Research on Learning OWL Ontology from Relational Database

Chun-hua Liao¹, Ya-fen Wu², Guo-hua King¹

¹Computer School of Jiangxi University of Traditional Chinese Medicine, Nanchang, China;
²Research Department, Jiangxi University of Traditional Chinese Medicine, Nanchang, China;

*Corresponding author e-mail: lch0422@163.com

Abstract. Ontology learning is the pivotal technology to construct knowledge base efficiently, and the data of existing knowledge systems are mostly organized in relational mode. In order to reuse and share knowledge in existing knowledge systems, it is necessary to construct ontology for existing knowledge systems. After comparing and analyzing the formal definitions of relational data schema and ontology, this paper proposes a method of extracting knowledge ontology from relational data model. This method builds OWL ontology knowledge base architecture based on metadata of relational data schema, maps different data elements into components of OWL ontology, and implements with prototype tool based on Java platform. As a case study, this method is applied to transform the database of online examination system into OWL ontology model. The method proposed in this paper can be applied to the practice of transferring the widely existing relational-based knowledge base system to the knowledge base system based on OWL ontology framework.

1. Introduction

When constructing the knowledge base, the ontology technology is often introduced. By formalizing the concepts and their relationships, it ensures that knowledge understanding is unique and accurate in the process of transferring domain knowledge, and improves the sharing and reuse of knowledge ontology. Ontology construction techniques include manual methods and ontology learning methods. Tools such as WebOnto [2], Protégé[3], Onto Edit [4], and KAON [5] provide a friendly graphical interface and consistency checking mechanism for building ontologies manually. The advantage of these tools is that developers don't need to know the details of ontology description language and, avoid the occurrence of grammatical errors, thus concentrating on the organization of ontology content. The disadvantage is that ontology developers still need to input and edit the names, constraints, attributes of each concept one by one. For example, Cyc [6] and Mikosmos [7] depend on manual input of a large amount of knowledge, and then can infer or acquire new knowledge based on these knowledge. Therefore, manual ontology development is still a time-consuming and error-prone task. Building ontology has become a bottleneck in knowledge engineering. Therefore, Ontology Learning has become a hotspot in the research of ontology construction method, which is an automatic or semi-automatic method.
2. Related research
Ontology learning, also known as ontology mining, uses machine learning and statistical techniques to automatically or semi-automatically acquire ontology from existing data sources. According to the form of data resources, ontology learning can be divided into three methods based on structured data, semi-structured data and unstructured data. The research of ontology learning technology based on unstructured data mainly focuses on obtaining ontology from pure text. In the learning process, the NLP technology is applied to preprocess the ontology [8], and then use statistics, machine learning [9, 10] to acquire knowledge from it, or mix linguistic and statistical techniques. With the increase of network resources, semi-structured data (XML/HTML web pages, DTD, RDF annotated web pages) have become the data source of ontology learning. Such as data mining methods are used to extract the classification relationships of concepts from the data warehouse described in XML / RDF [11]. And ontoBuilder [12] provides the function of helping users to generate cost bodies from semi-structured data sources marked by XML and HTML.

Most current knowledge sources are dynamic Web content supported by relational databases. At the same time, all existing information systems store data in the form of relational database. Therefore, the research on methods and tools for learning OWL ontology from RDB is a hot topic in the field of ontology learning. Kashyap [13] developed an ontology design method based on a relational schema domain database. Rubin [14] automatically acquires data from the relational database of genetic pathology and adds it to ontology.

The above work tends to mine incomplete or inaccurate domain semantics from relational systems. For example, current solutions usually interpret primary and foreign keys in relational schemas as the performance of is-a hierarchy (i.e., subtype and supertype relationships) in relational schemas. In fact, the primary and foreign keys can also be the result of the transformation of multiple relationships.

