

## Improving response time of database systems by semantification of relational data

Mustafa YENİAD\*, Yakup KUTLU

Department of Computer Engineering and Natural Sciences, Faculty of Engineering, Yıldırım Beyazıt University,  
Ankara Turkey

Received: 13.01.2015

Accepted/Published Online: 07.08.2015

Final Version: 20.06.2016

**Abstract:** New generation web technologies present the new problem of intelligent representation of data in semantically structured systems. In this study, data hosted in a relational database system were converted to a new generation of web technologies (i.e. Web 3.0) with the semantic mapping method and semantification of the data was accomplished. The success of the transformation procedure was verified by a cross-comparison of the data between the relational database and the semantic data repository. By comparing the response time for performing SQL queries and equivalent SPARQL queries such as select, insert, update, and delete, operations hold in the semantic data repository was performed. The results show that the obtained speedup for the above operations on a semantic data repository offers a fetching performance that is at least four times better than that of the relational database system. This study may also be considered as an authentic solution to one of the obstacles in the implementation of the new generation of semantic web technology due to relational data that have not yet been semantified.

**Key words:** Semantic web, semantification, semantic mapping, SPARQL

### 1. Introduction

The idea of the semantic web (SW) is the redefinition of web-based resources to enable machines to access and easily understand the relevant material with the software agents. This idea of the SW was put forward by Berners-Lee et al. in 2001 [1]. They did not describe the SW as a different entity from the existing web, but rather as an extension of it. He also suggested that the first step of the SW is providing a mechanism for machines to process and ‘understand’ the content of web resources. In short, the SW may be regarded as an up-to-date version of web technology, a new upper-level (metadata) technology, and a social movement on behalf of open source technologies for a new generation of artificial intelligence [2]. This new technology starts from the formation of a new database format called resource description framework (RDF). This new framework is the biggest obstacle when using this technology. Since the existing technology hosts an enormous amount of data, it continues to constitute the biggest problem [3]. The semantification of the relational database is a research area, yet there is still no established standard for this action. Erling and Mikhailov used Virtuoso software’s ‘RDF views’ property to read the relational data as RDF triples without setting up an ontology; however, this situation constitutes a limitation for other popular relational databases [4]. Auer et al. used the Triplify tool to produce the RDF without using ontology from a relational database according to the SQL queries created manually; however, the absence of SPARQL support in Triplify creates limitations for the semantic validation process [5]. Bumans prepared SQL queries that generate RDF outputs in a simple ontology from data in the

\*Correspondence: myeniad@ybu.edu.tr

relational database and thus obtained RDF triples. However, the obligation to know the database schema continued due to not using any mapping language. In addition, confirmative SPARQL queries could not be executed because RDF triples were not transferred to a semantic data repository [6].

The aim of this study is to improve the response time of database systems by semantification of the relational data stored in a traditional database system. For this purpose, university student affairs automation software was used as a data resource. First the database schema pattern was determined according to the tables, fields, and the relations between them. Afterwards, an ontology was modeled that describes the university student affairs workflow. Then all the relational data were converted and exported as RDF triples format according to the mapping procedures applied between the relational database components and the ontology assets by semantic mapping method. Afterwards, the RDF data stack was imported into the semantic data repository and the RDF triples were verified with original database content. Finally, the traditional database system and semantic data repository were compared according to their response time to different types of equivalent SQL and SPARQL queries between them.

## 2. Semantic web components

### 2.1. Ontology

An ontology is the knowledge based on the communication between people or systems and represents the common sense recognized by the relevant system [7]. It is a philosophical study of the nature of being or reality that describes the basic categories of their relations. Ontology may be considered as the heart of SW technology. In this context, ontologies necessitate defining the properties of related objects and the relationships between them. Semantic description languages, such as RDF, RDFS, and web ontology language (OWL), are used in SW for the modeling and publication of ontologies as well as for performing an advanced level of web searches on documents with software agents.

Figure 1 illustrates the hierarchy of languages, which is known as Semantic Web Layer Cake, in the SW. The languages from the bottom of the stack up to OWL are currently standardized and accepted to create SW applications. OWL, which is based on the XML syntax, is used for publishing and sharing ontologies.

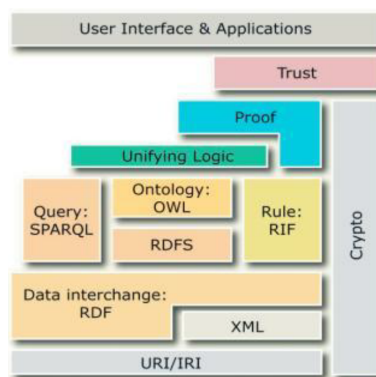


Figure 1. Semantic web layers ([http://en.wikipedia.org/wiki/Semantic\\_Web](http://en.wikipedia.org/wiki/Semantic_Web)).

### 2.2. RDF

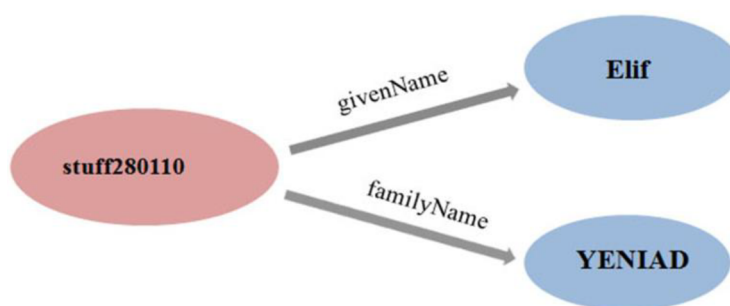
RDF is a data model for representing information for SW resources. In the SW, resources are expressed as triples. The core structure of the abstract RDF syntax is a set of triples, each consisting of a subject, a predicate, and an object. A set of such triples is called an RDF graph. An RDF graph consists of two nodes (subject

and object) and a triple connecting them (predicate). The abstract RDF syntax has two key data structures: RDF graphs are sets of subject–predicate–object triples, where the elements may be internationalized resource identifiers (IRIs), blank nodes, or data-typed literals. An RDF graph can be visualized as a node and directed diagram. For instance, ‘stuff280110’s givenName is Elif’ and ‘stuff280110’s familyName is YENİAD’ sentences can be represented as an RDF data set, as shown in Table 1.

**Table 1.** An example of an RDF data set.

Subject	Predicate	Object
stuff280110	givenName	Elif
stuff280110	familyName	YENİAD

Figure 2 visualizes the RDF data set in Table 1 as RDF graph instances.



