RGORDB: Rules for Generating Ontology from Relational Database

Rajeshkumar.T[1], John Paulraj.C[2], Mohan Prasanth.R[3], Nagarasan.M[4]

PG Scholar, Department of Computer Science and Engineering[1,2,3,4]

Adithya Institute of Technology, Coimbatore[1,2,3,4]

rajeshlbe@gmail.com[1], *johnpaulrajj@gmail.com*[2], *mohanprasanthcs@gmail.com*[3], *nagarasancs@gmail.com*[4]

Abstract-In recent years the growth of internet applications has highlighted the limit of traditional data model representation like UML. Ontologies are means of knowledge sharing and reuse and promise a more suitable knowledge representation for building more "intelligent" applications. On the purpose of generating ontologies from data sources on the practical application consideration, some rules on mapping relational database to ontologies are proposed. Firstly, map the relation schema information into the ontology as concepts. Secondly, achieve the attributes, and map them to the properties in the ontology. Thirdly, analyze the constraints in databases, and modify the property constraints as needed in ontology. Then, extract the records into the ontology as individuals of concepts. At last, an implementation of generating ontologies from MySQL data source using the rules above is recommended as a case study of these rules. The result shows the rules are reasonable and complete in generating ontologies.

Index Terms-Semantic Web; Ontology; Relational Database; OWL; SPARQL; Protégé.

1. INTRODUCTION

Nowadays, frequently it is a preferred way to get data from a relational database whose structure is defined by a relational database schema. At present, with the high speed of business data blowing up and economy developing, it is not rare that many systems run at the same time in a company, even in a department. On the other hand, the data should be unified in an industry, like manufacturing, medical treatment, financing, banking and so on. Because of the difference between the companies, the data schema structure cannot be shared. It could result the repeat work in developing information systems for these companies.

The Web Ontology Language (OWL) is aW3C standard for modeling ontologies in the Semantic Web. Ontologies have been established for knowledge sharing and are widely used as a means for conceptually structuring domains of interest. Ontologies serve as metadata schemas, providing a controlled vocabulary of concepts, each with explicitly defined and machine-processable semantics. Besides, the ontology technology has its own reasoning mechanism. We can define the concepts in the ontology, and decide those individuals that have different attribute values whether are the same objects. Ontologies conveythe knowledge of a certain field perfectly, and define many axioms among these concepts.

The Data Management is a way to unify, manage and integrate references data across the Information System of the company. These data can be of several kinds (Products, services, offers, prices, customers, providers, lawful data, financial data, organizations, structures, persons, etc.)[1]. This papers proposes the rules to generate the ontologies from the existing relational databases efficiently. The information can be extracted from the generated Ontologies, using the SPARQL query language.

This paper organized as follows, Section 2 explains the Relational database and Ontology schemas and types of mapping process. Section 3 explains the rules to generate the Ontology from Relational database. Section 4 explains the some of the works related to the Ontology generation. Finally conclusion is given in Section 5.

2. RELATIONAL DATABASE AND ONTOLOGY

This paper focuses on the conversion from relational databases to partial ontologies. Partial ontologies store limited information, and it cannotconvey all the information in the field they describe. The relational databases we discuss here should meet the demand of third normal form (3NF). A database that adhere to the third normal form means that it eliminates the transitive dependency from any candidate key of a relation R to a non-prime attribute in

R. In case of the conditions above, we can guarantee the databases reasonable and structural, and complete on semantic aspects. However, data in databases is just listed as pieces of records. They have no inner relations with each other.

In master data management, the data are divided into two parts: master data and transactional data.

• Master data represents the core objects in business behaviors, including customers, products and so on. Once it is stored in databases, we need to maintain the time-effectiveness and accuracy. It also includes some association data, which explains the relations between core objects.

• Transactional data represents the transactional flow records in business behaviors. It describes the business actions in the period of one transaction.

