

A NOVEL METHOD FOR GENERATING AN E-LEARNING ONTOLOGY

Mohammed Reda Chbihi Louhdi¹, Hicham Behja² and Said Ouatik El Alaoui³

^{1,3}Laboratoire Informatique & Modélisation,

Faculty of sciences Dhar El Mehraz, Fez, Morocco

²Ecole Nationale Supérieure d'Electricité et de Mécanique, Casablanca, Morocco

ABSTRACT

The Semantic Web provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries. The existing web applications need to express semantics that can be extracted from users' navigation and content, in order to fulfill users' needs. E-learning has specific requirements that can be satisfied through the extraction of semantics from learning management systems (LMS) that use relational databases (RDB) as backend. In this paper, we propose transformation rules for building owl ontology from the RDB of the open source LMS Moodle. It allows transforming all possible cases in RDBs into ontological constructs. The proposed rules are enriched by analyzing stored data to detect disjointness and totalness constraints in hierarchies, and calculating the participation level of tables in n-ary relations. In addition, our technique is generic; hence it can be applied to any RDB.

KEYWORDS

Semantic Web, Ontology Building, E-learning, Learning Management System, Relational Databases, Schema Mapping, Data Analysis.

1. INTRODUCTION

Nowadays, the Web became a huge universe in perpetual expansion. It's consists of interlinked pages and web applications that contains a lot of videos, photos and interactive contents. Over time web technologies have evolved to give web developers the ability to create new generations of useful and immersive web experiences. Today's web is a result of the ongoing efforts of an open web community that helps define these web technologies, like HTML5, CSS3 and WebGL and ensure that they're supported in all web browsers.

Before 1999, experts call the Internet "Read-Only" web. The average internet user's role was limited to reading the information which was presented to him. The lack of active interaction of common users with the web leads to the birth of Web 2.0 or the "read-write" web. It has the ability to contribute content and interact with other web users. This interaction and contribution has dramatically changed the landscape of the web, and created billions of terabytes of data. To manage this data, it must be machine understandable. The web 3.0 or the "semantic web" provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries. One of its objectives is to convert the current web into a "web of data", by encouraging the inclusion of semantic content in web pages and documents. Besides, the Semantic Web aims at making information on the Web machine processable and understandable, and therefore, facilitates interoperability between applications [1].

Currently, most of the existing web applications in multiple domains (e-learning, e-government, e-commerce, etc.) create and manage data in structured or semi-structured way without expressing semantics. This semantic can help enhancing the structure and content of the web sites. It can also be used by recommendation systems to provide users with suggestions to meet their informational needs and preferences. Semantics can be extracted from two sources:

- The users ' navigation in order to better understand what users need,
- The content, which is generally structured in relational databases, in order to make it machine understandable,

Semantics are expressed through a technology called ontology, which is a formal and explicit specification of a shared conceptualization that refers to an abstract model of some phenomenon in the world that identifies the relevant concepts of that phenomenon [2]. Ontologies are particularly specified in languages that make possible the abstraction of data structures and allow strategies implementation. Semantic Web is then expected to provide languages that can both express data and rules for reasoning about the data, and also to export rules from any existing knowledge-representation system onto the web.

E-learning is just-in-time education integrated with high velocity value chains. It is the delivery of individualized, comprehensive, dynamic learning content in real time, aiding the development of communities of knowledge, linking learners and practitioners with experts [3]. The web 2.0 provides new tools (like forums and wikis) that enhances the e-learning and upgrades it to the so called "collaborative e-learning" or the "e-learning 2.0" [4]. Actually, new needs appear, the learning management systems (LMS) must provide a custom content to the learners, according to their preferences, experiences and needs. To reach this objective, semantics must be extracted from the LMS (user's navigation and content).

In this work we will focus on extracting semantics from the content of an open source LMS: "Moodle". It uses a relational database as a backend to manage its data. This database encapsulates conceptual data models and hides a strong semantics that could be exploited in the process of ontology extraction. Thus, we propose a method that uses schema mapping technique combined with a data analysis to generate an OWL ontology from a relational database. Our proposal covers all possible cases in databases and allows having richer ontologies.

The rest of this paper is organized as follows. Section 2 discusses related works in recommendation systems for Learning Management Systems, and the methods for extracting semantics using ontology engineering from relational databases. Section 3 describes the proposed transformation rules. Implementation in Moodle is presented in Section 4. Finally, Section 5 concludes this paper, and discusses the perspectives of this work.

2. RELATED WORKS

Due to the exponential increase in the amount of resources available and accessible on the web, Recommendation systems have seen their popularity grow in recent years. Combining techniques of information filtering, personalization, artificial intelligence, social networks and human-computer interaction, recommendation systems provide users with suggestions to meet their informational needs and preferences [5].

To produce recommendations, a number of approaches is possible: (1) the approach by content [6] which makes recommendations by comparing the semantic content of resources with the user's tastes, (2) the approach based on knowledge [7] that makes recommendations by exploiting knowledge about the user and pre-established heuristics, and (3) the approach by collaborative

filtering [8], which makes recommendations by analyzing, at the same time, the user's opinions and those of other users about the resources they have consulted.

Many works proposed to integrate recommendation systems in Learning Management Systems, in order to meet the specific needs of each learner. In [5] authors propose a Plugin that can be integrated on any e-Learning platform to keep tracks of the web searches made by learners, in order to use it for the recommendation. This Plugin also includes a new hybrid method to rank web documents before presenting them to the learner. In the paper [9], authors describe an automatic personalization approach aiming to provide online automatic recommendations for active learners without requiring their explicit feedback. Recommended learning resources are computed based on the current learner's recent navigation history, as well as exploiting similarities and dissimilarities among learners' preferences and educational content. A technique for developing a solution of personalized recommendations for e-learning systems is presented in [10]. It adopts an ontology-based modeling of user profiles and document models. The proposed recommendation technique is a hybrid one that involves two phases: (1) Collaborative filtering phase: user conceptual navigation is tracked in order to predict the next concept which will be focused by the user, according to his fingerprints and interests profile, (2) Content-based filtering: this concept is used in order to select the documents to be effectively recommended, in concordance with the user competences profile. In [11], the user-based collaborative filtering method is chosen as the primary recommendation algorithm, combined with online education. The authors analyze the requirement of a web-based e-learning recommendation system, and divide the system workflow into five sections: data collection, data ETL, model generation, strategy configuration, and service supply. Moreover, an architecture is proposed, based on which further development can be accomplished. In this architecture, there are seven modules, and four of them are core modules: recommendation models database, recommendation system database, recommendation management, data/model management.