3. Preliminaries

3.1. Main ideas
Description Logic [15] (DL) is a technique for knowledge representation and description. It mainly describes the things in a specific field and the relationship between things, and can infer new concepts and relationships from known concepts and relationships. Conceptual reasoning includes determining the consistency (satisfiability), inclusion, equivalence and dissimilarity of concepts; case detection refers to determining whether a given individual is an instance of a given concept. Both the relational schema and the OWL ontology belong to the conceptual layer, and there is a perfect semantic connection between them [16]. Taking the ER model of the relational database as the intermediate model, the OWL ontology can be extracted directly from the relational model through reverse engineering [17].

The formal definitions of relational schema and OWL DL based on descriptive logic are given below.

3.2. Relational mode
Relational schema is collection of relationships. Mathematically, relations are defined as a subset of Cartesian products on a series of domains, which is basically consistent with the tables in relational database. Mathematical relationships (r), tuples (t) and tuple variables correspond to tables, rows and attributes in a relational database system.

Definition 1: A formally defined relational schema can be seen as a six-tuple: R (Tb, Attr, D, Dom, Fk, Pk), where R represents a relational schema. Tb is the collection of tables in schema R, and includes entity tables (Et) and contact tables (Rt), that is Tb∈Et∪Rt; D is the set of data fields in R, and Dom is the mapping sets from attributes to domain D.

In practice, the types of tables can be discerned by analyzing the primary key, foreign key and instance data of each table.

The relational data pattern predicates used in the mapping rules are shown in Table 1.
Table 1. The predicates of relational schema.

| Predicate          | Meaning                                                                                           |
|--------------------|---------------------------------------------------------------------------------------------------|
| Attr(Tb)           | All sets of attributes in table Tb, obviously, Aᵢ ∈ Attr(Tb) is true.                             |
| Fkey(Tb)/Pkey(Tb)  | Foreign key / primary key and Fkey(Tb) ∈ Attr(Tb) or Pkey(Tb) ⊆ Attr(Tb)                         |
| notNull(Tb)        | Table Tb's non-null constrained columns                                                            |
| Unique(Tb)         | Uniqueness Constraint Column of Table Tb                                                           |
| [... ]             | A set of attributes, such as \{fkey(Tb)\}, represents a set of foreign key columns               |
| ref(fk(Tb),pk(S))  | S ∈ Et, Tb ∈ Rt; fkeyᵣ(Tb) ⊆ attr(Tb) and fkeyᵣ(Tb) ⊆ Pkey(S), (n > 1)                           |
| Tb.t               | Tuple t of table Tb                                                                               |

3.3. OWL model

The OWL [18] language consists of three sub-language: Lite, DL, and Full, which are enhanced in their ability to express. Among them, OWL DL supports situations that need not only abundant expressive ability but also strong reasoning function. OWL DL contains all language constraints in OWL language to ensure the computational integrity and determinability of reasoning. An OWL DL ontology for capturing domain knowledge in relational databases is defined as follows:

**Definition 2.** The OWL ontology can be represented as binary Onto = (CeptID, Axiom), where:

1. CeptID = CID ∪ DRID ∪ OPID ∪ DPID is a limited set of identifiers, which is divided into disjoint sets: CID is class identifier; DRID is data range identifier; OPID is identifier of object attribute; and The DPID is data type property identifier.

2. Axiom is a finite axiomatic sequence, which includes class axioms and attribute axioms. Each axiom is implemented by several constructors acting on identifiers or descriptions (such as SubClassOf, ObjectProperty, etc.). The description is either a direct class identifier or an anonymous class constructor restriction constructed from an identifier.

The OWLDL grammatical descriptions of the above identifiers, descriptions, class axioms and attribute axioms are given in Table 2.

Table 2. Outline of OWL Language (Part)

| Abstract syntax | Description                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------------|
| Class           | A set of classes that share some of the same attributes. There is a built-in public class Thing, which is the parent of all classes |
| SubClassOf(Cᵢ, Cᵣ) | Description of parent and child classes in OWL, where Cᵢ represents child classes and Cᵣ represents parent classes |
| objectProperty/ dataProperty | Object attributes/data type attributes                                                               |
| domain/ range   | