

# Automatic Transformation of Relational Database Schema into OWL Ontologies

**Imants Zarembo**

*Rezekne Higher Education Institution*

**Abstract.** Ontology alignment, or ontology matching, is a technique of mapping different concepts between ontologies. For this purpose at least two ontologies are required. In certain scenarios, such as data integration, heterogeneous database integration and data model compatibility evaluation, a need to transform a relational database schema to an ontology can arise.

To conduct a successful transformation it is necessary to identify the differences between relational database schema and ontology information representation methods, and then to define the transformation rules. The most straight forward but time consuming way to carry out transformation is to do it manually. Often this is not an option due to the size of data to be transformed. For this reason there is a need for an automated solution.

The automatic transformation of OWL ontology from relational database schema is presented in this paper; the data representation differences between relational database schema and OWL ontologies are described; the transformation rules are defined and the transformation tool's prototype is developed to perform the described transformation.

**Keywords:** ERD, OWL, transformation.

## I INTRODUCTION

Relational databases are one of the most popular storage solutions for all kinds of data. To fully realize vision of the semantic web, the gap between legacy relational databases must be narrowed. The problem of determining the rules of how to map an entity from a legacy relational database to an equivalent entity in an ontology is commonly referred to as the database-to-ontology mapping problem. It is important to note that the term mapping has been interchangeably used for two different problems: mapping and transformation. A mapping maps the entity of one data source to at most one entity of another. A mapping can be viewed as a collection of mapping rules all oriented in the same direction. A transformation is the process of expressing the entities of data source with respect to the entities of another data source. In this paper, the mapping as a transformation is considered.

The transformation process is the first step towards relational database integration into the semantic web. Other tasks such as ontology matching requires that the relational database data model is transformed into an ontology before ontology matching can take place.

The use of ontologies is important for many rapidly expanding technologies as they provide the critical semantic foundation. In the context of computer and information science, the ontology defines a set of representative primitives using which a domain of knowledge is modeled. In comparison with relational databases, Web Ontology Language (OWL) is fully

built upon formal logic, which allows logical reasoning. The reasoning can help to validate the consistency of the conceptual model and to find the properties and the relationships, which were not previously apparent. Another important benefit of using ontologies are the semantic search capability. Semantic search improves search accuracy by understanding the contextual meaning of terms as they appear in the searchable data.

This paper presents an automatic transformation of OWL ontology from relational database schema. The transformation process applies the transformation rules to the entity relationship diagrams and creates appropriate ontology entities.

## II WEB ONTOLOGY LANGUAGE

The OWL 2 Web Ontology Language, informally OWL 2, is an ontology language for the Semantic Web with formally defined meaning [1]. OWL ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents. OWL ontologies can be used along with information written in Resource Description Framework (RDF). OWL ontologies themselves are primarily exchanged as RDF documents.

OWL allows for greater machine interpretability of Web content than the content supported by XML, RDF and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages:

OWL Lite, OWL DL and OWL Full. OWL DL is used for the purposes of this paper to support the maximal expressiveness without losing computational completeness.

### III RELATIONAL DATABASE SCHEMA

Many legacy systems have been documented using Entity Relationship Diagrams (ERD) or Extended ER diagrams.

An Entity Relationship Diagram is a representation of data within a domain, a visual form of relational database. It consists of entities and relationships between these entities.

For legacy systems to be used as a component of emerging semantic web, these systems need to be upgraded. Both ERD and OWL represent entities and their relationships which provides opportunity of transformation of ERD to OWL ontology.

### IV RELATED WORK

There is a considerable amount of research on the topic of transforming and mapping of relation databases to ontologies. Approaches vary dramatically, ranging from manual transformation to relational database structure duplication using an ontology [2]. Although relational database transformation and mapping techniques have different goals in mind, similar results can be achieved by mapping relational databases to RDF datasets. The mapping produces correspondences between a relational database and an ontology, while transformation uses relational database to create a new object – an ontology.

In their paper [3] Zhang and Li analyse mappings and construct rules based on relational database to generate ontology concepts, properties, axioms and instances. The authors of [4] propose discovering mappings between relational database schema and ontology by exploiting mappings based on virtual documents and mapping consistency validation. Alalwan et al. in [5] propose automatic transformation system based on Structure Query Language (SQL) and metadata to extract semantic aspects which otherwise could not be inferred from the SQL. In [6] the authors propose an approach of learning OWL ontology from data in relational database. The approach proposes a technique that can be used to acquire ontology from relational database automatically by using a group of learning rules instead of using a middle model. The approach presented in [7] proposes to extract database metadata information from relational database using reverse engineering technique, and then to analyse the corresponding relationship between relational database and OWL ontology. The author of [8] maps relational data into ontology and fills ontology with data from relational database using mapping rules. Astrova et al. [9] propose to map relation database

constructs (tables, columns, data types, constraints and rows) to an ontology.

World Wide Web Consortium (W3C) offers several recommendations in relation to mapping of relational databases to RDF. Recommendation “A Direct Mapping of Relational Data to RDF” [10] provides simple direct mapping from relational data to RDF. In [11] RDB to RDF mapping language is described, which allows creating customized mappings from relational databases to RDF datasets in the form of graphs.

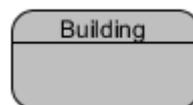


Fig. 1. ER class example

```
<owl:Class rdf:about="#Building">
</owl:Class>
```

Fig. 2. OWL class example

The D2RQ Mapping Language [12] provides a declarative mapping language for mapping relational database schemas to RDF vocabularies and OWL ontologies. The mapping defines a virtual RDF graph that contains information from the database.

### V MATERIALS AND METHODS

Relational databases and ontologies are designed with different goals in mind, however they are similar. To successfully transform data model represented by relational database schema into OWL ontology (Fig. 3), a set of transformation rules must be defined. During the transformation the differences in data models can lead to data or semantic meaning loss.

The transformation rules must take into account the differences between both information representation methods. The rules defined in this paper describe how to transform each of the major relational database constructs into an OWL ontology.

The created ontology is described in OWL-DL language, which is recommended by W3C for publishing and sharing ontologies on the web. OWL-DL is based on Description Logics [13] and has computational properties for reasoning systems.

#### A. Classes

Ontology class (Fig. 2) creation depends on particular database table cases (Fig. 1) [14]. Three different cases are taken into account in the proposed approach:

##### Case 1

For the first case to occur, a table must be used only to relate other tables in a many-to-many relationship. (Fig. 4).

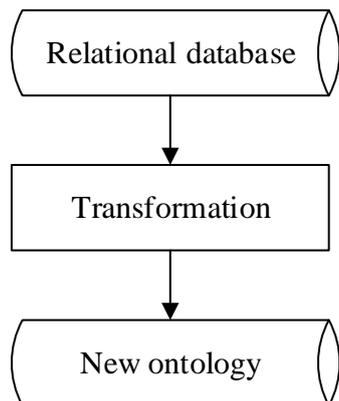


Fig. 3. Transformation process

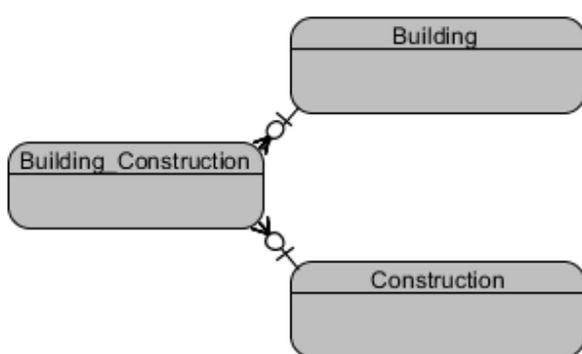


Fig. 4. ER class relationship example

```

<owl:ObjectProperty
  rdf:about="Building_Construction">
  <rdfs:range
    rdf:resource="#Building"/>
  <rdfs:domain:resource="#Construction"/>
</owl:ObjectProperty>
  
```

Fig. 5. OWL class relationship example

The relationship can be divided into two disjoint subsets of columns, each take part in a referential constraint with the related tables (Fig. 5).