Most of Learning Management Systems use relational databases as backend. Those databases contain strong semantics that can be exploited in recommendation systems and store important and useful information, which is a valuable source for ontology learning. The existing methods for ontology engineering from relational databases are based on three types of techniques [12], [13]:

- Reverse engineering: consists in recovering the conceptual data model from the physical schema of an existing database [14],
- Schema mapping: transforms the components of the conceptual data model or the physical model into ontology's concepts and relations [12],
- Data Mining [15]: exploits the records of the database in order to extract knowledge.

Recently, some methods [16], [17], [18], [19] proposed to combine schema mapping with reverse engineering [16], [18] or with Data Mining [17], [19]. Their objective is to extract more information from the database that will make richer the generated ontology. Methods that focus merely on transforming database schema generate inconsistent ontologies having a weak semantics. In other words, simple schema mapping leads to weak semantics and inconsistency and hence, will not be able to extract information enabling strong ontology building, especially in case of highly abstracted databases.

However, some existing methods [16], [18], [20], [21], [22] use the conceptual data model as a source of ontology learning, because it is semantically richer than the relational model. Unfortunately, in the most cases, the databases are available in a physical format (the corresponding conceptual data model is not available). In addition, the mapping operation (from the conceptual data model to the relational model) may create new tables and new attributes that

make difficult the ontology populating task (since there are many differences between the components of the conceptual data model and those of the relational model).

Schema mapping technique is used by ontology building methods. It converts the relational database schema (or the ER Model) to an ontology by using a set of predefined transformation rules. Some works [23], [24], [25], [26], [27] map ontologies to relational databases schemata in order to maintain interoperability between them. The schema mapping is performed using mapping rules that update the ontology when the database is modified and vice-versa.

Note that most methods based on the schema mapping technique cannot handle some complex cases like multiple inheritance, many-to-many relations with attributes, and the n-ary relations. The multiple inheritance case was treated by [22] where authors reproduce the hierarchy found in the conceptual data model in the taxonomy of the ontology. Concerning the many-to-many relations with attributes, they were supported by the transformation rules of some methods [20], [21], [22], [28]. The n-ary relation is a difficult case, because only binary relations between classes can be represented through object properties in the ontology. However, some works [20], [21], [28], [29], [30] propose solutions to represent n-ary relations in OWL ontologies. In [21], [28], [29], [30], authors create a class for the bridge table related by two object properties mutually inverse. The method [20] uses AllValuesFrom restrictions to link the class corresponding to the bridge table, with the classes that correspond to the participating tables to the n-ary relation. This solution is more representative than the first one, because the existence of a record in the bridge table is conditioned by the existence of records in tables that participate to the n-ary relation.

The foreign key columns (or one-to-many relations) and the simple inheritance cases are processed by the most of existing methods. Furthermore, the most existing methods transform the simple attributes into Data Type Properties. Some of them [21], [22], [28], [29], [30] suggest to add restrictions to the attributes that have a constraint (Primary Key, NOT NULL or UNIQUE). In the Table I, we present the main methods and the different cases treated from the relational model.

Table 1. The main methods and the different cases captured from the relational model.

Methods	One-to-many relation	Simple inheritance	Multiple inheritance	Many-to-many relation	Many-to-many relation with attributes	n-ary relations
[16], [17]	X	X				
[18], [19], [31]	X			X		
[20], [28]	X	X		X	X	X
[21]	X			X	X	X
[22]	X	X	X	X	X	
[29], [30]	X	X		X		X
[32], [33], [34], [35], [36]	X	X		X		
Our method	X	X	X	X	X	X

In this work, we propose to use the relational model as a source for ontology learning. Unlike the methods mentioned previously, we propose exhaustive transformation rules that deal with most of existing cases in databases (table1). Moreover, we analyze the database records to recover some disappeared aspects during the mapping from the conceptual data model to the relational model

(like disjointness and totalness in simple inheritance cases and the participating level of tables in n-ary relations).

3. TRANSFORMATION RULES

The process of transforming a relational database into an OWL ontology begin with classifying tables of the database schema into six categories according to their attributes. After that, we transform the tables of each category into ontological components by applying both mapping rules and data analysis. The latter finds disjointness and totalness in inheritance cases, and spots the participation level of tables in n-ary relations. The table 2 shows the proposed classification. In this work, we suppose that the databases are at least in the third normal form. Note that the first two categories contains only strong entities, hence the four others contains weak entities.

Table 2. The different categories adopted for classifying the database tables.

Category	Features
1	Tables containing only simple attributes without foreign keys constraint (Example : Tables <i>PERSON</i> and <i>PROGRAM</i> in Figure 1)
2	Tables containing at least one foreign key (Example: Table <i>ACTIVITY</i> in Figure 1).
3	Tables whose entire primary key is also a foreign key referencing a single table. (Example: Tables <i>STUDENT</i> and <i>TEACHER</i> in Figure 1).
4	Tables containing a composite primary key (two or more fields) which is also a foreign key whose fields are referencing exactly two tables (Example: Table <i>SUPERVISION</i> in Figure 1).
5	Tables containing a composite primary key (two or more fields) which is also a foreign key whose fields are referencing more than two tables. Simple attributes are not duplicated in any of the referenced tables (Example: Table <i>OFFERED_COURSE</i> in Figure 1).
6	Tables containing a composite primary key (two or more fields) which is also a foreign key whose fields are referencing more than two tables. Some simple attributes are duplicated in the referenced tables (Example: Table <i>PEDAGOGICAL_PROJECT</i> in Figure 1).

After classifying the database tables, we apply the appropriate transformation rules for each table's category (see table 2). In the rest of this paragraph, we will present the proposed transformation rules. All these rules are illustrated by examples using the database which is presented in Figure 1.