According to the data classification above, we can get a brief description of business data. The transformation process in our work is shown as figure 1. It contains two steps.

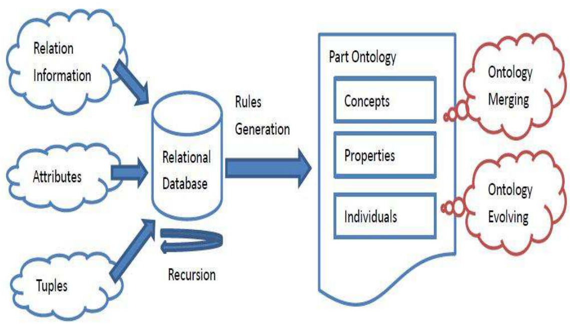

Fig 1. Framework of Ontology Transformation

• Schema transformation, which extracts the schema information of the databases, and convert it to the concepts and properties of the new ontology.

• Data transformation, which extracts the records stored in databases and fulfill the instances in the new ontology.

2.1. *Relational Database (R)*

In relational database, a relation is defined as aset of tuples that have the same attributes. In order to describe the database in a clear and ontology-like way in semantic aspect, we define a relation in database structure as a quadruple tuple just as follows: R = <RS, RA, RC, RT>

- RS represents the relation schema information, for instance, relation name, relation type including table, view and other types.
- RA represents the relation body schema information, including attribute name, attribute type, and attribute default value.
- RC represents the constraints information of the relation, including primary key, foreign key, unique, column null or not, both primary key and foreign on one attribute.
- RT represents the tuples that this relation holds. One tuple represents the information of one object.

2.2. *Definition 2 Ontology (O)*

The structure of ontology is defined as a quintuple tuple just as follows:O = <C, A, I, CD, AD>

- C represents a set of concepts in this ontology.
- A represents a set of properties in this ontology.
- I represents a set of instances in this ontology.
- CD represents a set of concept definitions in this ontology. It defines the equivalent concept and super concept.
- AD represents a set of property definitions in this ontology. It defines the domain value type and range value type.

2.3. *Mapping Process:*

Mapping is a critical operation in many application domains, such as semantic web, schema or ontology integration, data integration, data warehouses, e-commerce, etc. We can distinguish three types of mapping: 1) schema mapping, 2) ontology mapping, and 3) database-to-ontology mapping, on which we focus in this paper.

(1). Schema Mapping: Mappings are established between the schema of the individual databases. This process takes two schemasas input and produces a mapping between elements of the two schemas that correspond to each other. Some interesting works in this area are the works of Fuxman et al.[5] and Miller et al. [7]. We refer also to [8] as a survey on existing approaches.

(2). Ontology Mapping: The main purpose of this process is to relate the vocabulary of two ontologies that share the same domain of discourse. Ontology mapping is somewhat similar to database schema matching, but it has many particularities due to the structural and conceptual differences between ontologies and databases. Kalfoglou et al. gives in [6] an excellent survey on ontology mapping.

(3). Database-to-Ontology Mapping: This is the process whereby a database and an ontology are semantically related at a conceptual level, i.e. correspondences are established between the database components and the ontology components.

3. RULES FOR GENERATING ONTOLOGY

RGORDB maps constructs of a relational database to Ontology, using the names of constructs of the relational database as the names of constructs of the ontology. A prerequisite for this mapping is the mapping of constructs of a relational model to an ontological model. This mapping is defined by a set of rules for:

- Mapping tables
- Mapping columns
- Mapping data types
- Mapping constraints
- Mapping rows.

3.1. *RDB and Ontology*

Table 1: Relational Database and Ontology

Relational Database	**Ontology**
Table	Class
Column	Property
Data	Instances
Primary Key	Functional Property
Foreign Key	Object Property
Column Constraints	Property Restrictions

3.2. *Rules*

For all the tables of banking system, classes are defined in OWL DL. Tables are mapped to classes based on the rule.