For that reason all columns in the table are considered foreign and primary keys, because their combination uniquely defines the rows of the table.

*Case 2*

For the second case to occur, a table must be related to another table by a referential integrity constraint. In this case all primary keys of the table are considered foreign keys.

*Case 3*

If none of the previous cases occur, the third case is considered.

*B. Attributes*

An entity can have one of several types of attributes [16]:

- Simple attributes;
- Composite attributes;
- Multi-valued attributes.

*Simple attributes*

Simple attribute of entity (Fig. 6) can be mapped into datatype property of corresponding OWL class (Fig. 7). The domain of the datatype property is set to the entity it belongs and range is the actual datatype of that attribute. The ER diagram data type values can be mapped to XSD datatypes in OWL using the mappings shown in Table 1.

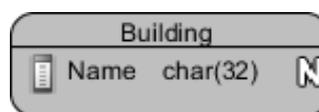


Fig. 6. ER simple attribute example

```

<owl:DatatypeProperty
  rdf:about="hasBuildingName">
  <rdfs:range
    rdf:resource="#string"/>
</owl:DatatypeProperty>
  
```

Fig. 7. OWL simple attribute example

TABLE I  
MAPPINGS BETWEEN SQL AND XSD DATATYPES

SQL	XSD
integer/int	Xsd:integer
float	Xsd:float
char/varchar/vchar	Xsd:string
time	Xsd:time
date	Xsd:date
datetime	Xsd:datetime
boolean	Xsd:boolean

*Composite attribute*

There are two ways to map a composite attribute to OWL datatype property:

- To transform only simple component attributes of composite attribute to datatype properties of corresponding OWL class, but ignore the composite attribute itself.
- To transform composite attribute to datatype property and then map its component attributes to subproperty of corresponding datatype property.

The first approach is preferable for relational database transformation because relational schema has only instances of simple, component attributes. Composite attributes are ignored.

### Multi-valued attribute

Databases cannot handle multi-valued attributes efficiently, while ontologies have an adequate way to deal with them. In the proposed approach multi-valued attributes are put in separate relations to avoid tuple duplication.

Multi-valued attributes are mapped to datatype property, but without a “functional” tag.

### C. Generalization and specialization

Subtype relations in the ERD (Fig. 8) can be transformed to subClassOf in the OWL ontology (Fig. 9). OWL subClassOf represents the generalization or specialization hierarchy.

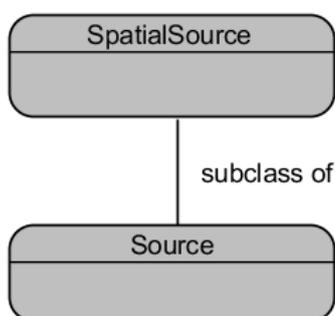


Fig. 8. ERD subclass example

```

<owl:Class rdf:about="SpatialSource">
  <rdfs:subClassOf
    rdf:resource="#Source"/>
</owl:Class>
    
```

Fig. 9. OWL subclass example

### D. Enumerator

In ERD enumerators are simple classes, often structurally indistinguishable from other classes. For that reason it is not always possible to automatically detect if the class contains an enumerator.

To transform ERD enumerator class to OWL ontology, the class must first be specifically marked and the transformation algorithm must be able to identify the type of marking to perform a successful transformation. Types or markings can vary. One of the more common approaches is to mark table name with special suffix or postfix, which can be uniquely identified by the transformation tool.