Rule 1: The tables that contain only simple columns (without foreign key constraint) are transformed into simple classes into the ontology (category 1). Example:

```
<owl:Class rdf:ID="PERSON"/>
<owl:Class rdf:ID="PROGRAM"/>
```

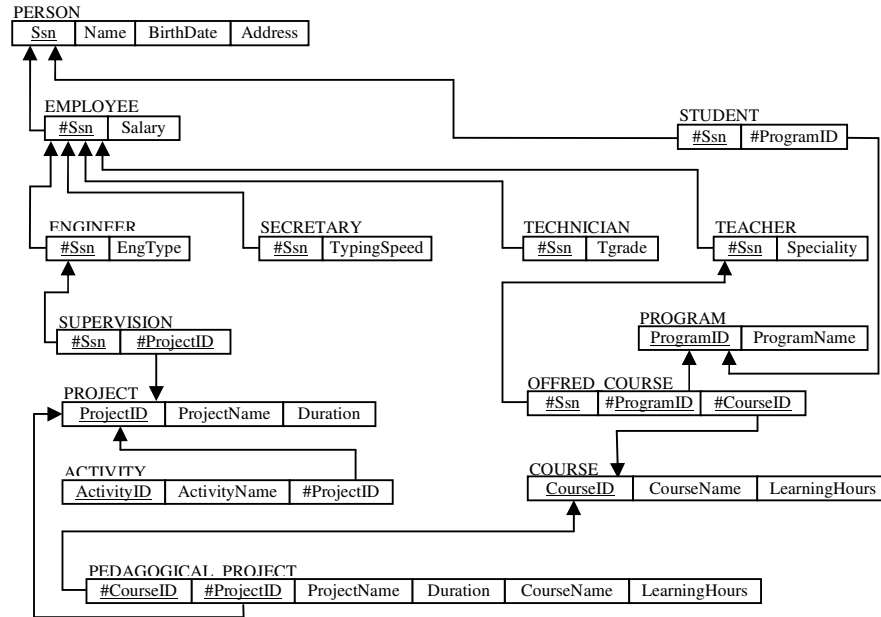


Figure 1. Example of a normalized database

Rule 2: Tables of the second category are transformed into simple classes in the ontology. Each foreign key is mapped into two Object-Properties (mutually inverse). The first one has the class corresponding to current table as domain, and its range is the referenced table by the foreign key. The second one (inverse of the first Object-Property) is declared as inverse functional. Example:

```
<owl:Class rdf:ID="ACTIVITY"/>
<owl:ObjectProperty rdf:ID="activityHasProject">
  <rdfs:domain rdf:resource="#ACTIVITY" />
  <rdfs:range rdf:resource="#PROJECT" />
</owl:ObjectProperty>
<owl:InverseFunctionalProperty rdf:ID="project'sActivity">
  <rdfs:domain rdf:resource="#PROJECT" />
  <rdfs:range rdf:resource="#ACTIVITY"/>
  <owl:inverseOf rdf:resource="#activityHasProject" />
</owl:InverseFunctionalProperty>
```

Rule 3: we can identify simple inheritance relationships from tables of the third category. All tables in this category are sub-tables in hierarchies. Each sub-table is transformed into a class in the ontology and is declared as a subclass of the table referenced by the foreign key (which is also the primary key of each sub-table). Example:

```
<owl:Class rdf:ID="PERSON" />
<owl:Class rdf:ID="STUDENT">
  <rdfs:subClassOf rdf:resource="#PERSON" />
</owl:Class>
<owl:Class rdf:ID="TEACHER"/>
  <rdfs:subClassOf rdf:resource="#PERSON" />
</owl:Class>
```

After reproducing simple inheritance relations into the taxonomy of the ontology, we will identify disjointness and totalness constraints in those relations.

In a simple inheritance relation, disjointness means that an entity can be a member of at most one of the subclasses (that have the same level) of a hierarchy [37]. To identify the existence of disjointness between tables having the same level in a hierarchy, we propose the algorithm presented in Figure 2.

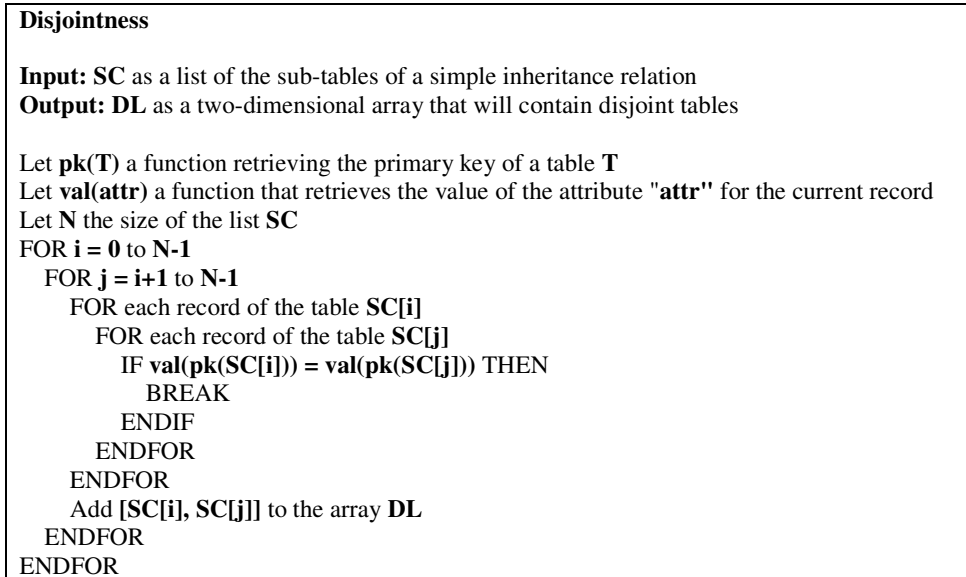


Figure 2. Disjointness detection algorithm

The following OWL code illustrates the disjointness constraint in a simple inheritance relation:

```
<owl:Class rdf:about="#TECHNICIAN">
  <rdfs:subClassOf rdf:resource="#EMPLOYEE"/>
  <owl:disjointWith rdf:resource="#SECRETARY"/>
  <owl:disjointWith rdf:resource="#TEACHER"/>
</owl:Class>
<owl:Class rdf:about="#SECRETARY">
  <rdfs:subClassOf rdf:resource="#EMPLOYEE"/>
  <owl:disjointWith rdf:resource="#TEACHER"/>
</owl:Class>
<owl:Class rdf:about="#TEACHER">
  <rdfs:subClassOf rdf:resource="#EMPLOYEE"/>
</owl:Class>
```

