

Building Domain Ontologies From Relational Database Using Mapping Rules

Maruf Pasha*, Abdul Sattar

Institute of Computing, Bahauddin Zakariya University, Multan.

** Corresponding author's Email: maruf.pasha@bzu.edu.pk*

Abstract: Ontologies have long been considered the core of semantics as they offer shareable and reusable knowledge about a particular domain. Improving and sharing domain specific knowledge residing in a database is one of the key challenges faced while developing any application. Due to the ever growing amount of data on the Web, it is almost impossible to extract meaningful data and the amount of manual work during creation of ontology poses series of challenges. And researchers are facing challenges such as the unavailability of well-formed databases, domain expert's help for extracting cardinality restrictions and generation of un-resolvable URIs. Therefore, we have focused on domain specific relational databases for constructing ontologies as a solution. Our aim is to analyze the various Ontology construction approaches from relational databases and identify the advantages and disadvantages of these techniques, so that an enhanced and efficient approach can be proposed. We have performed detailed analysis of various ontology construction techniques from relational database (RDB) based on database schema analysis (meta-data, cardinality restrictions and datatype information), stored data (through data mining) and also performed a comparative analysis of these techniques.

Keywords: Ontology; Web Ontology Language (OWL); Relational database; Ontology construction.

1. Introduction

Semantic Web provides technology to capture, share, and reuse structured and machine-readable domain specific knowledge and makes it available on web [1, 2]. Ontology is a description of domain-specific knowledge in order to share it with different applications [3]. "Ontology is a defined specification of conceptual model", defined by Gruber [4]. Machine and people also share information through ontology [5]. Main components of ontology (web) are: classes, subclasses, properties (datatype, object), and individuals [6, 7]. Ontology editors such as Protégé [8] are used to entering more elements in ontology for completeness. Current ontology usages are digital library, Semantic Web, and information intelligent retrieval system, etc [6]. Web Ontology Language (OWL) has been recommended by The World Wide Web Consortium (W3C) as a formal language for authoring ontology [7].

Improving and sharing domain specific knowledge that resides in database is one of the challenges in implementing applications. One of the ways for representing domain specific knowledge is to generate ontology from relational database (RDB). Moreover; RDBs provide accessibility and scalability of stored useful information [9] and serve as a useful data source for the surface web [10]. Many works have been done to add semantics to relational databases (RDBs) schemas [11].

In the schema mapping techniques, relation database schema is converted to ontology based on the mapping rules. One of the challenges in schema mapping technique is how accurately translation process encompasses all aspects represented by relational database schema. Various research groups have proposed various techniques to translate relational database schema components to ontology elements [8]. Proposed a tech-

nique to learn ontology from structured, semi-structured and un-structured data. Automapper [12] is a Semantic Web interface for RDBs to generate the data source and the respective mapping ontology from relational database automatically. In this technique, relational database schema is used to create OWL ontology. A rule based translation process of relational database schema to ontology maps data-source to domain. A Semantic Bridge module for relational database is used for translating queries into OWL ontologies.

In XTR-RTO [13], Extensible Markup Language (XML) document is used as source for generating OWL ontology. The transformation process converts XML schema to RDB schema and then, RDB schema to OWL ontology. Different attributes used to describe relational database are: DbName, Relation, Relation-List, Table and Attribute. RTAXON [14] uses data mining approach for ontology construction. This work presented a technique that mines (extracts) all database content to identify categorization patterns (subsumption relation). These categorization patterns are used to generate class hierarchies. Concept hierarchies can be generated based on the relationships among attributes of a relation. This approach uses lexical clues identified from attribute names that show specific role of attribute in the relations (i.e. categorizing the tuples in the relation). In [15], Li et al. presented an ontology learning technique that generates OWL ontology, based on learning rules from normalized relational database schemas (upto 3NF). Since these learning rules depend upon relational database schema, therefore learning rules are applied for concepts(classes), datatype and object properties (properties) and property characteristics (i.e. cardinalities that define property specifically with more details), subsumption relationship(class hierarchy) and data instances. Mapping rules for generating ontology are (1) all inter-related tables/relations combined to make single class; (2) Map each table or relation to a class that represents entity rather than relation between relations/tables; and (3) instances used to identify inclusion dependencies based on subsumption relationship of classes.

2. Ontology Construction

This section contains various approaches to ontology construction based on different characteristics.

2.1 Automapper

Automapper [12] is Semantic Web interface for generating ontology from relational database automatically. Automapper is an application independent tool

Table 1 Departments Table

ID*	NAME
1	System Solutions
2	Research and Development
3	Management

* ID is a Primary key

Table 2 Staffing Table

Staff Name	Project	DeptID	Hours	Role
Mattf	Alpha	1	100.5	Developer
MikD	Alpha	2	50.2	Tech Load
MattG	Beta	1	92.0	Architect
DaveK	Beta	1	120.0	Developer
MikD	Beta	2	30.8	Consultant
DaveK	Alpha	1	87.8	Indagator

that generates a basic ontology from a relational database schema and dynamically produces instance data using ontology.

The following class descriptions, axioms and restrictions are currently generated by Automapper:

- maxCardinality is set to 1 for all nullable columns and is used for descriptive purposes.
- minCardinality is set to 1 for all non-nullable columns and is used for descriptive purposes.
- All datatype and object properties that represent columns are marked as Functional Properties. To ensure global uniqueness and class specificity, these columns are given URIs based on concatenating the table and column names.
- All values from restriction reflect the datatype or class associated with each column and is used for descriptive purposes.

Table 1 and Table 2 list the contents of departments and staffing tables, respectively.

From this schema, Automapper creates the data source ontology and class-specific inverse functional rules, as shown in the Figure 1.

2.2 XTR-RTO

In XTR-RTO [13], Extensible Markup Language (XML) document is used as source for generating OWL ontology. The translation process converts XML schema to RDB schema and then, RDB schema to OWL ontology. Different attributes used to describe RDB are: DbName, Relation, RelationList, Table and Attribute.

```

dsont:Hresources.Departments a owl:Class;
  rdfs:subClassOf
  [
    a owl:Restriction;
    owl:onProperty
    dsont:hresources.departments.ID;
    owl:allValuesFrom xsd:decimal;
  ]
,
[
  a owl:Restriction;
  owl:onProperty
  dsont:hresources.departments.ID;
  owl:allValuesFrom xsd:decimal;
  "1"^^xsd:nonNegativeInteger
]
dsont:Hresources.Staffing a owl:Class ;
  rdfs:subClassOf
  [
    a owl:Restriction ;
    owl:onProperty
    dsont:hresources.staffing.name ;
    owl:maxCardinality
    "1"^^xsd:nonNegativeInteger
  ]
,
[
  a owl:Restriction ;
  owl:onProperty
  dsont:hresources.staffing.name ;
  owl:allValuesFrom xsd:string
]
,
[
  a owl:Restriction ;
  owl:onProperty
  dsont:hresources.staffing.deptid.Object ;
  owl:maxCardinality
  "1"^^xsd:nonNegativeInteger
]
.

```

```

<rd:Relation rdf:ID="BOOK">
  <rd:hasAttribute>
    <rd:AttributeList>
      <rd:Attributerdf:resource=
      "http://localhost/book/BOOK/BOOK.owl#BOOK_ID"/>
      <rd:Attribute rdf:resource=
      "http://localhost/book/BOOK/BOOK.owl#TITLE"/>
      <rd:Attribute rdf:resource=
      "http://localhost/book/BOOK/BOOK.owl#AUTHOR"/>
      <rd:Attribute
      rdf:resource="http://localhost/book/BOOK/BOOK.owl#
      PRINTER_ID

```

All tables are mapped to rdb: Relation and rdb: RelationList, where as each attribute is mapped to rdf: Attribute and rdb: hasType. The entity-relation model is used for organizing database, which clearly expresses the relationship between data. Therefore metadata information and structural restrictions are extracted from relational database to construct ontologies. The ontology contains:

- Vocabularies for describing relational database systems such as: rdb: DBName, rdb: Relation, rdb: RelationList, rdb: Table, rdb: Attribute, rdb: PrimaryKeyAttribute, and rdb: ForeignKeyAttribute.
- Semantic relationships between vocabularies such as: rdb: hasRelation, db: hasAttribute, rdb: primaryKey, db: hasType and rdb: isNullable.
- Restrictions on the vocabularies and their semantic relationships such as: each relation has zero or more attributes, and each attribute has exactly one type.

