World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:3, 2009

New Methods for E-Commerce Databases Designing in Semantic Web Systems (Modern Systems)

Karim Heidari, Serajodin Katebi, and Ali Reza Mahdavi Far

***Abstract*—**The purpose of this paper is to study Database Models to use them efficiently in E-commerce websites. In this paper we are going to find a method which can save and retrieve information in E-commerce websites. Thus, semantic web applications can work with, and we are also going to study different technologies of E-commerce databases and we know that one of the most important deficits in semantic web is the shortage of semantic data, since most of the information is still stored in relational databases, we present an approach to map legacy data stored in relational databases into the Semantic Web using virtually any modern RDF query language, as long as it is closed within RDF. To achieve this goal we study XML structures for relational data bases of old websites and eventually we will come up one level over XML and look for a map from relational model (RDM) to RDF. Noting that a large number of semantic webs get advantage of relational model, opening the ways which can be converted to XML and RDF in modern systems (semantic web) is important.

***Keywords*—**E-Commerce, Semantic Web, Database, XML, RDF.

I. Introduction

A major paradigm shift is happening in computing .IT infrastructures moved from transactional architectures to client-server architectures in the 1980s and are now moving from client-server architectures toward navigational systems. This shift is reflected by fundamental changes in data structures:

• Transactional architectures are characterized by the hierarchical and the network (CODASYL) data models. These models started to evolve in the 1960s and are still in use today. Large amounts of operational data still reside in database systems such as IMS.[1]

• Client-server architectures were characterized by relational data models. Relational databases became the standard database technology in the 1980s and dominated the database market in the 1990s. The relational data model structures information in a way that allows different clients to interpret data items in various combinations.

• Navigational IT architectures require a new data model, which is still evolving. It seems that data models based on regular grammars are promising. At least, XML fits into this category.[1]

II. Semantic Web

Semantic refers to “meaning”. The semantic web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The Semantic Web is the abstract representation of data on the World Wide Web, based on the RDF standards and other standards to be defined. It is being developed by the World Wide Web Consortium (W3C), in collaboration with a large number of researchers and industrial partners.

III. Conceptual Modeling

Conceptual modeling is an early but important step in the design of information systems. While originally applied only to databases, conceptual modeling techniques are now applied to object oriented systems, too. In this section we will see how conceptual modeling can be utilized for XML-based architectures [1].

IV. The Entity Relationship Model

Developed by Peter Chen in the 1970s, entity relationship modeling (ERM) can be considered to be the ancestor of all modern modeling methods (Chen 1976). The acronym ERD may be more popular meaning entity relationship modeling, the graphical representation of an entity relationship model. In an ERM diagram, attributes are displayed as ellipsoids connected by straight lines to the owning entity sets, resulting in the typical ERM millipedes in this model Relationships may exist between entities. For example in below figure we have two entities (customer and product) that there is one relationship. Customer orders product (see below figure)[1]:

F. Karim Heidari is with the Islamic Azad University, Zarghan, Iran (phone: +98 9173151343; e-mail: karim.heydari@engineer.com) .
S. Katebi is with the Shiraz University, Iran (phone:+98 9173122254; e-mail:katebi@shirazu.ac.ir).
Ali Reza Mahdavi Far is with the Pasargad University, Shiraz, Iran (phone: +98 9177130906 ; email:mahdavi.a.r@gmail.com).