Concerning the totalness, it specifies that every entity in the superclass must be a member of at least one subclass in the hierarchy [37]. To identify the existence of totalness in a hierarchy, we propose the algorithm presented in Figure 3. The following OWL code illustrates the totalness constraint in a simple inheritance relation:

```
<owl:class rdf:ID="EMPLOYEE">
  <owl:unionOf rdf:parseType="Collection">
    <owl:class rdf:about="#SECRETARY" />
    <owl:class rdf:about="#TEACHER" />
    <owl:class rdf:about="#TECHNICIAN" />
  </owl:unionOf>
</owl:class>
```

```

Totalness

Input: - ST : the super-table
         - ssT : a list of the sub-tables of ST
Output: F as Boolean to flag the existence of totalness

Let pk(T) a function retrieving the primary key of a table T
Let val(attr) a function that retrieves the value of the attribute "attr" for the current record
Let N the size of the list ssT
F ← FALSE
FOR each table T of ssT
  FOR each record of ST
    FOR each record of T
      IF val(pk(T)) = val(pk(ST)) THEN
        N ← N - 1
        BREAK
      ENDIF
    ENDFOR
  ENDFOR
ENDFOR
IF N = 0 THEN
  F ← TRUE
ENDIF

```

Figure 3. Totalness detection algorithm

In the case of existence of both a disjointness and totalness in a simple inheritance relationship, we combine the two previous proposals. The following example illustrates this situation:

```

<owl:Class rdf:about="#TECHNICIAN">
  <rdfs:subClassOf rdf:resource="#EMPLOYEE"/>
  <owl:disjointWith rdf:resource="#SECRETARY"/>
  <owl:disjointWith rdf:resource="#TEACHER"/>
</owl:Class>
<owl:Class rdf:about="#SECRETARY">
  <rdfs:subClassOf rdf:resource="#EMPLOYEE"/>
  <owl:disjointWith rdf:resource="#TEACHER"/>
</owl:Class>
<owl:Class rdf:about="#TEACHER">
  <rdfs:subClassOf rdf:resource="#EMPLOYEE"/>
</owl:Class>
<owl:class rdf:ID="EMPLOYEE">
  <owl:unionOf rdf :parseType="Collection">
    <owl:class rdf:about="#SECRETARY" />
    <owl:class rdf:about="#TEACHER" />
    <owl:class rdf:about="#TECHNICIAN" />
  </owl:unionOf>
</owl:class>

```

Rule 4: The tables containing a composite primary key (two or more columns) which is also a foreign key whose fields are referencing exactly two tables (category 4), are mapped into two Object-Properties mutually inverse. Example:

```

<owl:ObjectProperty rdf:ID="hasProject">
  <rdfs:domain rdf:resource="#ENGINEER" />
  <rdfs:range rdf:resource="#PROJECT" />
</owl:ObjectProperty>

```



```

<owl:ObjectProperty rdf:ID="hasEngineer">
  <rdfs:domain rdf:resource="#PROJECT" />
  <rdfs:range rdf:resource="#ENGINEER" />
  <owl:inverseOf rdf:resource="#hasProject"/>
</owl:ObjectProperty>

```

If a table of category 4 contains simple columns (table resulted from a many-to-many relation with attributes), we apply rules 4 and 5 to this table.

Rule 5: the tables of the category 5 are resulting from n-ary relations. Their primary keys are composed by several foreign keys (more than two) referencing the participating tables to the relation.

To represent n-ary relations from the relational model, we create a class corresponding to the bridge table related to the classes that correspond to the participating tables to the n-ary relation by OWL restrictions (*allValuesFrom* or *someValuesFrom*). These restrictions are depending on the participation level of tables in the relation. In Figure 6, we present an example of transforming the n-ary relation *OFFRED_COURSE* in the example given in the Figure 1.

To define the participation level of each table to the n-ary relation, we check if all records of the participating tables are referenced in the bridge table. If so, we use an *allValuesFrom* restriction, else a *someValuesFrom* restriction is used.

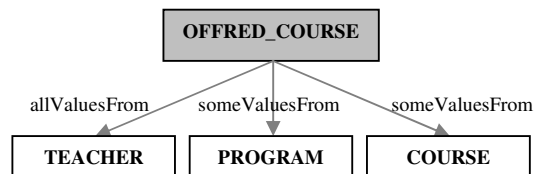


Figure 6. Example of transforming a n-ary relation

To illustrate this solution, we present below the OWL code corresponding to the n-ary relation *OFFRED_COURSE*:

```

<owl:Class rdf:ID="OFFRED_COURSE">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom>
        <owl:Class rdf:about="#COURSE"/>
      </owl:someValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="hasCourse"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom>
        <owl:Class rdf:about="#PROGRAM"/>
      </owl:someValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="hasProgram"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

```

    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:allValuesFrom>
          <owl:Class rdf:about="#TEACHER"/>
        </owl:allValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="hasTeacher"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

```

Rule 6: in the conceptual data model, the tables of the category 6 are subclasses in more than one class/subclass relationship (multiple inheritance). In the relational model, these tables can be confused with the bridge tables of n-ary relations (category 5), since they are weak entities whose primary key consists of two or more foreign keys referencing two or more tables. To distinguish between both categories (5 and 6), we supposed that tables of category 6 must contain in addition to the primary key, inherited attributes (during the mapping process) belonging to super-tables.

To map the multiple inheritance case, we reproduce the same hierarchy in the taxonomy of the ontology. Each sub-table (of category 6) is transformed into a subclass of the classes corresponding to the tables referenced by the foreign keys of the sub-table. For example, the OWL code for the *PEDAGOGICAL_PROJECT* (Figure 1) is as follows:

```

<owl:Class rdf:ID="COURSE" />
<owl:Class rdf:ID="PROJECT"/>
<owl:Class rdf:ID="PEDAGOGICAL_PROJECT">
  <rdfs:subClassOf rdf:resource="#PROJECT" />
  <rdfs:subClassOf rdf:resource="#COURSE" />
</owl:Class>

```

Rule 7: Concerning the transformation of the columns (without foreign key constraint), we create for each attribute a datatype property for which the domain is the class corresponding to the table containing this column and the range is the type in XML schema. Example:

```

<owl:DatatypeProperty rdf:ID="Name">
  <rdfs:domain rdf:resource="#PERSON"/>
  <rdf:range rdf:resource="&xsd:String"/>
</owl:DatatypeProperty>

```

For attributes with special constraints such as NOT NULL, UNIQUE and Primary Key, we propose to treat them as follows:

NOT NULL: add the MinCardinality restriction to the Datatype Property with the value 1,

```

<owl:Class rdf:ID="PERSON">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Name" />
      <owl:minCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

UNIQUE: declare the Datatype property as inverse functional,

```
<owl:InverseFunctionalProperty rdf:ID="Program_name">
  <rdfs:domain rdf:resource="#PROGRAM"/>
  <rdf:range rdf:resource="&xsd:String"/>
</owl:InverseFunctionalProperty>
```

Primary Key: add the MinCardinality restriction to the Datatype Property with the value 1, and declare it as inverse functional,

```
<owl:InverseFunctionalProperty rdf:ID="Ssn">
  <rdfs:domain rdf:resource="#PERSON"/>
  <rdf:range rdf:resource="&xsd:String"/>
</owl:InverseFunctionalProperty>
<owl:Class rdf:ID="PERSON">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Ssn" />
      <owl:minCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

The table 3 summarizes the proposed transformation rules.

Table 3. Summary of the proposed transformation rules.

Rule	Case	OWL Component
1	Strong entity	Class
2	Foreign Key	Two Object Properties mutually inverse 1. The First one : - Domain: Class corresponding to the table containing the column, - Range : Class corresponding to the referenced table, 2. The Second one is the inverse of the first, and it is declared as inverse functional
3 & 6	Simple and Multiple inheritance	Reproducing inheritance relations into the taxonomy of the ontology
4	Many-to-Many relation	Two Object Properties mutually inverse that rely the classes corresponding to the participant tables of the relation
5	N-ary Relation	A class corresponding to bridge table, related to the classes that correspond to the participating tables with "is-a" relation restricted with the OWL restriction "AllValuesFrom" or "SomeValuesFrom" according to the participating level of each table of the relation
4 & 5	Many-to-Many relation with attributes	Combination of the above two cases
7	Simple column	Data Type Property - Domain : Class corresponding to the table containing the column, - Range : The column type expressed with XML Schema,
	Column with UNIQUE constraint	Inverse Functional Property - Domain : Class corresponding to the table containing the column, - Range: The column type expressed with XML Schema,

	Column with NOT NULL constraint	Data Type Property - Domain : Class corresponding to the table containing the column, - Range : The column type expressed with XML Schema, - Minimal cardinality = 1
	Primary Key	Inverse Functional Property - Domain : Class corresponding to the table containing the column, - Range : The column type expressed with XML Schema, - Minimal cardinality = 1

4. IMPLEMENTATION IN MOODLE LMS

To evaluate the efficiency of the proposed transformation rules, we implemented the proposal with Java and the Jena API (Java framework for building Semantic Web applications) for automatically OWL ontology building from relational database. Having a friendly-user interface, our system allows building OWL file that contains the definition of the extracted ontology from a MySql database.

We use the database of the Learning Management System of the Cadi Ayyad University (CAU) of Marrakech – Morocco (<http://minfo.uca.ma/>), which is based on the open source LMS Moodle. This system is used to provide computer sciences courses for students of the bachelor program. Moreover, collaborative tools (forums, wikis and blogs) are enabled in order to allow learners and authors to discuss and collaborate.

The Moodle database isn't in normalized form. All the primary keys of all tables have the same names: "id". After exporting the database of the LMS of CAU, all foreign keys constraints disappear. To recover them, we use the database creation files (in XML format) that contain the metadata of all tables.

The exported database contains 198 tables (for the version 1.9.4 of Moodle). Otherwise, the pedagogical platform of the CAU (in production since 2009) contains 107 courses and more than 39000 users (until July 2013). Some statistics about the stored data are presented in the table 4, and the metadata of the database structure are presented in the table 5.

Table 4. Some statistics about the pedagogical platform of the CAU.

Category	Information	Number
Users	Administrator	5
	Course Responsible	2
	Teacher Editor	73
	Teacher non Editor	157
	Learner	38874
	<i>All users</i>	<i>39111</i>
Forums	Discussions	354
	Topics	224
	Posts	4654
Other collaboration tools	Wikis	4
	Messages échangés en CHAT	3373

Table 5. Metadata of Moodle database of the pedagogical platform of the CAU.

Metadata		Count
Entities	Strong	198
	Weak	0
Columns	Primary Key (PK) constraint	198
	Foreign Key (FK) constraint	166
	PK and FK constraints	0
	Others	1431

Using the proposed transformation rules to map Moodle database into ontology, the obtained numbers of concepts, Object Properties and Data Type Properties are respectively 198, 332 and 1629. All the existing cases in the Moodle database were successfully transformed into ontological components. The resulted OWL file is very big (more than 14000 lines). Due to the number of pages limit, we will present a small part of the ontology that concerns a part of the Moodle question's module. The relational model, the ontology (presenting the Object Properties relations between concepts) and the OWL code (containing the concepts, Object Properties and some Data Type Properties) of the question's module are presented respectively in the figures 7, 8 and 9.

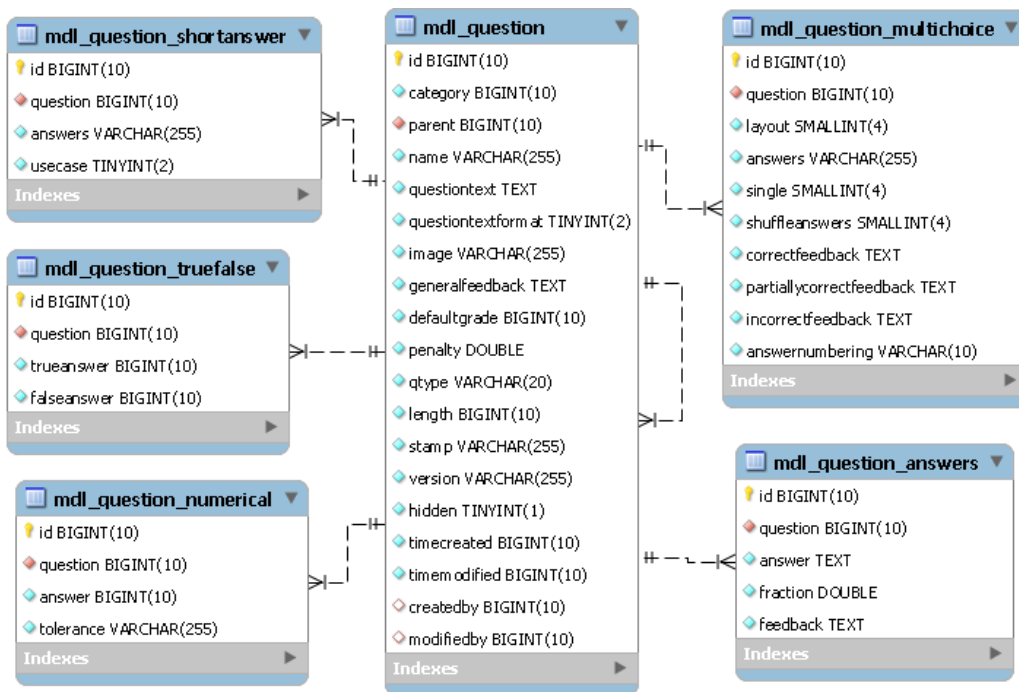


Figure 7. The relational model of a sub-part of the Question's module in the Moodle Database

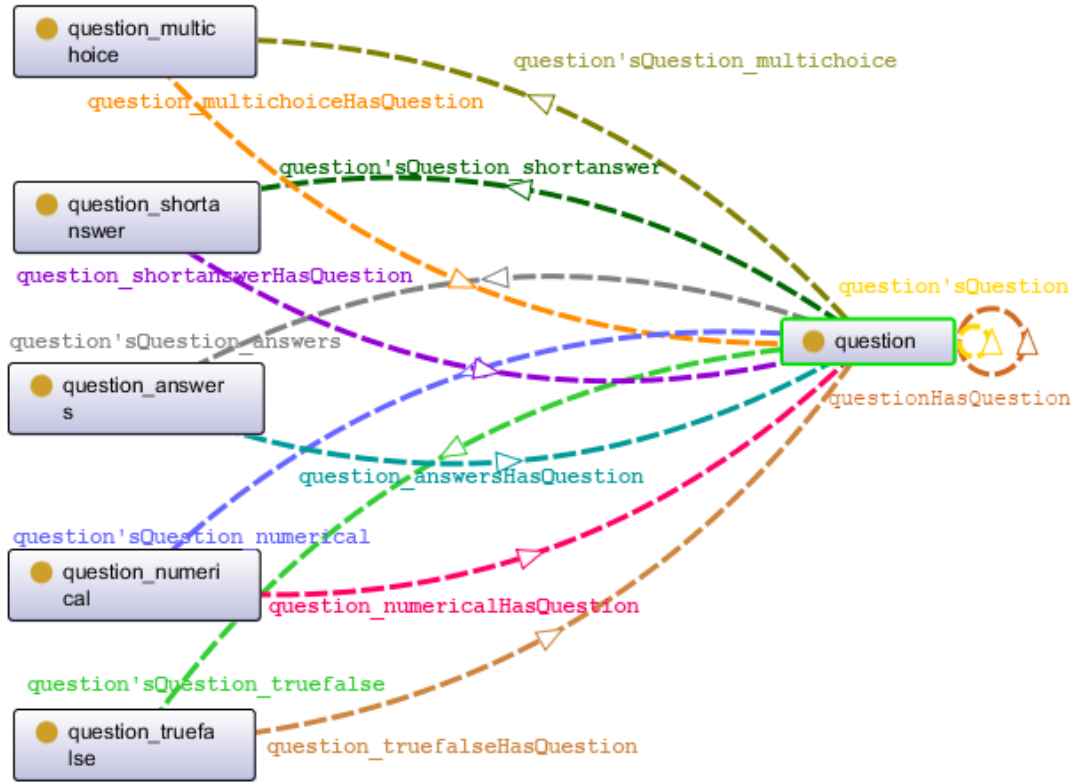


Figure 8. The ontological model of a sub-part of the Question's module in the Moodle Database