Mapping approach used in XTR-RTO is described below:

- Each table is mapped to an instance of type rdb: Relation and added to type rdb: RelationList.
- Each attribute is mapped to an instance of type rdb: Attribute and an instance of type rdb: hasType is generated simultaneously. If the attribute is the foreign key, an instance of type rdb: ReferenceAttribute and an instance of type rdb: ReferenceRelation are generated to represent this information.

Generate the restrictions of each instance of type rdb: Attribute, such as cardinality restriction and foreign key restriction. There is one table in this relational database, as illustrated in Table 3.

Figure 1 Ontology and inverse functional rules created by Automapper

Table 3 Book Table in Relational Model

BOOK_ID	TITLE	AUTHOR	PRINTER_ID
---------	-------	--------	------------

```
"/>
  </rdb:AttributeList>
</rdb:hasAttribute>
</rdb:Relation>
<rdb:PrimaryKey rdf:ID="BOOK_ID">
  <rdb:isNullable>>false</rdb:isNullable>
  <rdb:type>string</rdb:type>
</rdb:PrimaryKey>
<rdb:Attribute rdf:ID="TITLE">
  <rdb:isNullable>>false</rdb:isNullable>
  <rdb:type>string</rdb:type>
</rdb:Attribute>
<rdb:Attribute rdf:ID="AUTHOR">
  <rdb:isNullable>>false</rdb:isNullable>
  <rdb:type>string</rdb:type>
</rdb:Attribute>
<rdb:Attribute rdf:ID="PRINTER_ID">
  <rdb:isNullable>>false</rdb:isNullable>
  <rdb:type>string</rdb:type>
</rdb:Attribute>
<rdb:referenceAttribute>
  <rdf:Attribute rdf:resource=
http://localhost/book/PRINTER/PRINTER.owl
#PRINTER#PRINTER_ID/>
</rdb:referenceAttribute>
```

Figure 2 OWL description of BOOK Table

Suppose the database is saved in local host (mylocalpc), and the OWL ontology describing the database has a namespace "http://mylocalpc/book.owl", which is changeable as user desires. Figure 2 illustrates the relations in the ontology:

The ontology generated from table BOOK is shown in Figure 3.

2.3 RTAXON

A data mines approach for ontology construction. This technique mines (extracts) database contents to find categorization patterns (subsumption relation). These categorization patterns are then used to generate class hierarchies. Concept hierarchy identification by identifying the relations found in attributes of the relation is used as a base for learning ontology. Due to the assumption that attribute names tell meaning full role in the table/relation, lexical clues of attribute names

```
<?xml version="1.0" encoding="UTF-8">
<rdf:RDF>
  <rdb:DBName rdf:ID="BOOK">
    <rdb:hasRelation>
      <rdb:RelationList>
        <rdb:Relation rdf:resource=
http://localhost/book/BOOK/BOOK.owl
#BOOK/>
      </rdb:RelationList>
    </rdb:hasRelation>
  </rdb:DBName>
</rdf:RDF>
```

Figure 3 Relations in the ontology

are identified (i.e. for categorizing the tuples). In Figure 4, the Products relation contain Category attribute that serves as categorizing attribute. RDBToOnto [16] is a tool that uses RTAXON [14] method for converting RDB to ontology.

Section 1 presents the pattern identification procedure, whereas Section 2 discusses subclasses generation base on identified patterns.

1) Categorizing attributes identification:

There are two sources used to identify categorizing attributes: names of columns/attributes and data diversity in column.

a) Identification of lexical clues in attribute names:

The lexical clue that shows specific role of the attribute in the relation may be part of the name, as in the attribute names Category or Product Type. In Figure 4, for example, the Products contain Category attribute that is used as categorizing attribute.

b) Estimation of data diversity through entropy:

First filtering step helps to identify categorizing attribute using lexical clues. For example, Category column in the Products relation can be used to derive subclasses.

