTAINER

CONTAIN_TYPE	CLASS_ID	Seq	OCCUR_TYPE

OCCURANCE

OCCUR_TYPE	CONTAIN_TYPE	CLASS_ID	nbr

CONSTRAINT

CONST_TYPE	PROPERTY_ID	ID

SUB_CLASS_OF

SUPER_ID	SUB_ID
1	2
3	4

SUB_PROPERTY_OF

SUPER_ID	SUB_ID

Figure 5. Meta Tables Populated from the RDFS Example

Using the meta tables in Figure 5, the relational database tables in Figure 6 can be obtained. Database tables are populated from each record in class and property tables of meta tables.

Class Tables

PERSON

PERSON_ID	PERSON_NAME	AGE
1	Malee	35
2	John	25

CUSTOMER

CUSTOMER_ID	LOCATION
1	Bangkok
2	Phuket

CATEGORY

CATEGORY_ID	CATEGORY_NAME
1	Stationary

PRODUCT

PRODUCT_ID	PRODUCT_NAME	PRICE
1	Pencil	$35
2	Pen	$25

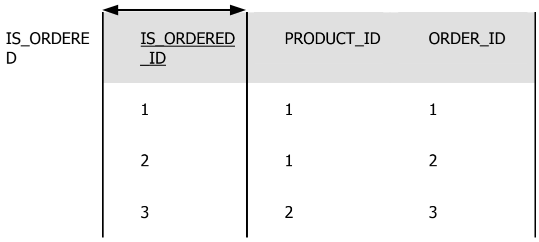
ORDER

ORDER_ID	ORDER_NBR	AMOUNT
1	Ord#1	20
2	Ord#2	15
3	Ord#3	5

Property Tables

ORDERS

ORDERS_ID	CUSTOMER_ID	ORDER_ID
1	1	1
2	2	2
3	2	3

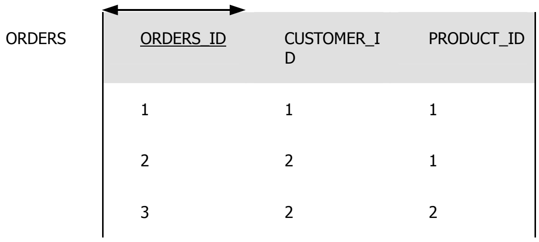


IS_ORDERED

IS_ORDERED_ID	PRODUCT_ID	ORDER_ID
1	1	1
2	1	2
3	2	3

Figure 6. Relational Database Tables

In case of two property tables contain a common class id as in Figure 6, the tables can be combined as shown in Figure 7.



ORDERS

ORDERS_ID	CUSTOMER_ID	PRODUCT_ID
1	1	1
2	2	1
3	2	2

Figure 7. Property Tables Redefined by Obtaining Relationships of Data in RDF Documents

4.3 RDF Data Sources

To use RDF as a data source of data warehouses or multidimensional cubes, a transformation to relational databases is required. At present, the processing of RDF/RDFS documents as databases is not efficient due to the lack of database management system (DBMS) support of RDFS as a database model. Query processing and optimization technologies and other important data management facilities such as concurrency control and recovery control which are commonly found in a Relational DBMS are not available in an RDF engine. Querying RDF/RDFS documents is based on tree traversal and simple pattern matching. From the productivity point of view, SQL requests made on relational databases are considered simpler and take less time to formulate than using the RDF-based language such as SPARQL [10]. Also, from the availability point of view, relational DBMS and supported Business Intelligence Software Tools which are widely available are mostly based on relational databases. A transformation from RDF and RDFS to relational databases so that they are in the easy-to-use and widely available form is therefore a logical approach of data management.

SPARQL is currently a working draft under development by W3C's RDF Data Access working group (DAWG). Since SPARQL is not a state-full protocol, using cursors with a special isolation and locking level on relational database could not be applied [11]. SPARQL does not support the data modification operations like INSERT, UPDATE, or DELETE in SQL. The query using SPARQL on RDF document still does not support aggregate functions like COUNT, MAX, MIN, or AVG in SQL in this current version (see Example 1 and 2). Finding the rank of the result cannot be applied using SPARQL (see Example 2). The calculation on returned value cannot be done using SPARQL (see Example 3).

The query processing on RDB makes process of building data warehouse or cube on RDF/RDFS document easier. SQL can be used in order to do the queries.

Example 1: Find customer(s) that send more than one order.

SQL

```
SELECT person.person_name
FROM orders
INNER JOIN [order] ON orders.order_id=[order].order_id
INNER JOIN person ON orders.customer_id=person.person_id
GROUP BY person.person_name
HAVING COUNT(*) > 1
```

Result:

person_name
John

Example 2: Find the product which has highest order amount

SQL

```
SELECT product.product_name, SUM ( [order].amount )
AS sum_amount
FROM orders
INNER JOIN [order] ON orders.order_id=[order].order_id
INNER JOIN product ON
orders.product_id=product.product_id
GROUP BY product.product_name
HAVING SUM([order].amount) =
 ( SELECT TOP 1 SUM(amount) AS amt
   FROM orders AS orders2,[order] AS order1
   WHERE orders2.order_id = order1.order_id
   GROUP BY orders2.product_id
   ORDER BY amt DESC )
```

Result:

person_name	sum_amount
Pencil	35

Example 3: Calculate the total price for each particular order

SQL

```
SELECT [order].order_nbr, ( [order].amount * product.price )
AS total_price
FROM orders
INNER JOIN [order] ON orders.order_id=[order].order_id
INNER JOIN product ON orders.product_id=product.product_id
```