Enumerations can be transformed into named data types in OWL (Fig. 10). The value of this built-in OWL property must be a list of individuals that are the instances of the class. First existence of the data type must be declared, then the definition of the data type can be given using the "DatatypeDefinition" axiom. The allowed values of the Enumeration are listed in the "DataOneOf" statement.

```

<owl:Class rdf:ID="CoveregeType">
  <rdfs:subClassOf>
  <owl:Class>
    
```

```

<owl:oneOf
  rdf:parseType="Collection">
  <owl:Thing rdf:about="#forest"/>
  <owl:Thing rdf:about="#water"/>
  <owl:Thing rdf:about="#grass"/>
</owl:oneOf>
</owl:Class>
</rdfs:subClassOf>
</owl:Class>
    
```

Fig. 10. OWL enumerator example

### E. Cardinality

In OWL owl:cardinality is a built-in property that links a restriction class to a data value belonging to the range of XML Schema datatype nonNegativeInteger. OWL 2 contains six different cardinality axioms, three for object properties and three for data properties:

- “ObjectMinCardinality”;
- “ObjectMaxCardinality”;
- “ObjectExactCardinality”;
- “DataMinCardinality”;
- “DataMaxCardinality”;
- “DataExactCardinality”.

```

<owl:ObjectProperty rdf:about="Diameter">
  <rdfs:domain
    rdf:resource="#NetworkSegment"/>
  <rdfs:range>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource="#hasDiameter"/>
      <owl:qualifiedCardinality
        rdf:datatype="integer">1
      </owl:qualifiedCardinality>
      <owl:onDataRange
        rdf:resource="#float"/>
    </owl:Restriction>
  </rdfs:range>
</owl:ObjectProperty>
    
```

Fig. 11. OWL exact cardinality example

In OWL 2 a cardinality constraint puts a restriction on the number of values that the property can take, in the context of this particular class description.

```

<owl:ObjectProperty
  rdf:about="BuildingName">
  <rdfs:domain rdf:resource="#Address"/>
  <rdfs:range>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource="#hasBuildingName"/>
      <owl:minQualifiedCardinality
        rdf:datatype="integer">0
      </owl:minQualifiedCardinality>
      <owl:onDataRange
        rdf:resource="#string"/>
    
```

```

</owl:Restriction>
</rdfs:range>
<rdfs:range>
<owl:Restriction>
<owl:onProperty
  rdf:resource="#hasBuildingName"/>
  <owl:maxQualifiedCardinality
    rdf:datatype="integer">1
  </owl:maxQualifiedCardinality>
  <owl:onDataRange
    rdf:resource="#string"/>
  </owl:Restriction>
</rdfs:range>
</owl:ObjectProperty>

```

Fig. 12. OWL exact cardinality example

A restriction containing a “MinCardinality” constraint describes a class of all individuals that have at least N semantically distinct values for the property concerned, a “MaxCardinality” constraint describes a

class of all individuals that have at most N semantically distinct values for the property concerned.

An “ExactCardinality” constraint (Fig. 11) describes a class of all individuals that have exactly N semantically distinct values for the property concerned, where N is the value of the cardinality constraint [1].

In the case when "MinCardinality" and "MaxCardinality" constraints have the same values they can be replaced with the single "ExactCardinality" constraint to make ontology more concise (Fig. 12). In the case when the cardinality upper limit is 1, the property can be marked functional by adding a "FunctionalObjectProperty" or "FunctionalDataProperty" to object or data properties of the ontology accordingly.

If an attribute is mandatory its cardinality is always 1. If an attribute is not mandatory its minimal cardinality is 0 and maximal cardinality 1.