However, with complex or large databases, the first step often chooses several candidates. Then selecting suitable candidate based on data diversity in column data where selected candidate might have typical degree of redundancy identified by the concept of entropy. Entropy is a measure of the uncertainty of a data source. Attributes with highly repetitive content will be characterized by low entropy. Conversely, among attributes of a given relation, the primary key will have the highest entropy since all values in its extension are distinct.

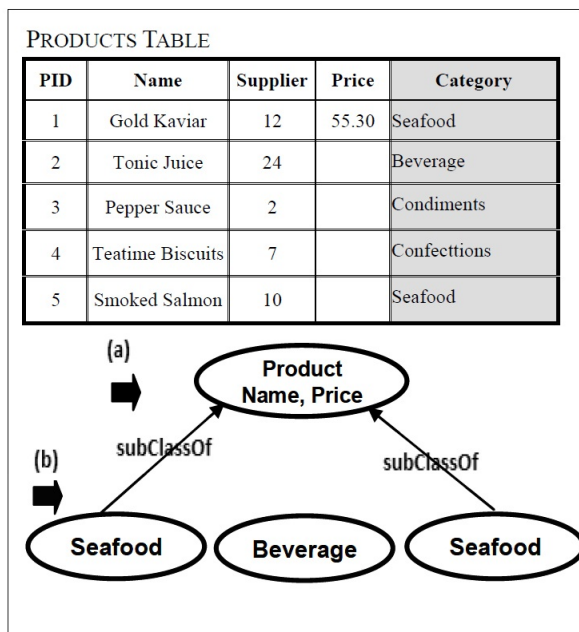


Figure 4 Categories employed for hierarchy generation

2) *Generation and population of the subclasses:*

Identified categorizing attribute is used to generate subclasses. A subclass is created based on each value type of the attribute (i.e. for each element of the attribute active domain). However, proper handling of the categorization source may require more complex mappings. As illustrated in Figure 4, in the first part (a), each relation (or table) definition from the relational database schema is the source of a class (or concept) in the ontology. All attributes other than reference keys translated into datatype properties. The primary-foreign key relations are the reliable source for associating classes and, in this example, each relationship is translated into an object property, where the (b) part shows how subclasses are generated based on categorizing attribute (Category) of Products table.

As illustrated in Figure 5, values used for subclass generation are extracted from other tables. In this example, CatId attribute used as categorizing attribute in Albums table/relation is linked with Categories relation through foreign key relationship. Categories relation contains all album categories.

2.4 Learning Ontology from Relational Database

An ontology learning technique generates ontology (OWL) automatically based on learning rules from normalized relational database schemas (upto 3NF). Since these learning rules depend upon relational database schema, therefore learning rules are applied to con-

