A New OWL2 Based Approach for Relational Database Description

Naïma S. Ougouti  
University of Sciences and Technology of Oran-Mohamed Boudiaf (USTO-MB)  
Faculty of Mathematics and Computer Science, LSSD Laboratory, Oran, 31000, Algeria  
E-mail: Souad.ougouti@univ-usto.dz

Hafida Belbachir  
University of Sciences and Technology of Oran-Mohamed Boudiaf (USTO-MB)  
Faculty of Mathematics and Computer Science, LSSD Laboratory, Oran, 31000, Algeria  
E-mail: h_belbach@yahoo.fr

Youssef Amghar  
LIRIS UMR 5205, Insa of Lyon, 69000, France  
E-mail: Youssef.amghar@insa-lyon.fr

Abstract—Nowadays, the scientific community is more and more interested by the mediation problem within Peer-to-Peer (P2P) systems and by data sources migration within the semantic web. Data integration and interoperability become a necessity to meet the need for information exchange between heterogeneous information systems. They reflects the ability of an information system to collaborate with other systems sometimes of a very different nature and aims at developing architectures and tools for sharing, exchanging and controlling data. In this context we have proposed a new heterogeneous and distributed data management system in a P2P environment called MedPeer. Among this system functions, we have focused in this article on relational databases description through the use of ontologies. We thus propose RelationalOWL2E, a new approach that, starting from the relational schema, generates an ontology based on the OWL2 language. Our main contribution lies in the semantics we have added to relational databases concepts in representing attributes by rich XML schema datatypes, primary keys, unique keys, foreign keys and by associating to each class a set of synonyms in order to guide the process of discovering semantic correspondences.

Index Terms—Semantic Web, Ontologies, Relational Databases, Web Ontology Language (OWL2), Schema Representation

I. INTRODUCTION

Since those past few years we have witnessed the emergence of new applications that need to share information between different systems. This is the case of e-government, e-learning, e-commerce, bioinformatics and electronic libraries. However, in this context, information systems, designed and developed by different organizations, generally constitute heterogeneous and autonomous data sources.

As commerce and computer science are developing rapidly, databases become more widely used and translating data between multiple distributed databases becomes a growing need, so database integration is long standing open problem with extensive research literature [1]. Thus, interoperability has become a necessity to meet the need for information exchange between heterogeneous information systems. It reflects the ability of an information system to collaborate with other systems sometimes of a very different nature and aims at developing architectures and tools for sharing, exchanging and controlling data.

Semantic web and ontologies give solutions for interoperability. The goal of Semantic Web is to add semantics to the existing data on the web and thus create an integrated web of data [2]. Ontologies are very useful in increasing Information Retrieval performance. they deals with occurrence of events, their instances and user defined relations between concepts. This represents background knowledge on Semantic level where Semantic level is defined as set of semantic entities including their concepts and relations instead of simple words which are used in thesaurus [3].

In this context, we have introduced a new data integration system in a P2P environment named MedPeer [4]. It has a Super-peer architecture based on peers regrouping according to media type (Texts, Images, Relational databases, semi-structured...). Super-peers form between them a pure P2P network. This architecture combines a centralized approach with a non structured one thus providing the advantages of centralized research such as autonomy, tasks distribution and robustness for a distributed research. Each super-peer manages the peers containing the same type of media it represents; it is selected according to its calculation capacities and bandwidth. In addition, it must have all necessary information to be able to direct requests arriving to it towards relevant peers. Semantic mediation is essential because schema sources are different. This function is achieved by a source description module that has for principal goal to regulate peers syntactic and semantic heterogeneity.
problem in a community. Each peer data source will be described by an ontology using our new approach.

These ontologies will be regularly sent to the super-peer community, to enable it to generate semantic correspondences with domain ontology. All this permits to deal with possible data sources modifications and with system dynamicity.

In this article, we will focus on this latter problem by presenting a new relational schema representation format based on the OWL Web Ontology Language in its second version named Relationnal.OWL2E. By exploiting the different opportunities provided by OWL2 [5], and our ontology, we are now able to describe and share any relational database schema.

This paper is organized as follows:

In Section 2, we will present a state of the art of the main approaches that describe relational databases with ontologies. In section 3, we will introduce Relationnal.OWL2E our new OWL2 based approach for relational database description. In Section 4, we will illustrate our approach with an example before our conclusion.

II. STATE OF THE ART

Wanting to take advantage from the benefits brought by the Semantic Web, several works the goal of which is the passage from a relational database to a newer format (XML / RDF / OWL) have emerged. We have chosen to present six approaches [6][7][8][9][10][11], more recent methods are in [12][13][14][15].