Rule 1: Create a class in Ontology for the RDB table

Create table account (ac_no integer primary key, b_name char(15) references branch (b_name), balance float check (balance > 1000)

Rule 2: Create Ontology Data type Property for each RDB attribute and its data type.

From the Table structure get the column names and data types.

Ontology: <owl:DatatypePropertyrdf:ID = “ac_no”>
<rdfs:domainrdf:resource = “account”/>
<rdfs:rangerdf:resource =“&xsd;positiveInteger”/>
</owl:DatatypeProperty>
<owl:DatatypePropertyrdf:ID = “#b_name”>
<rdfs:domainrdf:resource = “account”/>
<rdfs:rangerdf:resource = “&xsd;string”/>
</owl:DatatypeProperty>
<owl:DatatypePropertyrdf:ID = “#balance”>
<rdfs:domainrdf:resource = “account”/>
<rdfs:rangerdf:resource = “&xsd;float”/>
</owl:DatatypeProperty>

Rules 3:Map each row of data with Ontology instances of classes as declared in RDF syntax

Insert into account (ac_no, brance_name, balance) values (36883, Saravanampatty, 200000)

Ontology : <account>
<ac_no rdf:datatype=”&xsd:integer”>36988</ac_no>
<b_name df:datatype=”&xsd:string”>SVPatty</b_name>
<balance rdf:datatype =”&xsd:double”>20000 </balance>
</account>

Rule 4: Create Inverse Functional Property and set minimum Cardinality value as “1” for an RDB column for which two different rows cannot have the same value and for setting Not Null constraint respectively.

For example, one account number is assigned to one person only. A column constraint primary key is mapped to both an inverse functionalproperty and a minimum cardinality of 1. Primary key ac_no (account table structure)

Ontology: <owl:InverseFunctionalPropertyrdf:ID = “ac_no”/>
<owl:classrdf:ID = “account”>
<rdfs:subClassOf>
<owl:restriction>
<owl:OnPropertyrdf:resource = “#ac_no”/>
<owl:minCardinalityrdf:datatype = “&xsd: nonNegativeInteger”1/>
</owl:restriction>
</rdfs:subClassOf>

```
</owl:class>
```

Rule 5: Create Ontology Object Property with domain and range constraints to set a foreign key, if foreign key is not a primary key or part of a primary key

Branch name of account table is the foreign key for branch name of branch table and it is not a primary key or part of primary key.

```
Ontology :<owl:ObjectPropertyrdf:ID =
“b_name”>
<rdfs:domainrdf:resource = “#account”/>
<rdfs:rangerdf:resource = “#branch”/>
</owl:ObjectProperty>
```

Rule 6: If foreign key is a primary key or part of a primary key in RDB, create Object Property with domain and range constraints accompanied by a cardinality of 1

Create table borrower(c_name char(15) references customer (c_name), l_num float primary key(c_name, l_num))

```
Ontology: <owl:ObjectPropertyrdf:ID =
        “c_name”>
<rdfs:domainrdf:resource = “#borrower”/>
<rdfs:rangerdf:resource = “#customer”/>
</owl:ObjectProperty>
<owl:Classrdf_ID=”#borrower”>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onPropertyrdf:resourse=”#c_name”/>
<owl:cardinality
rdf:datatype=”&xsd:nonNegativeInteger”1/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

Rule 7: Create Ontology property restriction for Cardinality to map constraints at column levelie., for the restrictions on how many distinct values a property may or must take

```
Ontology: <owl:Classrdf:about="#account">
<rdfs:subClassOf>
<owl:Restriction>
<owl:onPropertyrdf:resource="#balance"/>
<owl:minCardinality
rdf:datatype="&xsd;nonNegativeInteger">1000
</owl:minCardinality>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