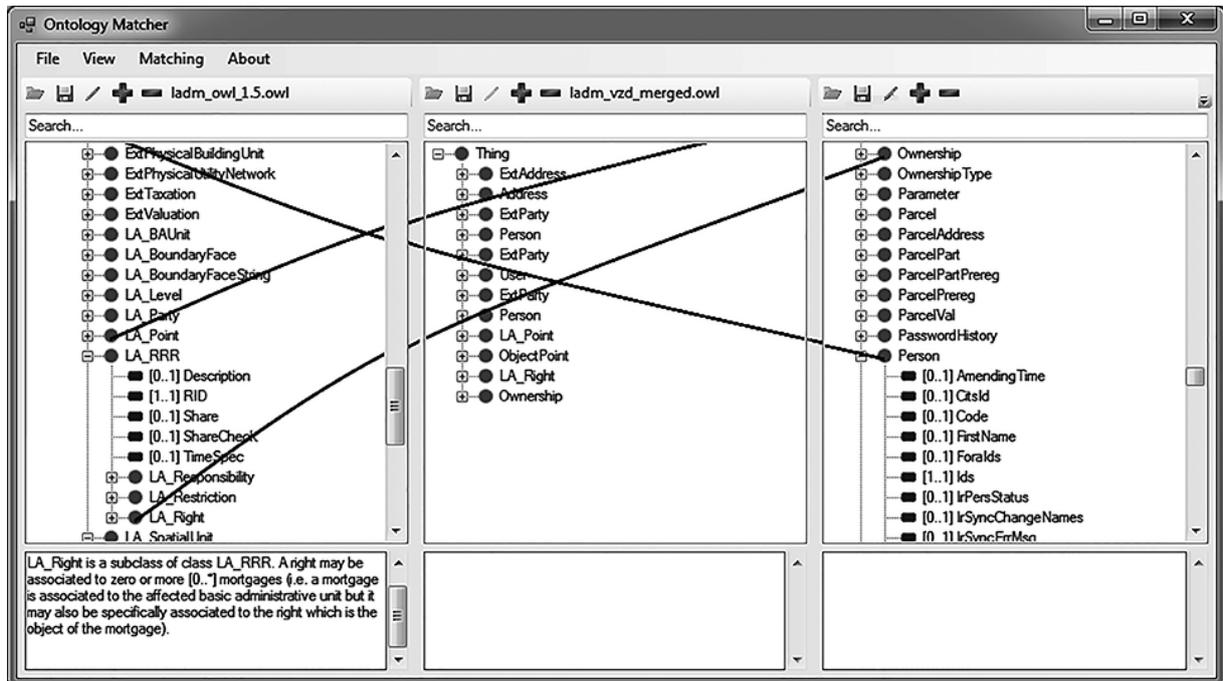


Fig. 13. Screenshot of ontology matching tool prototype

### VI MATCHING TOOL

The transformation process was carried out using a tool prototype specifically developed for this task. In Fig. 13, the screenshot of the tool can be seen with two ontologies opened and several concepts matched. The tool consists of three parts:

1. The first part is designed for automatic transformation of UML geospatial profile class diagrams into OWL ontologies [17], using similar approach to the one proposed in this paper.

2. The second part is designed for automatic transformation of relational database schemas into OWL ontologies, using the approach proposed in this paper.

3. The third part is designed for manual and automatic ontology matching. Ontology matcher consists of several name based ontology matching algorithms [15] that can be used automatically [18].

## VII RESULTS AND DISCUSSION

The transformation process employed to transform relational database data model, represented with ER diagrams, into an OWL ontology can be described with the following steps:

1. Relational database model is prepared for further processing as an XML Metadata Interchange (XMI) format file. XMI is an Object Management Group (OMG) standard for exchanging metadata information via Extensible Markup Language (XML).
2. Next, an XMI file is parsed by the transformation tool.
3. The transformation rules are applied and the ontology is generated.
4. The resulting ontology is then saved to OWL format file.
5. The OWL ontology is ready for visualization and further use by specially designed tools (e.g. Protégé).

The transformed ontology was put to the test using the ontology matching tool prototype. The tests consisted of automatically matching two ontologies using name based ontology matching algorithms. The matching results showed average similarity of 0.5. The full results were published in [18].

The solution presented in this paper can be useful to software engineering practitioners who are concerned with semantic web technology integration with legacy databases and to those who need to transform relational database data models to ontologies for ontology matching purposes.

## VIII CONCLUSIONS

The approach of relational database schema transformation into OWL ontology was presented in this paper. Some of the similarities and differences of ERD and OWL ontologies were analysed. The transformation rules defined. The prototype tool for automatic ERD transformation to OWL ontologies developed and the process of the transformation described.