Relationaal.OWL [6], translates the majority of relational model concepts into OWL, from relational schema to data including integrity constraints. This system defines four classes and a set of properties allowing to link them together.

In Table 1 are listed the predefined classes and Table 2 contains the different properties.

The prefixes rdf, rdfs, dbs, xsd and owl represent namespaces used in the Relationnal.OWL ontology.

<table>
<thead>
<tr>
<th>Classes of Relationnal.OWL</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdf:ID</td>
</tr>
<tr>
<td>dbs:Database</td>
</tr>
<tr>
<td>dbs:Table</td>
</tr>
<tr>
<td>dbs:Column</td>
</tr>
<tr>
<td>dbs:PrimaryKey</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Properties of Relationnal.OWL</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdf:ID</td>
</tr>
<tr>
<td>dbs:has Table</td>
</tr>
<tr>
<td>dbs:has Column</td>
</tr>
<tr>
<td>dbs:isIdentified By</td>
</tr>
<tr>
<td>dbs:references</td>
</tr>
<tr>
<td>dbs:scale</td>
</tr>
<tr>
<td>The scale an entry of the Column may have.</td>
</tr>
</tbody>
</table>

OntoGrate [7], is a relational database integration system in a P2P (Peer-to-Peer) environment. To represent relational schemas in OWL, the authors have extended the expressiveness of the web ontology language. They thus have introduced a new language, Web-PDDL, an extension of PDDL (Planning Domain Definition Language) based on the logic applied to first order predicates. At first, the database concepts are translated through the use of the Web-PDDL language. Once the ontology generated, the system has a syntax adapter named PDDOWL, which translates the first Web-PDDL ontology into OWL ontology. In the final Generated ontology, a table is transformed into a class, subclass of the class sql: relationship ( Defined in OntoGrate system as the class representing tables), an attribute is transformed into an OWL property, a constraint is seen as an axiom (rule) and a primary key constraint as a functional OWL constraint (owl: FunctionalProperty ).

RDF Gateway [8], is a system that translates a relational database schema into RDFS or OWL ontology via the schema_type parameter, which specifies the ontology default output.

The SQL Data service is a module of RDF Gateway system that queries the database and extracts the relational schema then transforms it into RDFS or OWL ontologies. In this system a table is translated into a class, an attribute into a property rdfs:property for an RDFS output or owl:DatatypeProperty for OWL output, a foreign key into a property rdfs:property or owl:ObjectProperty and finally the datatype of attributes are translated into XML Schema datatypes.

OWL_K (K for Key) [9], is an extension of OWL to manage identification constraints which are equivalent to primary keys of the relational model. This work was motivated by the difficulties of the OWL DL dialect to capture their semantics. The default vocabulary of OWL was extended to take into account these constraints.
The system proposes:

- **The ICAssertion class** which represents the identification constraint.
- The property **onClass** which is the class (table) on which falls the identification constraint.
- The property **byProperty** which represents a property (attribute) participating to the identification constraint.

The default OWL description logic language has also been extended to take into account the new concepts semantic.

In this system, datatypes are translated by using XML Schema and foreign keys are translated by using cardinality constraints (owl:minCardinality, owl:maxCardinality).

“Reference [10]” developed a tool called DB2OWL to create ontology from a relational database. It looks for some particular cases of database tables to determine which ontology component has to be created from which database component. The created ontology is expressed in OWL-DL language which is based on Description Logics. The mapping process starts by detecting some particular cases for tables in the database schema. According to these cases, each database component (table, column, constraint) is then converted to a corresponding ontology component (class, property, relation). The set of correspondences between database components and ontology components is conserved as the mapping result to be used later.

R2O [11] is an extensible, fully declarative language to describe mappings between relational DB schemas and ontologies. It is intended to be expressive enough to describe the semantics of these mappings. R2O is a RDBMS independent high level language that works with any DB implementing the SQL standard. Its main features are:

1) Its mapping defines how to create instances in the ontology in terms of the data stored in the DB.
2) Its mapping definition can be used to automatically populate an ontology with instances extracted from the DB content and can also be used to automatically characterize data sources to allow dynamic query distribution in intelligent information integration approaches.

### III. RELATIONAL.OWL2E

The solution we are proposing here is an extension of the Relational.OWL proposed system [6]. We chose this approach because of its specificity to translate almost all the concepts of the relational model in OWL ontologies.

Our main contribution lies in the semantics we have added to relational databases concepts in representing attributes by rich XML schema datatypes, primary keys, unique keys, foreign keys and taking into account the NULL and NOT NULL constraints of the relational model. We have also associate to each class a set of keywords (synonyms) in order to capture the semantics of the terms used to guide the process of discovering semantic correspondences.