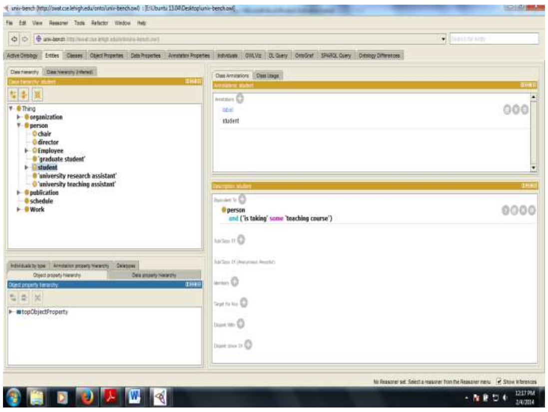

Fig 2: Ontology Class view in Protégé

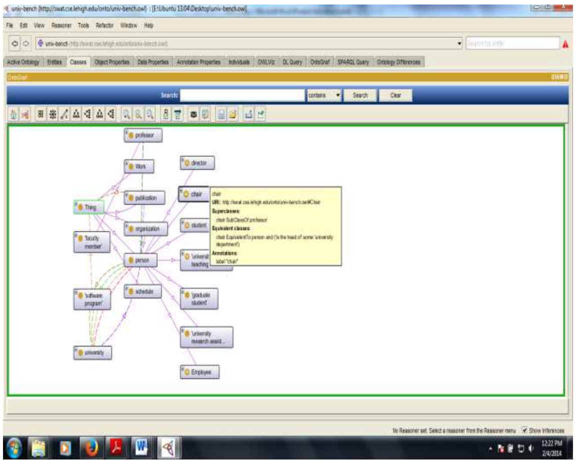

Fig 3: Graph view of Ontology in Protégé

4. LITERATURE REVIEW OF ONTOLOGY GENERATION SYSTEMS

Noreddine GHERABI, Khaoula ADDAKIRI, Mohamed BAHAJ[3] proposes a solution formigrating an RDB into Web semantic. The solution takes an existing RDB as input, and extracts its metadata representation (MTRDB). Based on the MTRDB, a Canonical Data Model (CDM) is generated. Finally, the structure of the classification scheme in the CDM model is converted into OWL ontology and the recordsets of database are stored in owl document. A prototype has been implemented, which migrates a RDB into OWL structure, for demonstrate the practical applicability of our approach by showing how the results of reasoning of this technique can help improve the Web systems.

Shufeng Zhou, Haiyun Ling, Mei Han, Huaiwei Zhang [4] provides unified ontology and improve the quality of ontology generation, approach proposed in this paper firstly extracts database metadata information from relational database using reverse engineering technique, and then analyzes the correspondent relationship between relational database and OWL ontology, and presents an ontology generation from relational database. Finally, a prototype tool of the generator, implemented based on Jena in Java development platform, and case study demonstrates the feasibility and effectiveness of the approach.

LEI ZHANG, JING LI[9] proposed to improve the efficiency of ontology construction for ontology applications. How to generate ontology automatically from database resources is an emerging task in ontology construction. Aiming at solving the problem, a method for automatic ontology building using the relational database resources to improve the efficiency is proposed in the paper. Firstly, mapping analysis of ontology and database is done. Secondly, construction rules of ontology elements based on relational database, which are used to generate ontology concepts, properties, axioms, instances are put forward. Thirdly，Ontology automatic Generation System based on Relational Database (OGSRD) is designed and implemented. Finally, the practical experiments prove the method and system feasibility.

Ivan Bedini ,Benjamin Nguyen[10] defined the requirements that an ontology must meet in order to fit these new use cases and provide a meticulous survey with a comparative analysis of experiences and software for automatic ontology generation, investigating in detail which aspects of ontology development can be done automatically and which ones require further research. The main contributions of this paper are the presentation of anew framework for evaluating the automation of ontology generation and an exhaustive comparative analysis of existing software geared towards automatic ontology generation.