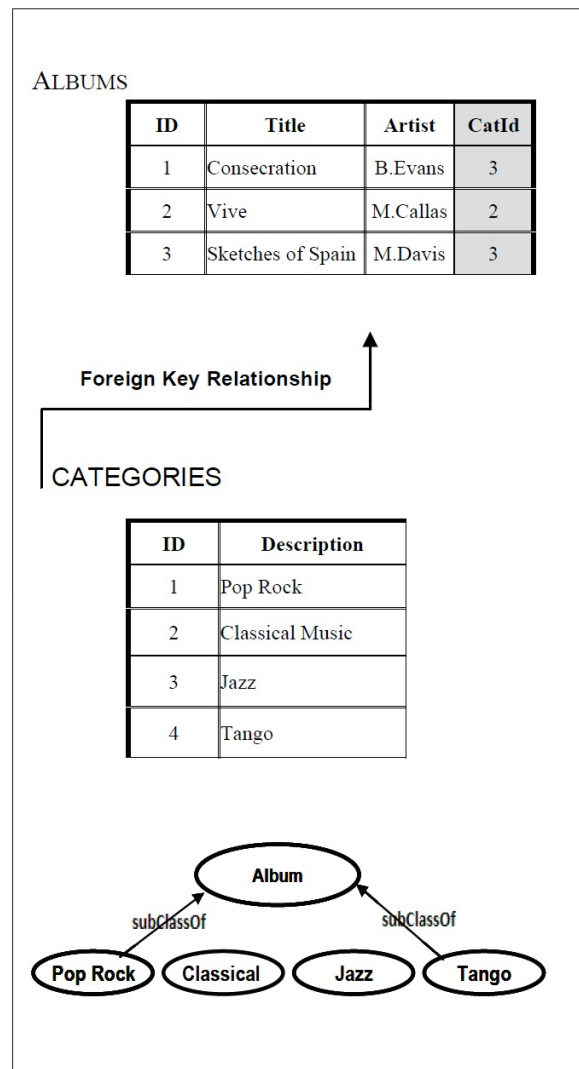


Figure 5 Categories employed for hierarchy generation are further defined in an external relation.

cepts (classes), datatype and object properties (properties) and property characteristics (i.e. cardinalities that define property specifically with more detail), subsumption relationship (class hierarchy) and data instances.

Mapping rules for generating ontology are (1) all inter-related tables/relations combined to make single class; (2) Map each table or relation to a class that represents entity rather than relation between relations/tables; and (3) instances used to identify inclusion dependencies based on subsumption relationship of classes.

Mapping rules are used to learn ontology from relational database. These rules are: rules for acquiring concepts (classes), datatype, and object properties, class-subclass relationships (hierarchy), cardinality and instances.

1) *Acquiring classes:*

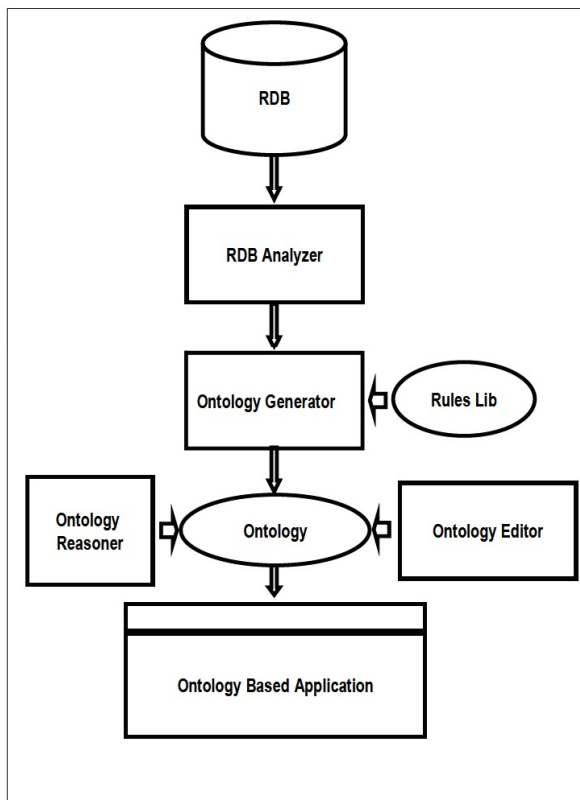


Figure 6 Ontology Learning Framework

During acquiring classes, several inter-related relations are combined to derive single class. These rules integrate the several inter-related relations into single class, when these relations/tables are used to describe single entity.

2) Learning properties and property characteristics:

Object and datatype properties are two types of properties. Reference relationship between relations is used to learn object properties. Complex or non-binary (n-ary) relations are not supported by ontology languages and converted to several binary relations. All attributes converted to datatype property except those used in object properties.

3) Learning hierarchy:

If two relations in database have primary-foreign key relationship then these two classes or properties are organized in a hierarchy.

4) Learning cardinality:

Cardinality restrictions are used to further specify properties of ontology, minCardinality and maxCardinality of the property will be 1 if attribute is primary or foreign key. The minCardinality of the property is 1 if any attribute is declared as NOT NULL. Furthermore, the maxCardinality of the property is 1, if any attribute is declared as UNIQUE.

5) Learning instances:

For each class in ontology, tuples in relations are translated into its instances and the instances are linked based on foreign keys in the database tuples.

6) Implementation:

The overall framework of ontology learning is presented in Figure 6. The input for the framework is data stored in relational database. The framework uses database analyzers to extract schema information from database, such as the primary keys, foreign keys and dependencies. Then the obtained information is transferred to ontology generator. The ontology generator will generate ontology based on the schema information and rules. As a last step, user can modify and refine the obtained ontology with the aid of ontology reasoner and ontology editor.