We called this ontology Relational.OWL2E because it is based on OWL2 and Relational.OWL that we have extended (E).

We obtain information on the database content from its data dictionary (catalog), and then we generate the corresponding ontology by translating tables, attributes (columns), datatypes (possibly with length restrictions), primary keys, unique keys and foreign keys into ontology concepts.

We thus defined 5 classes and 9 properties between them; they are summarized in the two following tables:

<table>
<thead>
<tr>
<th>Classes in Relational.OWL2E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
</tr>
<tr>
<td>Table</td>
</tr>
<tr>
<td>Column</td>
</tr>
<tr>
<td>PrimaryKey</td>
</tr>
<tr>
<td>UniqueKey</td>
</tr>
</tbody>
</table>

#### Table 4. Properties in Relational.OWL2E

<table>
<thead>
<tr>
<th>Properties</th>
<th>rdfs:domain</th>
<th>rdfs:range</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>HasTable</td>
<td>Database</td>
<td>Table</td>
<td>A Database belongs to a set of Tables.</td>
</tr>
<tr>
<td>hasColumn</td>
<td>Table PrimaryKey UniqueKey</td>
<td>Column</td>
<td>A Table belongs to a set of Columns.</td>
</tr>
<tr>
<td>hasPrimaryKey</td>
<td>Table</td>
<td>PrimaryKey</td>
<td>A Table is identified by a Primary Key</td>
</tr>
<tr>
<td>hasUniqueKey</td>
<td>Table</td>
<td>UniqueKey</td>
<td>A table may have unicity constraints on certain attributes</td>
</tr>
<tr>
<td>hasForeignKey</td>
<td>Table</td>
<td>Table</td>
<td>A table references another table in a foreign key relation.</td>
</tr>
<tr>
<td>References</td>
<td>Column</td>
<td>Column</td>
<td>A column references another column in a foreign key relation.</td>
</tr>
<tr>
<td>hassynonym</td>
<td>Database Table Column</td>
<td>Rdfs:literal</td>
<td>The name of a database, a table or a column may have synonyms</td>
</tr>
<tr>
<td>Isa</td>
<td>Table</td>
<td>Table</td>
<td>hierarchical relationship between two tables</td>
</tr>
</tbody>
</table>

#### A. Relational.OWL2E Ontology Serialization

In what follows we will give a few Relational.OWL2E ontology extracts, in RDF/XML syntax.

```xml
<owl:Class rdf:ID="Table"/>
```
B. Translating algorithm

First part

1. Firstly, all tables are extracted from a database.
2. The Database name represents a Database class in the relational_owl2e ontology.
3. Each table name will be expressed as a table class name having as property value.
4. From each table are extracted attributes, primary, unique and foreign keys.
5. Each attribute name will be expressed as a hasColumn property value.
6. The primary key will be expressed by the hasPrimaryKey property on the PrimaryKey class containing the list of attributes participating in the key, each attribute being expressed as a hasColumn property value.
7. The unique key will be expressed similarly as the primary key, but with the hasUniqueKey property on the UniqueKey class containing the list of attributes participating in the key, each attribute being expressed as a hasColumn property value.
8. The foreign key will be expressed by the hasForeignKey property. This property value will be the table referenced by the foreign key. Each foreign key attribute will be expressed as a Column class instance and linked to the referenced column by the references property having for value the referenced attribute column.

The result of this algorithm first part will be an ontology that describes all the database schema concepts. Attributes datatypes are treated in the algorithm second part.

Second Part

1) Each attribute will be expressed as a datatype property whose domain (rdfs:domain) is the name of the class representing the table containing the attribute and its image (rdfs:range). Its datatype is expressed with XML Schema datatypes in the following way:

- Integer is expressed by the XML Schema integer datatype, with possible restrictions on datatype values intervals, thanks to the XML schema facets minInclusive, maxExclusive, minInclusive, minExclusive.
- Decimals are expressed by the decimal XML schema datatype, with possible restrictions, thanks to the totalDigits and fractionDigits facets.
- String will be expressed by the string XML schema datatype. We use minLength and maxLength facets to express the minimum and maximum number of characters allowed. For the minLength facet value, if the attribute accepts null values, then minLength will be 0, otherwise 1.
- The Set datatype, is translated into a string datatype, its maxLength facet value will be extracted from the MySQL catalog.
- The Enum datatype, will be expressed by the owl:oneOf property composed of the different enum attribute values.
- Temporal datatypes will be expressed by one of the many temporal XML schema datatypes.
- Binary datatypes will be expressed by the hexBinary XML schema datatype. The minLength facet value is 0 if the attribute value is null, 1 otherwise.