**Figure 2.** RDF graph instances for the data set in Table 1.

Asserting an RDF triple says that a certain relationship, indicated by the predicates (givenName and familyName), holds between the resources denoted by the subject and object. In order to describe the relationship in a way that is understandable globally, the predicate was referenced with the term ‘namespace’. On its own, namespace does not have a well-defined meaning in the context of RDF, but is informally used to describe the relationship between the subject and object in the RDF data set. Namespace IRIs and prefixes are not a formal part of the RDF data model. They are merely a syntactic convenience for abbreviating IRIs. Studies for determining a general metadata standard still continue. Dublin Core Metadata, the Friend of a Friend (FOAF) project, and persistent uniform resource locators (PURLs) are examples of metadata projects. Furthermore, to date there is no established standard vocabulary for RDF triples. RDF triples can be presented in multiple notations, and RDF/XML, N-Triple, Notation 3, and Turtle are the most common RDF formats. XML declaration is located at the beginning of the RDF code. Then the RDF codes start RDF as the root element and definitions of namespace abbreviations. The sublabels of the subject are shown in Figure 3.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:j.0="http://xmlns.com/foaf/spec/#" >
<rdf:Description rdf:about="http://localhost/otomasyon.owl#stuff280110">
<j.0:givenName>Elif</j.0:givenName>
<j.0:familyName>YENİAD</j.0:familyName>
</rdf:Description>
</rdf:RDF>
```

**Figure 3.** Representation in RDF/XML notation of the data set in Table 1.

### 2.3. Triple store

A triple store is a semantic data repository for storing and retrieving RDF triples via a semantic query language known as SPARQL. Some triple stores have been built as a native RDF database engine, whereas others have been built on top of existing SQL-based relational database engines. Nevertheless, it seems likely that native triple stores will have an advantage in performance and SPARQL support. By achieving widespread use of the SW technology, semantic stores will be used indispensably on server systems in the future.

### 2.4. R<sub>2</sub>O mapping language

Relational to Ontology (R<sub>2</sub>O) is an extensible, fully declarative XML-based language that allows us to describe complex mapping expressions between relational database elements and ontology assets. Through the R<sub>2</sub>O mapping definition process, relational data from the database content can be extracted according to a modeled ontology.