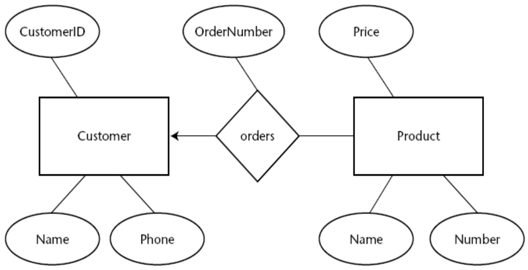

Fig. 1 Entity relationship modeling for customer and product

V. The Resource Description Framework

RDF can be seen as an enabling technology for semantic modeling, as a generic "assembler language" on top of which domain and task specific languages can be built. RDF applications include the Dublin Core and also DAML and OIL languages for the description of ontologies that we will discuss them. RDF provides an open standard for describing Web resources but not just Web resources. In fact, RDF allows statements to be made about anything, even about off line resources and the weather. As long as we can identify a resource with a URI (Universal Resource Identifier), we can use RDF to say something about this resource. And because we can assign a unique URI to almost anything, including our children, our car, and our Nintendo, RDF has a wide application range. RDF statements have a very simple structure. Each statement has the form of a triple, consisting of predicate, subject, and object. For example, in the sentence "John has phone number 415-555-6789", the subject is "John" because we are talking about him, "has phone number" is the predicate, and the object is the actual phone number, "415-555-6789". In RDF, all statements have this form: Subject has property. Each property consists of a name/value pair, with property values being string literals or references to other resources. RDF is an abstract, conceptual framework for defining and using metadata, independent of any concrete implementation and syntax. However, to write RDF statements we require a concrete means of expression. One possibility is directed labeled graphs. We can see in Fig.2 [1].

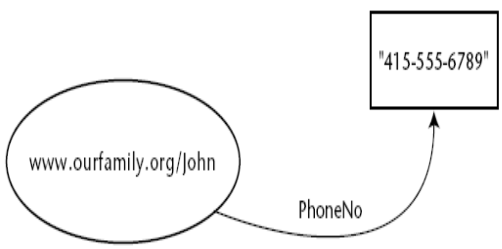

Fig. 2 Simple node and arc diagram. The resource (ellipsoid) has one literal property value (rectangle)

Another way to represent an RDF statement is the actual RDF syntax as defined in the RDF specification. (Currently, the RDF syntax is going through a revision cycle; see Beckett 2001.) This syntax is based on XML. Each RDF description is represented as an XML element. However, this does not mean that such a description can only describe XML resources:

```
<rdf:RDF>
<rdf:Description about="http://www.ourfamily.org/John">
<p:PhoneNo>
415-555-6789
</p:PhoneNo>
</rdf:Description>
</rdf:RDF>
```

VI. From ERM to RDF

Below figure shows how a previous ERM example is transformed into RDF. This example required the description of two resources, one for a Customer instance, and another for a Product instance. The relationship between Customer and Product is modeled through a property of the Customer instance. Note that in RDF we are talking about instances, while an entity relationship diagram is about types of entities and relationships.

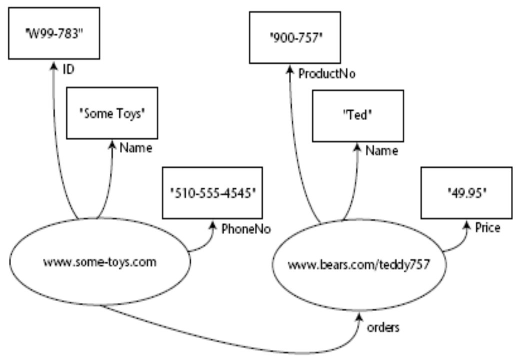

Fig. 3 Convert ERM to RDF

The following code shows the RDF serialization of the example in Figure 3. [1]

```
<rdf:RDF>
<rdf:Description about="http://www.some-toys.com">
<sales:ID> W99-783 </sales:ID>
<sales:Name> Some Toys </sales:Name>
<sales:PhoneNo> 510-555-4545</sales:PhoneNo>
<sales:orders
rdf:resource="http://www.bears.com/teddy757"/>
</rdf:Description>
<rdf:Description about="http://www.bears.com/teddy757">
<bears:ProductNo> 900-757 </bears:ID>
<bears:Name> Ted </bears:Name>
<bears:Price> 49.95 </bears:Price>
</rdf:Description>
</rdf:RDF>
```

VII. RDF Schema

RDFS or RDF Schema is a knowledge representation language, providing basic elements for the description of ontologies, otherwise called RDF vocabularies, intended to structure RDF resources. The data model of RDF schema allows creating classes of data. A class is defined as group of things with common characteristics. An object in the RDF schema is the instance of the class. The first version was released by W3C in 1998, and the final version was released in 2004. Classes and subclasses of RDF Schema are described below [2]:

rdfs:class : it is used to declare the resource of the class
rdfs:subClassOf : it is used to declare the attributes and hierarchies of the classes.
rdfs:domain of an property declares the class of the subject in a triple using this property as predicate.
rdfs:range of an RDF property that declares class or data type of the object in triple using this property as predicate.