```

<owl:Class rdf:about="question_numerical"/>
<owl:Class rdf:about="question_multichoice"/>
<owl:Class rdf:about="question_answers"/>
<owl:Class rdf:about="question_truefalse"/>
<owl:Class rdf:about="question_shortanswer"/>
<owl:Class rdf:about="question"/>
<owl:ObjectProperty rdf:about="question_multichoiceHasquestion">
  <rdfs:range rdf:resource="question"/>
  <rdfs:domain rdf:resource="question_multichoice"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="questionHasQuestion">
  <rdfs:range rdf:resource="question"/>
  <rdfs:domain rdf:resource="question"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="question_truefalseHasQuestion">
  <rdfs:range rdf:resource="question"/>
  <rdfs:domain rdf:resource="question_truefalse"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="question_answersHasQuestion">
  <rdfs:range rdf:resource="question"/>
  <rdfs:domain rdf:resource="question_answers"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="question_shortanswerHasQuestion">
  <rdfs:range rdf:resource="question"/>
  <rdfs:domain rdf:resource="question_shortanswer"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="question_numericalHasQuestion">
  <rdfs:range rdf:resource="question"/>
  <rdfs:domain rdf:resource="question_numerical"/>
</owl:ObjectProperty>
<owl:InverseFunctionalProperty rdf:about="question'sQuestion_multichoice">

```