RajiGhawi, Nadine Cullot[11] proposed a general interoperability architecture that uses ontologies for explicit description of the semantics of information sources, and web services to facilitate the communication between the different components of the architecture. It consists of 1) data provider services for mapping information sources to local source ontologies, 2) a knowledge base for representing reference domain ontology, and 3) several web services for encapsulating the different functionalities of the architecture. In this paper, focus on a component of the architecture which is a tool, called DB2OWL, that automatically generates ontologies from database schemas as well as mappings that relate the ontologies to the information sources. The mapping process starts by detecting particular cases for conceptual elements in the database and accordingly converts database components to the corresponding ontology components. A prototype of DB2OWL tool is implemented to create OWL ontology from relational database.

4.1 Major Purposes of Ontology

(1) Ontological analysis clarifies the structure of knowledge

Ontologies forms the heart of any system of knowledge representation. If do not have the conceptualizations that underlie knowledge, then do not have a vocabulary for representing knowledge.

(2) Ontologies enable knowledge sharing

Ontologies provide a means for sharing knowledge. Just described how demanding it can be to come up with the appropriate conceptualizations for representing some area of knowledge.

5. CONCLUSION

RGORDB generates ontology from relational database directly and automatically. At the same time, we consider that user intervention may be needed later to refine the generated ontology with the help of domain experts. Subsequently, we could get high quality ontology to provide better semantics for local database source in special domain. However, the ontology, generated fromrelational database using our generator, could be viewed as the local data source ontology without any instances, and several local ontologies could be integrated into a global ontology in the future for the entire system. For global ontology, the advantage of wrapping each information source to a local ontology is to allow the development of source ontology independently of other sources or ontologies. So we believe that this approach is more effective that a massive dump.

References:

[1] LudovicMenet, MyriamLamolle, Amar Zerdazi,(2008): Managing Master Data with XML Schema and UML, : International Workshop on Advanced Information Systems for Enterprises, iwaise, pp.53-59.

[2] Grigoris Antoniou and Frank van Harmelen, (2004): A Semantic Web Primer‖, The MIT

Press, Cambridvge, Massachusetts, London, England.

[3] Noreddine GHERABI, Khaoula ADDAKIRI, Mohamed BAHAJ,(Jan 2012) :Mapping relational database into OWL Structure with data semantic preservation‖ , International Journal of Computer Science and Information Security, Vol. 10, No. 1.

[4] Shufeng Zhou, Haiyun Ling, Mei Han, Huaiwei Zhang,(May 2010): Ontology Generator from Relational Database Based on Jena‖ , Computer and Information Science Vol. 3, No. 2.

[5] Fuxman, A., Hernández, M.A., Ho, H., Miller, R, Papotti, P., Popa, L.(2006): Nested Mappings: Schema Mapping Reloaded. Proc. VLDB Conf., pp. 67-78, Seoul, Korea.

[6] Kalfoglou, Y., and Schorlemmer, M.(2003): Ontology mapping: the state of the art. Knowledge Engineering Review, 18(1), 1-31.

[7] Miller, R., Haas, L., and Hernandez, M.A (2000): Schema Mapping as Query Discovery. Proc. VLDB Conf., pp. 77-88, Cairo, Egypt, 2000.

[8] Shvaiko, P., and Euzenat, J. A, (2005): Survey of Schema-Based Matching Approaches. J. Data Semantics IV 3730 ,146-171.

[9] Lei ZHANG, Jing LI,(Apr 2011), Automatic Generation of Ontology Based on Database , Journal of Computational Information Systems Vol. 7, No. 4.

[10] Ivan Bedini, Benjamin Nguyen,(2006), Automatic Ontology Generation: State of the Art‖, In Proceedings of the 2nd International Conference on Semantics Research.

[11] RajiGhawi, Nadine Cullot,(2007), Database-to-Ontology Mapping Generation for Semantic Interoperability‖, VLDB ’07, September 23-28.