VIII. Ontology Vocabulary

The main layer of semantic web architecture is Ontology vocabulary, which typically consists of hierarchical distribution of important concepts in a domain, along with descriptions of the properties of each concept. Ontologies play a pivotal role in the semantic web by providing a source of shared and precisely defined terms that can be used in metadata. The recognition of the key role in ontologies are likely to play in the future of the web that has led to extension of web mark up languages like XML Schema, RDF and RDF Schema. The recognition of the limitations in mark up languages led to the development of new web ontology languages such as OIL, DAML–ONT and DAML+OIL, OWL.

OWL (Web Ontology Language): OWL is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called ontology. OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web. OWL is a revision of the DAML+OIL web ontology language. OWL has been designed to meet the requirements of RDF, RDFS, XML Schema [3].

IX. Relational.owl

Relational.OWL is a Semantic Web-based representation format for relational data and schema components, which is particularly appropriate for exchanging items among remote database systems or to expose relational data on the Semantic Web. OWL, originally created for the Semantic Web enables us to represent not only the relational data itself, but also a part of its *interpretation*, i.e. knowledge about its format, its origin, its usage, or its original embedment in specific frameworks.

To describe the schema of a relational database with the techniques provided by RDF and OWL, we have defined reference OWL classes centrally, to which any document describing such a database can refer. The abstract representation of classes like *Table* or *Column* become a central part of the knowledge representation process realized within Relational.OWL. Additionally, we have specified possible relationships among these classes resulting in an ontology, a relational database can easily be described with. We call this central representation of abstract schema components and relationships the *Relational.OWL Ontology*.[4]

X. SPARQL query language for RDF

RDF is a directed, labeled graph data format for representing information in the Web. This specification defines the syntax and semantics of the SPARQL query language for RDF. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs.[5]

XI. Relational Databases and the Problems

Different database models have different aspects to them. The relational model is unarguably the most widely used database model today. Nevertheless, it is weak in capturing the semantics of a database. Thus, in the relational model, the semantics have to be separately described making it difficult to manage and use. As in relational modeling languages like SQL, database designers have to convert the real-world structures of the data to low level language constructs requiring an extra level of indirection. Likewise, database users who already have knowledge of the application domain will not be able to perform data manipulation until they have knowledge of the model constructs of the database languages. New applications are being developed everyday that demand data models that support complex relationships, rich constraints, and large-scale data handling. New fields such as bioinformatics and computer-aided design are evolving with great intensity. The amount of data stored in a database system is increasing astronomically as new applications of database emerge. We believe that the traditional database models -- e.g., relational model -- are not adequate to meet the new demands of database management but most of the information is still stored in relational databases [7].

XII. Mapping Process (relational database to RDF)

Mapping process consists of two main steps. First, the Relational.OWL representation of the schema and the data components of the original data source are generated. The schema representation becomes thereby an instance of the Relational.OWL ontology. In turn, the data items converted become instances of the schema ontology just created. This step could either be performed using the Relational.OWL application, i.e. the schema and data components are translated statically in a one-time process, or using a virtual representation of that RDF model, e.g. with RDQuery. The advantage of the latter is obvious, since the data stock, on which the queries are performed, is always up-to-date. This cannot be guaranteed using the Relational.

OWL application. Nevertheless, if the source database does not change frequently, a static translation into the Relational.OWL representation could be enough. Having created the Relational.OWL representation of the relational database, the second step including the actual mapping can be performed. The RDF model just created may now be queried with an arbitrary RDF query language. As long as the query language is closed, the resulting query response is again within the Semantic Web, i.e. it is a valid RDF model or graph and may then be processed by other Semantic Web applications using their own built-in functionality for reasoning tasks [6].

XIII. Implementation

Relataional.OWL, the software package presented here, connects to a relational database using a genuine JDBC-Connection and mediates between the relational and the semantic worlds. On the one hand, it converts a database schema automatically into a suitable RDF/OWL ontology and represents the corresponding data items as its instances. On the other hand, it processes schema and data representations and imports them into a suitable database.