### 2.5. SPARQL query language

SPARQL is a semantic query language for RDF repositories, used to retrieve and manipulate data stored in RDF format. It allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns. SPARQL was made a standard by the World Wide Web Consortium and is recognized as one of the key technologies of the semantic web. SPARQL 1.1 became an official W3C recommendation in March 2013.

## 3. Semantification process

The following steps were performed to semantify the relational database content according to modeled ontology:

- A student affairs database ER diagram was created according to tables, fields, and the relations shown in Figure 4.
- The classes that represent the automation data were identified in the ontology and assets were determined with properties.
- The ontology in Figure 5 was modeled according to the university's student affairs workflow via OWL.
- A mapping procedure was performed via R<sub>2</sub>O language between the ontology assets and the database fields.
- A relational database, which includes 9857 students, 30,993 courses, 1,188,956 exam grades, and 2099 staff data, was converted to RDF triples and an RDF stack file was obtained.
- The RDF data stack file was imported into the triple store system.
- Equivalent SQL and SPARQL queries, which were expected to return the same content from the database system and the triple store, were prepared.
- To query the content from both the database and the triple store, required programming scripts were coded.
- Data imported into the triple store were compared to the original database content through different types of SELECT queries. The results show that all the data in the triple store were the same as those in the database content. Therefore, semantic data were verified to relational data.

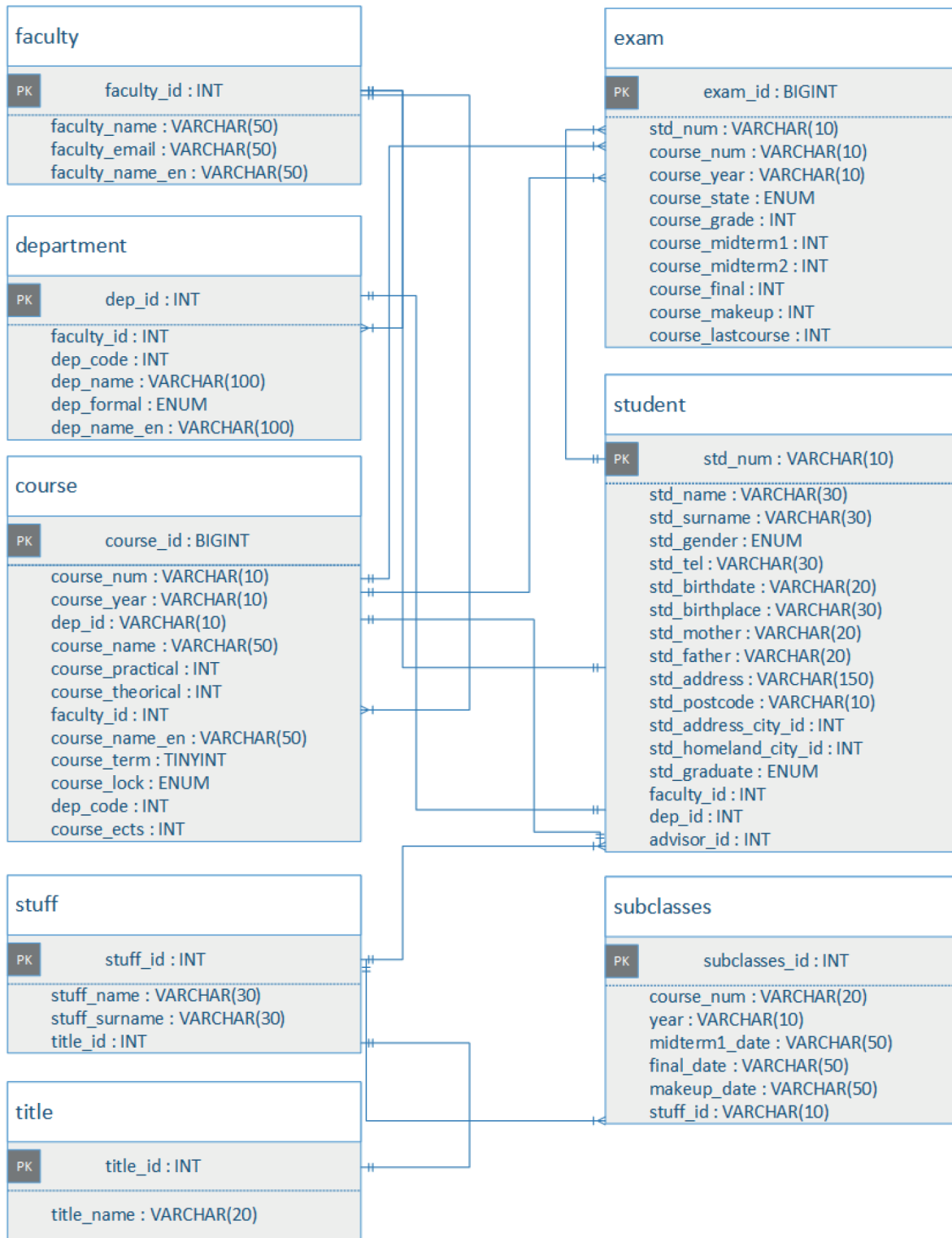


Figure 4. ER diagram of the student affairs database.

- Finally, the performances of the traditional database system and the semantic store were compared according to their response time to different types of queries between them.

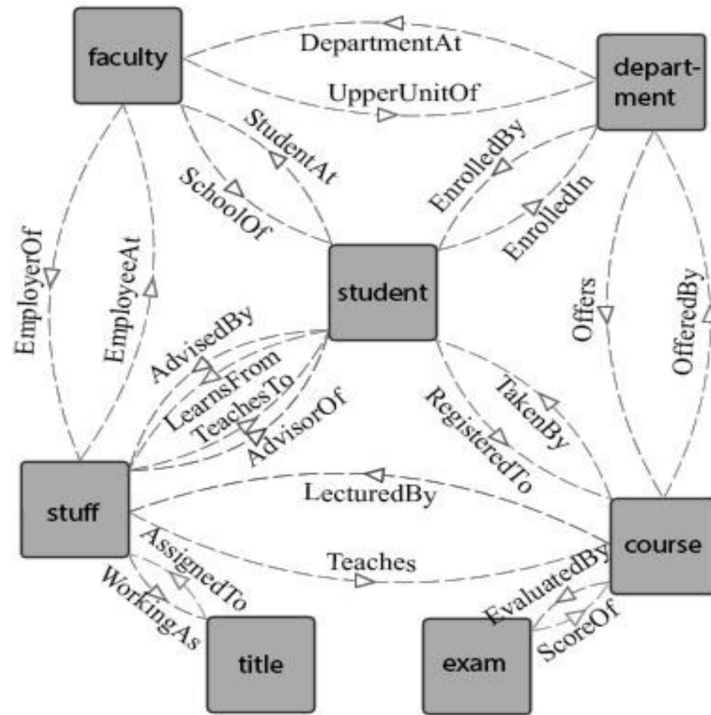


Figure 5. Ontology model of the student affairs workflow.

In this study, all the operations were performed through open source technologies such as the Linux operating system, PHP programming language, MySQL database system, Protégé editor, Eclipse platform with Neon-Toolkit, and Openlink Virtuoso (open source edition) as triple store.

The ontology model shown in Figure 5 has OWL codes, which follow in Figure 6.