The framework is a domain/application independent and can learn ontology for general or specific domains from relational database.

3. Results and Discussions

This survey shows that each approach defines common rules for mapping basic RDB schema pattern to ontology such as relations, properties, reference keys and cardinalities as shown in the following Table 4.

Ontology built either from relational database using automatic or semi-automatic ontology construction or by using any ontology editor (from scratch) has attracted growing attention. However, the logic behind ontology construction is to provide concepts vocabularies and their relationships within a domain of discourse. As described in Table IV, the approaches used in ontology construction from relational database are based on schema mapping and data mining approaches.

Automapper [12] is a Semantic Web interface for generating ontology from relational database automatically. Translation process of RDB schema into ontology is based on mapping rules that maps data-source to domain. These mapping rules depend on well-formed relational database schema. In many applications, sometimes well-formed relational database schema is not available, thus it is not guarantee for better results in ontology construction process.

XTR-RTO [13] uses metadata information extracted from relational database to construct ontology based on predefined schema translation rules. Effectiveness of these translation rules depends upon well-formed relational database schema. Sometimes the unavailability of well-designed relational database results in challenges in ontology construction process.

RTXON [14] identifies lexical clues for attribute names

Table 4 Comparative Analysis of Ontology Construction Approaches

Approach	Mapping Creation	Procedure	Sources Used	Mapping Rules	Limitations
Automapper: Relational Database Semantic Translation using OWL and SWRL [12]	Automatic	-Construction of ontology from RDB with the help of Configuration file	-Configuration file -RDB Schema -Mapping rules	Class,datatype property, and object property	-Un-normalized RDB -Un-resolvable URIs
Using Relational Database to Build OWL Ontology from XML Data Sources [13]	Semi-Automatic	- Construction of ontology from RDB	-RDB schema -Mapping rules	Relation,Attribute, and RefAttribute	-Un-normalized RDB -Need domain expert's help for extracting cardinality restrictions
Mining the Content of Relational Database to Learn Ontology with Deeper Taxonomies [14]	Semi-Automatic	-Schema analysis and Hierarchy mining from stored data	- RDB schema - Mapping rules - Stored data	Class,datatype properties, object property	-Attribute name does not represent its value
Learning Ontology from Relational Database[15]	Automatic	- Construction of Ontology from RDB without using middle model	-RDB schema -Mapping rules	classes, properties, cardinalities and instances	-Un-normalized RDB

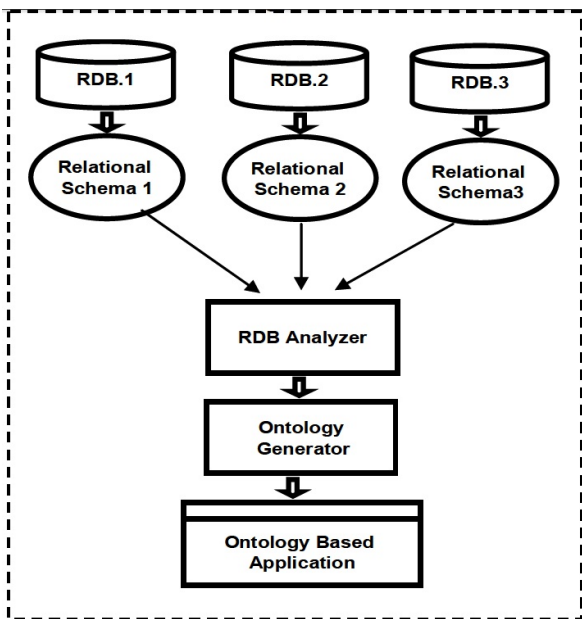


Figure 7 Framework for Ontology

(i.e. Category column represents a clue for categorizing products). However, attribute names sometimes do not reveal specific value (any lexical clue), thus better results cannot be guaranteed. The approach used in [15] generated an OWL ontology automatically based on learning rules from normalized relational database schemas. These learning rules depend upon relational database schema and in many applications, the unavailability of well-formed relational database results in inconsistent and incorrect ontology construction.