```

<owl:inverseOf rdf:resource="question_multichoiceHasQuestion"/>
<rdfs:domain rdf:resource="question"/>
<rdfs:range rdf:resource="question_multichoice"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="question'sQuestion">
<owl:inverseOf rdf:resource="questionHasQuestion"/>
<rdfs:domain rdf:resource="question"/>
<rdfs:range rdf:resource="question"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="question'sQuestion_truefalse">
<owl:inverseOf rdf:resource="question_truefalseHasQuestion"/>
<rdfs:domain rdf:resource="question"/>
<rdfs:range rdf:resource="question_truefalse"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="question'sQuestion_answers">
<owl:inverseOf rdf:resource="question_answersHasQuestion"/>
<rdfs:domain rdf:resource="question"/>
<rdfs:range rdf:resource="question_answers"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="question'sQuestion_shortanswer">
<owl:inverseOf rdf:resource="question_shortanswerHasQuestion"/>
<rdfs:domain rdf:resource="question"/>
<rdfs:range rdf:resource="question_shortanswer"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="question'sQuestion_numerical">
<owl:inverseOf rdf:resource="question_numericalHasQuestion"/>
<rdfs:domain rdf:resource="question"/>
<rdfs:range rdf:resource="question_numerical"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="id_question_truefalse">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
<rdfs:domain rdf:resource="question_truefalse"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="id_question_multichoice">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
<rdfs:domain rdf:resource="question_multichoice"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="id_question_answers">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
<rdfs:domain rdf:resource="question_answers"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="id_questionS_numerical">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
<rdfs:domain rdf:resource="question_numerical"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="id_question_shortanswer">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
<rdfs:domain rdf:resource="question_shortanswer"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:InverseFunctionalProperty>
<owl:InverseFunctionalProperty rdf:about="id_question">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
<rdfs:domain rdf:resource="question"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:InverseFunctionalProperty>

```

```

<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="id_question_truefalse"/>
</owl:Restriction>
<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="id_question_multichoice"/>
</owl:Restriction>
<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="id_question_answers"/>
</owl:Restriction>
<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="id_question_numerical"/>
</owl:Restriction>
<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="id_question_shortanswer"/>
</owl:Restriction>
<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="id_question"/>
</owl:Restriction>
<owl:DatatypeProperty rdf:about="trueanswer_question_truefalse">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
  <rdfs:domain rdf:resource="question_truefalse"/>
</owl:DatatypeProperty>
<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="question_truefalse"/>
</owl:Restriction>
<owl:DatatypeProperty rdf:about="createdby_question">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
  <rdfs:domain rdf:resource="question"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="questiontext_question">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="question"/>
</owl:DatatypeProperty>
<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="questiontext_question"/>
</owl:Restriction>
<owl:DatatypeProperty rdf:about="usecase_question_shortanswer">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#short"/>
  <rdfs:domain rdf:resource="question_shortanswer"/>
</owl:DatatypeProperty>
<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="usecase_question_shortanswer"/>
</owl:Restriction>
<owl:DatatypeProperty rdf:about="answer_question_answers">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="question_answers"/>
</owl:DatatypeProperty>

```