```

<owl:ObjectPropertyrdf:ID="LecturedBy">
<rdfs:domainrdf:resource="#course"/>
<rdfs:rangerdf:resource="#stuff"/>
<owl:inverseOf>
<owl:ObjectPropertyrdf:ID="Teaches"/>
</owl:inverseOf>
</owl:ObjectProperty>
<owl:ObjectPropertyrdf:ID="EnrolledBy">
<rdfs:rangerdf:resource="#student"/>
<owl:inverseOf>
...
    
```

Figure 6. OWL codes of the ontology model (497 rows in total).

### 3.1. Semantic mapping process

Semantic mapping is a visual representation of the information about the relationships between concepts or words [8]. A mapping procedure can be performed to map heterogeneous databases or mutually different ontologies. In addition, a mapping procedure can be performed between ontology classes with databases field. Specially developed mapping methods and tools are also available for different types of requirements.

In this study, the mapping procedure was performed through the R<sub>2</sub>O language between the ontology assets and the fields in the database. A very short extract of the R<sub>2</sub>O codes is presented in Figure 7.

```

<r2o>
<dbschema-desc name="semantic">
<has-table name="stuff">
<keycol-desc name="stuff_id"/>
<nonkeycol-desc name="stuff_surname"/>
<nonkeycol-desc name="stuff_name"/>
<nonkeycol-desc name="title_id"/>
</has-table>
<attributemap-def name="http://localhost/otomasyon.owl#hasStuffID">
<selector>
<aftertransform><operation oper-id="constant">
<arg-restriction on-param="const-val">
<has-column>semantic.group.stuff_id</has-column>
</arg-restriction>
</operation>
</aftertransform>
</selector>
</attributemap-def>
...

```

**Figure 7.** R<sub>2</sub>O codes of semantic mapping process (1443 rows in total).

All relational data in the database were converted to RDF triples and an RDF stack file was obtained as shown in Figure 8.

### 3.2. Data verification and performance benchmarking

MySQL is multithreaded, multiuser, high performance, robust, and most preferably a relational database management system in today's web servers. Virtuoso software developed by OpenLink is a hybrid database management system. It may be used as a preferred native semantic data repository due to advanced XML and RDF data storage features and full SPARQL support. Both MySQL and Virtuoso can easily be used with many programming languages thanks to their ODBC API support.

In total, 20 pairs of SQL and SPARQL queries (groups of 5 queries of SELECT, INSERT, UPDATE, and DELETE) were prepared to compare the relational data and the semantic data. A pair of the equivalent SQL and SPARQL queries is shown in Table 2.

In order to improve the reliability of the results, each query was run three times and the average response time of the queries was calculated. The results are shown in Table 3.