4. Proposed Framework

Table 4 summarizes each approach and defines common rules for mapping basic RDB schema pattern to ontology such as relations, properties, reference keys and cardinalities. Considering the above mentioned issues, we have proposed a novel framework that will take different normalized/un-normalized relational databases and will translate these relational databases into ontologies. Figure 7 illustrates the proposed framework.

Here we describe briefly how this framework will work:

- Get different relational database schema
- At second step relational database schema will be converted to ontology

5. Conclusion and Future Work

Mapping relational data into ontology from relational databases plays an important role during the creation and updating of ontology. The unaddressed question is how to correctly map RDB into ontology. The quantity of manual works during creation of ontology poses a series of challenges; therefore automated or semi-automated mapping based on rules is a good solution. In this paper, different approaches have been discussed and some of the approaches are also similar although ontology community relies more heavily on the higher expressive power of ontology languages and on reasoning techniques. In ontology construction process, researchers are facing challenges such as unavailability of well-formed databases, domain expert's

help for extracting cardinality restrictions and generation of un-resolvable URIs. On the other hand, ontology researchers are paying more attention toward further study of existing relational databases. It covers translation of additional relational database schema aspects to improve the ontology structure.

It shows that additional definition patterns learned from the data significantly enriches the ontology structure. Our future work in this domain is to focus on ontology construction from un-formed relational database and to explore how it is different between ontologies constructed and well-formed relational database.

References

- [1] John G. Breslin, D. O. Sullivan, A. Passant, L. Vasiliu, "Semantics web computing in industry", *School of Engineering and Informatics*, Galway, Ireland, 2010.
- [2] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web.", *Scientific American*, 2001.
- [3] C. Welty. "Ontology Research". *Artificial Intelligence (AI) Magazine*, 24(3), 2003.
- [4] Gruber TR. "A translation approach to portable ontology specifications". *Technical Report, KSL 92-71, Knowledge System Laboratory*, 1993.
- [5] GU Fang and CAO Cun-Gen, "Ontology Research and Existing Problems in Knowledge Engineering", *Computer Science*, 2004.
- [6] Deng ZH, Tang SW, Zhang M, Yang DQ, Chen J. "Overview of Ontology". *Acta Scientiarum Naturalium Universitatis Pekinensis*, 2002. (in Chinese with English Abstract)
- [7] P.F Patel-Schneider, P. Hayes and I. Horrocks (Recommendation 10 February 2004). "OWL Web Ontology Language Semantics and Abstract Syntax", W3C. from <http://www.w3.org/TR/owl-absyn/>.
- [8] DU Xiao-Yong, LI Man and WANG Shan, "A Survey on Ontology Learning Research", *Journal of Software*, Hangzhou, China, 2006.
- [9] C. Ritchie, "Relational Database Principles", *Thomson Learning*, 2003, ISBN 0-8264-5713-4.
- [10] K. C. C. Chang, B. He, C. Li, M. Patel, Z. Zhang, "Structured databases on the web: Observations and Implications", *Special Interest Group on Management of Data (SIGMOD) Record*, New York, USA, 2004.
- [11] Biskup, J. : "Achievements of relational database schema design Tudory revisited". *Semantics in Database LCNS 1358*. Springer Verlag, 1998.
- [12] M. Fisher, M. Dean, "Automapper: Relational database semantic translation using OWL and SWRL, OWLED 2008": *OWL experiences and Directions, Karlsruhe Germany 2008*. Available from: <http://www.webont.org/owlled.2008>.
- [13] J. Xu, W. Li, "Using relational database to build OWL ontology from XML data sources", *Proceeding International Conference on Computational Intelligence and Security Workshops*, Suzhou, 2007.
- [14] F. Cerbah, "Mining the content of relational database to learn ontology with deeper taxonomies", *International Conference on Web Intelligence and Intelligent Agent Technology*, Sydney, 2008.
- [15] M. Li, X. Du, S. Wang, "Learning ontology from relational database", *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, 2005.
- [16] F. Cerbah, "Learning highly structured semantic repositories from relational databases: The RDBToOnto", *Proceeding Extended Semantic Web Conference*, Tenerifi, 2008.