2) The primary key will be expressed by OWL2 owl:Haskey property on the class name representing the table containing this key and having for values the list of attributes participating in the primary key.
3) Each unique key (name) will be expressed as a subclass (of the class containing the unique key) containing the owl: Haskey property having for value the list of attributes participating in the unique key.
4) Foreign keys will be expressed by owl:Restriction property on the name of each attribute participating in the key (owl:onProperty) towards the referenced attribute (owl:someValuesFrom).

IV. EXAMPLE

This section provides an example on how to represent the schema of existing databases using Relational_owl2e. Firstly, we will present the relational schema to describe: it is a MYSQL relational database ‘Breeding’ which contains three tables ‘species’, ‘Race’ and ‘Animal’ then we will give some extracts from the generated OWL2 ontology.

A. Relational schema to describe

Create database Breeding;
Create table Species (id smallint(6) not null auto_increment, latin_name varchar(40) not null, primary key(id), unique key latin_name (latin_name));
Create table Race (id smallint(6) not null auto_increment, species_id smallint(6), primary key(id), constraint fk_race_species_id foreign key(species_id) references Breeding(id));
Create table Animal(id smallint(6) not null auto_increment, sex enum('male', 'female'), Birth_date datetime not null,
B. A few relational.OWL2E ontology extracts

Database Description
<owl:Class rdf:ID="Breeding">
  <hasTable rdf:resource="#Species" />
  <hasTable rdf:resource="#Race" />
  <hasTable rdf:resource="#Animal" />
</owl:Class>

Table Description
<owl:Class rdf:about="#Species">
  <hasPrimaryKey>
    <PrimaryKey>
      <hasColumn rdf:resource="#Species.id" />
    </PrimaryKey>
  </hasPrimaryKey>
  <hasUniqueKey>
    <UniqueKey>
      <hasColumn rdf:resource="#Species.latin_name" />
    </UniqueKey>
  </hasUniqueKey>
</owl:Class>

Attribute Description
<owl:DatatypeProperty rdf:ID="Species.id">
  <rdfs:domain rdf:resource="#Animal" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#short" />
</owl:DatatypeProperty>

Primary Key and Unique Key
<owl:Class rdf:about="#Species">
  <hasKey rdf:parseType="Collection" />
</owl:Class>

Foreign Key
<owl:Class rdf:about="#Race">
  <hasKey rdf:parseType="Collection" />
</owl:Class>

Copyright © 2015 MECS
V. CONCLUSION

The majority of the data on the web resides in Relational databases, the success of the Semantic Web hinges on offering efficient ways of integrating relational databases into the semantic web. This requires a prior description of them. In this work, we have presented our Relational.OWL2E new approach which generates the correspondent OWL2 ontology to a relational database schema. We tried to find the best OWL2 constructors to best express the relational concepts semantic.

The provided ontology can be improved to deal with other specific concepts and semantic properties. Our algorithm can be easily implemented for database management systems other than MySQL.

REFERENCES


Authors’ Profiles

Naïma S. Ougouti is born in 1971. She Holds an Engineer degree in the field of computer science in 1994 at Es-senia university of Oran (Algeria), then a Magister degree in 2004 at the university of sciences and technology of Oran –Mohamed Boudiaf (USTO-MB). She worked as Engineer of computer science in a big Algerian petroleum company from 1996 to 2005 then since 2006, she is an Assistant Professor at (USTO-MB) university, Her field of teaching concerns operating systems, networks and databases. She is also a Member of LSSD laboratory and an active participant in some research projects. Her current research interests include semantic web, information retrieval and interoperability.

Hafida Belbachir is born in 1955. She is Professor of computer Sciences at the University of Sciences and Technology of Oran–Mohamed Boudiaf (USTO-MB) in Algeria. She received her PHD in Computer Science at University of Oran in 1990. She heads the Database System Group in the LSSD Laboratory. Her research interests include Advanced Databases, DataMining and Data Grid. She is author of more than 80 papers in reviews and proceedings.

Youssef Amghar is born in 1956. He is Professor of management information systems at the Scientific and Technical University of Lyon where he is the head of Computer Science Department. He holds a PhD in Computer Science from the same university in 1989 following by an HDR in 1997. His field of teaching concern project management, databases and development processes. His is an active member of laboratory of information system of INSA de Lyon. His current research interests include information retrieval, interoperability of applications and legal documents. He is author of more than 100 papers related to these research activities and managed some projects about decisions support. Actually, he is responsible of research team working on the domain of service oriented computing.