```

<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="answer_question_answers"/>
</owl:Restriction>
<owl:DatatypeProperty rdf:about="tolerance_question_numerical">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="question_numerical"/>
</owl:DatatypeProperty>
<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="tolerance_question_numerical"/>
</owl:Restriction>
<owl:DatatypeProperty rdf:about="single_question_multichoice">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#short"/>
  <rdfs:domain rdf:resource="question_multichoice"/>
</owl:DatatypeProperty>
<owl:Restriction>
  <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
  <owl:onProperty rdf:resource="single_question_multichoice"/>
</owl:Restriction>

```

Figure 9. The OWL code of a sub-part of the Question's module in the Moodle Database

5. CONCLUSION AND FURTHER WORKS

In this paper, we have proposed a novel method for generating an e-Learning ontology using a technique which consists in a set of transformation rules for building OWL ontologies from relational databases. The schema mapping uses the transformation rules to transform the components of the physical model into ontology's components. The data analysis is used to recover some disappeared aspects during mapping conceptual data model to the relational model (like disjointness and totalness in simple inheritance cases and the participating level of tables in n-ary relations).

We have implemented our proposal using the Moodle database of the pedagogical platform of the CA University. The obtained results are satisfactory compared to other methods in terms of the number of the treated concepts. Moreover, our method covers all possible cases in databases. The generated ontology has richer non-taxonomic relations.

A major direction for improvement could be to add a reverse engineering phase before applying the transformation rules in order to detect generalization and specialization inheritance cases. This process will allow us to recover disappeared tables during mapping conceptual data model to the relational model. Furthermore, integrating a reverse engineering phase will make the built ontologies richer in terms of taxonomic relations. Finally, we also suggest refine the obtained ontology by renaming its components in order to make them more expressive.

ACKNOWLEDGMENTS

We want to thank Misters *Driss Goujdami* and *Abdelali Rochdi* for providing us a dump file of the database of the pedagogical platform of the *Cadi Ayyad University*. We also want to thank Mister *My. Mehdi Bouhamidi* for his technical support to deploy the dump file in another database server.

REFERENCES

- [1] W3C, the semantic web activity, Available at : <http://www.w3.org/2001/sw/>
- [2] T. Gruber, Ontology, In Ling Liu & M. Tamer Özsu (Ed.), the Encyclopedia of Database Systems (Springer-Verlag, 2009, 1963-1965).
- [3] Drucker, P. (2000). Need to Know: Integrating e- Learning with High Velocity Value Chains. A Delphi Group White Paper. Available at : <http://www.delphigroup.com/whitepapers/pdf/20001213-e-learning-wp.pdf>
- [4] Stephen Downes, E-learning 2.0 eLearn Lagazine, October 2005. Available at: <http://elearnmag.acm.org/featured.cfm?aid=1104968>
- [5] H. Moutachauik, H. Douzi, A. Marzak, H. Behja, B. Ouhbi, (2012) “ Plugin of Recommendation based on a hybrid method for the ranking of documents in the e-learning platforms”, 5th International Conference, ICISP 2012, pp 587-595.
- [6] M. Pazzani & D. Billsus. (2007) The Adaptive Web, chapter Content-Based Recommendation Systems, pages 325–341. Springer Berlin / Heidelberg.
- [7] R. Burke, K. Hammond, and E. Cooper. (1996) “Knowledge based navigation of complex information spaces”, In Proc. of the 13th National Conference on Artificial Intelligence (AAAI’96), pages 462–468.
- [8] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. (1992) “Using Collaborative filtering to weave an information tapestry”, Communications of the ACM, Vol. 35, No. 12, pp. 61–70.
- [9] Khribi, M. K., Jemni, M., & Nasraoui, O. (2009). “Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval”, Educational Technology & Society, Vol. 12, No. 4, pp. 30–42.
- [10] Brut, M.; Sèdes, F., (2010) “Ontology-Based Solution for Personalized Recommendations in E-Learning Systems. Methodological Aspects and Evaluation Criterias”, Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on, pp.469-471.
- [11] Huiyi Tan; Junfei Guo; Yong Li, (2008) “E-learning Recommendation System”, Computer Science and Software Engineering, 2008 International Conference on , vol.5, pp.430,433.
- [12] Alexander Maedche & Steffen Staab, (2005) “Ontology Learning for The Semantic Web”, IEEE Intelligent Systems, Vol. 16, No. 2, pp. 72-79.
- [13] A. Gomez-Perez, D. Manzano-Macho, A Survey of Ontology Learning Methods and Techniques, Deliverable, 1.5. OntoWeb Consortium, (2003). Available from: <http://www.sti-innsbruck.at/fileadmin/documents/deliverables/Ontoweb/D1.5.pdf>, last visited: 03.15.2010
- [14] R. Chiang, T. Barron, V. Storey, Reverse engineering of relational databases: Extraction of an EER model from a relational database, Data & Knowledge Engineering, Vol. 12, n. 2, pp. 107–142, 1994
- [15] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From Data Mining to Knowledge Discovery in Databases, AI Magazine, Vol. 17, pp. 37-54, 1996
- [16] Chen He-ping, He Lu & Chen Bin, (2008) “Research and implementation of ontology automatic construction based on relational database”, Proceedings of the International Conference on Computer Science and Software Engineering, pp. 1078-1081.
- [17] Heru Agus, Su-Cheng Haw & Ziyad.T. Abdul-Mehdi, (2011) “Ontology extraction from relational database: Concept hierarchy as background knowledge”, Knowledge-Based Systems, Vol. 24, No 3, pp. 457-464,
- [18] Justas Trinkunas & Olegas Vasilecas, (2007) “Building ontologies from relational databases using reverse engineering methods”, Proceedings of the international conference on Computer systems and technologies.
- [19] Farid Cerbah, (2008) “Mining the Content of Relational Databases to Learn Ontologies with Deeper Taxonomies”, Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 553- 557.
- [20] Zhuoming Xu, Xiao Cao, Yisheng Dong & Wenping Su, (2004) “Formal Approach and Automated Tool for Translating ER Schemata into OWL Ontologies”, Advances in Knowledge Discovery and Data Mining, pp. 464-475.
- [21] Igor Myroshnichenko & Marguerite C. Murphy, (2009) “Mapping ER Schemas to OWL Ontologies”, Proceedings of the IEEE International Conference on Semantic Computing, pp. 324-329.
- [22] Sujatha R. Upadhyaya & P. Sreenivasa Kumar, (2005) “ERONTO: A Tool for Extracting Ontologies from Extended ER Diagrams”, Proceedings of the 20th ACM Symposium on Applied Computing, pp. 666-670.

- [23] Shihan Yang & Jinzhao Wu, (2010) "Mapping Relational Databases into Ontologies through a Graph-based Formal Model", Proceedings of the Sixth International Conference on Semantics Knowledge and Grid, pp. 219-226.
- [24] Christian Bizer, (2003) "D2R MAP - a database to RDF mapping language", Proceedings of the 12th International World Wide Web Conference.
- [25] Nikolaos Konstantinou , Dimitrios-emmanuel Spanos , Michael Chalas , Emmanuel Solidakis & Nikolas Mitrou, (2006) "VisAVis: An Approach to an Intermediate Layer between Ontologies and Relational Database Contents", Proceedings of Workshops and Doctoral Consortium, The 18th International Conference on Advanced Information Systems Engineering - Trusted Information Systems.
- [26] Shihan Yang, Ying Zheng & Xuehui Yang, (2010) "Semi-automatically building ontologies from relational databases", Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology, pp. 150 - 154.
- [27] Jesús Barrasa , Óscar Corcho & Asunción Gómez-pérez, (2004) "R2O, an Extensible and Semantically based Database-to-Ontology Mapping Language", Proceedings of the 2nd Workshop on Semantic Web and Databases, pp. 1069-1070.
- [28] Irina Astrova , Nahum Korda & Ahto Kalja, (2007) "Rule-Based Transformation of SQL Relational Databases to OWL Ontologies", Proceedings of the 2nd International Conference on Metadata & Semantics Research, pp. 415-424.
- [29] Man Li, Xiao-Yong Du & Shan Wang, (2005) "Learning ontology from relational database", Proceedings of the International Conference on Machine Learning and Cybernetics, pp. 3410-3415.
- [30] Zdenka Telnarova, (2010) "Relational database as a source of ontology creation", Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 135-139.
- [31] Kobra etminani , Mohsen Kahani & Yanehsari N.R, (2009) "Building ontologies from relational databases", Proceedings of the First International Conference on Networked Digital Technologies, pp. 555-557.
- [32] Changjun Hu, Huayu Li, Xiaoming Zhang & Chongchong Zhao, (2008) "Research and Implementation of Domain-Specific Ontology Building from Relational Database", Proceedings of the Third ChinaGrid Annual Conference, pp. 289 - 293.
- [33] Ahmed Waqas, Aslam Muhammad Ahtisham, Shen Jun & Yong Jianming, (2011) "A light weight approach for ontology generation and change synchronization between ontologies and source relational databases", Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design, pp. 208-214.
- [34] Nadine Cullot, Raji Ghawi & Kokou Yétongnon, (2007) "DB2OWL: A Tool for Automatic Database-to-Ontology Mapping", Proceedings of the 15th Italian Symposium on Advanced Database Systems, pp. 491-494.
- [35] Irina Astrova & Ahto Kalja, (2006) "Towards the Semantic Web: Extracting OWL Ontologies from SQL Relational Schemata", Proceedings of the 2nd International Conference on Metadata & Semantics Research, pp. 62-66.
- [36] Guohua Shen, Zhiqiu Huang, Xiaodong Zhu & Xiaofei Zhao, (2006) "Research on the Rules of Mapping from Relational Model to OWL", Proceedings of the OWLED*06 Workshop on OWL: Experiences and Directions.
- [37] Ramez Elmasri & Shamkant Navathe, (2011) Fundamentals of Database Systems sixth Edition Pearson Education.