The proposed approach allowed to successfully transform relational database components like tables, fields and relationships between tables to an ontology.

Automatic ontology creation from ER diagrams is the first step towards legacy database integration with semantic web technology.

Further work on this topic includes studying relational database denormalization techniques and applying them to the transformation process. This would help to improve transformed ontology hierarchical structure, which is lacking in the proposed

approach when transforming normalized relational databases.

## IX REFERENCES

- [1] W3C. "OWL 2 Web Ontology Language Document Overview (Second Edition)", 2012. [Online]. Available: <http://www.w3.org/TR/owl2-overview/>. [Accessed March 10, 2015].
- [2] D. E. Spanos, P. Starvrou and N. Mitrou, Bringing Relational Databases into the Semantic Web: A Survey, *Semantic Web*, vol 3, pp. 169-209, Apr. 2012.
- [3] L. Zhang and J. Li, Automatic Generation of Ontology Based on Database, *Journal of Computational Information Systems*, pp. 1148-1154, Apr. 2011.
- [4] W. Hu and Y. Qu, *Discovering Simple Mappings Between Relational Database Schemas and Ontologies, ISWC'07/ASWC'07 Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, pp. 225-238, 2007.
- [5] N. Alalwan, H. Zedan and F. Siewe, "Generating OWL Ontology for Database Integration", *Advances in Semantic Processing, 2009. SEMAPRO '09*, pp. 22-31, Oct. 2009.
- [6] M. Li, X. Y. Du and S. Wang, *Learning Ontology from Relational Database, Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference*, pp. 3410-3415, Aug. 2005.
- [7] S. Zhou, H. Ling, M. Han and H. Zhang, "Ontology Generator from Relational Database Based on Jena", *Computer and Information Science*, pp. 263-267, 2010.
- [8] Z. Telnarova, Relational database as a source of ontology creation, *Computer Science and Information Technology (IMCSIT), Proceedings of the 2010 International Multiconference*, pp. 135-139, Oct. 2010.
- [9] I. Astrova, N. Korda and A. Kalja, *Rule-Based Transformation of SQL Relational Databases to OWL Ontologies, In Proceedings of the 2nd International Conference on Metadata & Semantics Research*, 2007.
- [10] W3C. "A Direct Mapping of Relational Data to RDF", 2012. [Online]. Available: <http://www.w3.org/TR/rdb-direct-mapping/>. [Accessed May 11, 2015].
- [11] W3C. "R2RML: RDB to RDF Mapping Language", 2012. [Online]. Available: <http://www.w3.org/TR/r2rml/>. [Accessed May 11, 2015].
- [12] "The D2RQ Mapping Language", 2012. [Online]. Available: <http://d2rq.org/d2rq-language> [Accessed May 11, 2015].
- [13] F. Baadar, I. Horrocks and U. Sattler, "Description Logics as Ontology Languages for the Semantic Web", *Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence*, 2003.
- [14] R. Ghawi and N. Cullot, "Database-to-Ontology Mapping Generation for Semantic Interoperability", *VLDB '07*, September 23-28, 2007.
- [15] J. Euzenat and P. Shvaiko, *Ontology Matching, 2nd edition*. Springer-Verlag, Berlin Heidelberg (DE), 2013.
- [16] M. Fahad, "ER2OWL Generating OWL Ontology from ER Diagram", *Intelligent Information Processing IV*, pp. 28-37, 2008.
- [17] I. Zarembo, S. Kodors, *Automatic Transformation of UML Geospatial Profile to OWL Ontologies, Virtual Multidisciplinary Conference (QUAESTI 2013)*, ISBN: 978-80-554-0826-2, Slovak Republic, pp. 225-229, Dec. 2013.
- [18] I. Zarembo, A. Teilans, A. Rausis, J. Buls, "Assessment of Name Based Algorithms for Land Administration Ontology Matching," *ICTE in Regional Development*, vol. 43, pp. 53-61, Dec. 2014.