Result:

order_nbr	total_price
Ord#1	$700
Ord#2	$525
Ord#3	$125

5. BUSINESS INTELLIGENCE

Business Intelligence (BI) technologies are employed in decision support systems (DSS) and executive-level support systems (ESS). Data warehouses whose data sources are mostly ETL (Extract, Transform and Load) from operation-level relational databases are used to keep and represent data in an easy-to-use format such as multidimensional cubes. Since RDF/RDFS documents are good source of information, they should be first accumulated for later ETL into the cube format. The accumulation and integration of RDF/RDFS information from various sources is an apparently complex task. We therefore propose that they be transformed into the relational database format before the integration and ETL activities.

The multidimensional cube is designed based on executives' requirements. From an application in section 4, the executives might want to see the orders based on product, location, and age of customer in order to make the suitable marketing on each product for the particular type of customers. The cube then can be designed as in Figure 8 and the example of executives' reports is given in Figure 9.

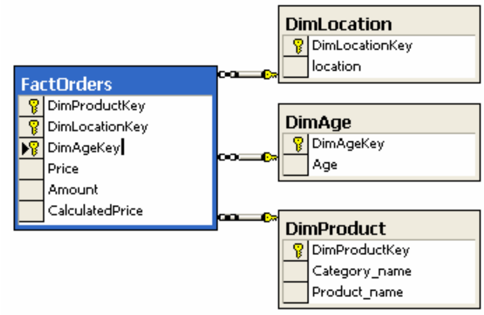
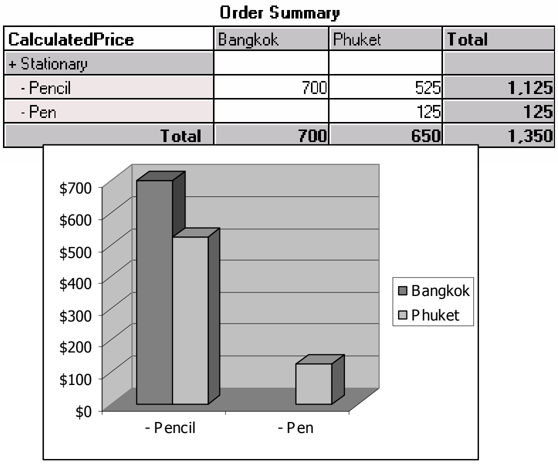


Figure 8. A Cube Design Example



Order Summary

CalculatedPrice	Bangkok	Phuket	Total
+ Stationary			
- Pencil	700	525	1,125
- Pen		125	125
Total	700	650	1,350

Figure 9. A BI Report Example

6. CONCLUSIONS

This paper presents a methodology for transforming RDF documents and schemas to relational database. Complete conceptual meta schema and meta tables of RDFS are introduced and the examples illustrate how RDF documents and their schemas are transformed and stored in relational tables. After the conversion from RDF data source to relational tables is done, multidimensional cubes and executive reports can be defined based on the organization needs.

REFERENCES

1. S.K. Kim, B.G. Kim, J. Lee, and H.C. Lim, The Strategies for storing RDF Documents using RDBMS, in *Proc. of The 6th International Conference on Advanced Communication Technology,* Volume 2 (2004), pp.1027–1029.
2. L. Ding, K. Wilkinson, C. Sayers, and H. Kuno, Application-Specific Schema Design for Storing Large RDF Datasets, in *Workshop on Practical and Scalable Semantic Systems* (Sanibel Island, Florida, October 2003).
3. Oracle, *Oracle® Spatial Resource Description Framework (RDF)*, 10g Release 2, (October 2005). http://download-east.oracle.com/docs/cd/B19306_01/appdev.102/b19307.pdf (Accessed May 10, 2007).
4. D. Brickley and R.V. Guha, *Resource Description Framework (RDF) Schema Specification 1.0,* W3C Candidate Recommendation (March 27, 2000). http://www.w3.org/TR/2000/CR-rdf-schema-20000327 (Accessed May 10, 2007).
5. D.L. McGuinness and F.V. Harmelen, *OWL Web Ontology Language Overview,* W3C Recommendation (February 10, 2004). http://www.w3.org/TR/2004/REC-owl-features-20040210/ (Accessed May 10, 2007).
6. C.J. Dates, *An Introduction to Database Systems*, Sixth Edition (Addison Wesley Publishing Company: 1995), pp.347-365.
7. D. Bell, *An introduction to structure diagrams in UML 2* (September 15, 2004). http://www128.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/ (Accessed May 10, 2007).
8. G.M. Nijssen and T.A. Halpin, *Conceptual Schema and Relational Database Design* (Prentice Hall of Australia Pty Ltd: 1989), pp.291-302.
9. T. Halpin, *Object-Role Modeling* (ORM/NIAM), Handbook on Architectures of Information Systems (1998).
10. E. Prud'hommeaux and A. Seaborne, *SPARQL Query Language for RDF,* W3C Working Draft (October 4, 2006). http://www.w3.org/TR/2006/WD-rdf-sparql-query-20061004 (Accessed May 10, 2007).
11. V. Bonstrom, A. Hinze, and H. Schweppe, Storing RDF as a Graph, in *Proceedings of the First Latin American Web Congress* (November 10-12, 2003), pp.27-36.