```

<rdf:Description rdf:about="http://localhost/otomasyon.owl#course141845" >
  <j.0:hasName> DATA STRUCTURES</j.0:hasName>
  <rdf:type rdf:resource="http://localhost/otomasyon.owl#course" />
  <j.0:hasYear>2013</j.0:hasYear>
  <j.0:hasSchool_ID>15</j.0:hasSchool_ID>
  <j.0:hasTermNumber>3</j.0:hasTermNumber>
  <j.0:hasPracticeDuration>0</j.0:hasPracticeDuration>
  <j.0:hasTheoreticalDuration>4</j.0:hasTheoreticalDuration>
  <j.0:hasDepartmentID>1566</j.0:hasDepartmentID>
  <j.0:hasID>1523311</j.0:hasID>
</rdf:Description>
<rdf:Description rdf:about="http://localhost/otomasyon.owl#group156134" >
  <j.0:hasStuffID>30029</j.0:hasStuffID>
  <rdf:type rdf:resource="http://localhost/otomasyon.owl#group" />
  <j.0:hasID>156134</j.0:hasID>
  <j.0:hasMakeupExamDate>22.01.2013 17:00:00</j.0:hasMakeupExamDate>
  <j.0:hasMidtermExamDate>27.11.2012 08:30:00</j.0:hasMidtermExamDate>
  <j.0:hasCourseID>1523311</j.0:hasCourseID>
  <j.0:hasYear>2013</j.0:hasYear>
  <j.0:hasFinalExamDate>08.01.2013 08:30:00</j.0:hasFinalExamDate>
</rdf:Description>
<rdf:Description rdf:about="http://localhost/otomasyon.owl#faculty15" >
  <j.0:hasName>Engineering Faculty</j.0:hasName>
  <j.0:hasTypeID>1</j.0:hasTypeID>
  <j.0:hasID>15</j.0:hasID>
<rdf:type rdf:resource="http://localhost/otomasyon.owl#faculty" />
</rdf:Description>
...

```

**Figure 8.** An extract from the RDF data stack file (1,701,388 rows in total).

**Table 2.** A pair of the equivalent SQL and SPARQL queries (40 queries in total).

SQL	SPARQL
<pre> SELECT student.std_num, student.std_name, student.std_surname FROM student LEFT JOIN exam ON student.std_num = exam.std_num WHERE exam.course_num IN (   SELECT exam_id   FROM course   WHERE course_name="DATA STRUCTURES" ) ORDER BY student.std_num </pre>	<pre> PREFIX myont: &lt;http://localhost/otomasyon.owl#&gt; SELECT DISTINCT ?std_num ?std_name ?std_surname ?course_num FROM &lt;http://127.0.0.1:8890/DAV/home/dba/semantic.rdf&gt; WHERE {   ?lesson myont:hasName "DATA STRUCTURES" ;     myont:hasID ?course_num .   ?exam myont:hasStudentID ?std_num ;     myont:hasCourseID ?course_num .   ?student myont:hasID ? std_num;     myont:hasFirstName ?std_name ;     myont:hasLastName ?std_surname . } ORDER BY ?std_num </pre>



**Table 3.** Response time of queries (in ms) for the relational database system and the semantic data repository.

Query type	Average response time (ms)		Average (a/b)
	Relational database system (SQL) (a)	Semantic store system (SPARQL) (b)	
SELECT	55.3	4.2	13.2
INSERT	10.6	2.2	4.8
UPDATE	20.4	1.9	10.7
DELETE	10.6	1.6	6.6

#### 4. Conclusion

In this study, a total number of 40 equivalent queries in SQL and SPARQL were used to fetch the data from a traditional relational database management system and semantic data repository software. It was observed that all the SELECT queries returned exactly the same results. This shows that the semantification process proved to be successful according to the developed ontology.

In total, 20 pairs of SQL and SPARQL queries were also executed on both relational database and triple store systems. As Table 3 illustrates, the results of the comparison of the average duration of the queries showed that the semantic data repository software offers at least four times better fetching performance than the relational database system.

#### References

- [1] Berners-Lee T, Hendler J, Lassila O. The Semantic Web. *Sci Am* 2001; 5: 34-43.
- [2] Pollock JT. *Semantic Web for Dummies*. Hoboken, NJ, USA: Wiley, 2009.
- [3] Pandey G. The semantic web: an introduction and issues. *International Journal of Engineering Research and Applications* 2012; 2: 780-786.
- [4] Erling O, Mikhailov I. RDF support in the virtuoso DBMS. In: Pellegrini T, Auer S, Tochtermann K, Schaffert S, editors. *Networked Knowledge–Networked Media*. Berlin, Germany: Springer, 2009. pp. 7-24.
- [5] Auer S, Dietzold S, Lehmann J, Hellmann S, Aumuelle D. Triplify: light-weight linked data publication from relational databases. In: *ACM 2009 International Conference on World Wide Web*; 20–24 April 2009; Madrid, Spain. New York, NY, USA: ACM. pp. 621-630.
- [6] Bumans G. Mapping between relational databases and OWL ontologies: an example. *Comput Sci Inf Tech* 2010; 756: 99-117.
- [7] Ullrich, C. Description of an instructional ontology and its application in web services for education. In: *ISWC 2004 Workshop on Applications of Semantic Web Technologies for E-learning*; 7–11 November 2004; Hiroshima, Japan. Berlin, Germany: Springer, pp. 17-23.
- [8] Johnson DD, Pearson PD. *Teaching Reading Vocabulary*. 2nd ed. New York, NY, USA: Holt, Rinehart and Winston, 1984.