Required Packages:

The following external JAR-Packages, or equivalent, are required for using Relational.OWL:

- commons-logging.jar
- concurrent.jar
- icu4j.jar
- jakarta-oro-2.0.5.jar
- jdom.jar
- jena.jar
- xercesImpl.jar
- JDBC-Driver, e.g.: mysql-connector-java-3.1.10-bin.jar

Relational.OWL is written in Java, uses JDBC for the database and the JENA framework for the Semantic Web connectivity. Since the RDF/OWL representation of the database is vendor-independent, a data and schema extract of a database from vendor A can easily be imported into a database from another vendor. Relational.OWL currently supports MySQL and DB2 databases, but a corresponding implementations for additional vendors may easily be added.[4]

XIV. Conclusion and Future Work

In this paper we have described ERM, RDF, RDF SCHEMA, Ontology, OWL.RELATIONAL, SPARQL and we have described how to map data from relational databases into a real RDF representation using a Semantic Web query language. To use such query languages for a mapping purpose, three main requirements have to be met. The relational database (i.e. its schema and data components) has to be described using the Relational.OWL ontology. This automatic semantic representation of the relational database can then be queried using any RDF query language.

A lot of work is being done in all layers of semantic web but not being much done in Proof and Logic. This paper gives an idea of semantic web and an analysis of various layers of semantic web architecture which may assist to meet the various challenges of layers of semantic web architecture.

References

[1] Berthold Daum, Udo Merten, “system Architecture with XML” , Morgan Kaufmann publishers,2003
[2] http://www.w3.org/TR/rdf-schema/ ,2009
[3] http://www.w3.org/TR/owl-features/,2009.
[4] http://dbs.cs.uni-duesseldorf.de/RDF/, 2009.
[5] http://www.w3.org/TR/rdf-sparql-query/,2009.
[6] Yuan An, Alexander Borgida, and John Mylopoulos. Inferring Complex Semantic Mappings between Relational Tables and Ontologies from Simple Correspondences. In CoopIS, DOA, and ODBASE, OTM Confederated International Conferences, Cyprus, Part II, volume 3761 of LNCS, pages 1152–1169. Springer, 2005.
[7] Saurabh Boyed, A Semantic Database Management System(SIM),Computer science Honors THESIS ,CS 379H , Spring 2003.

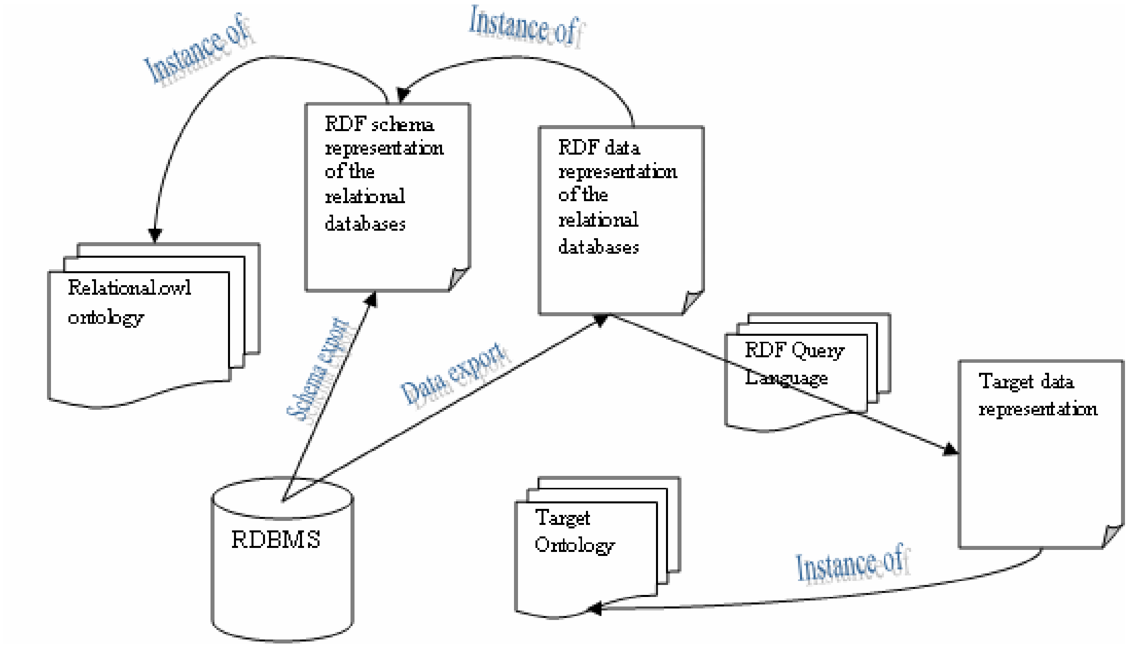

Fig. 4 Mapping Process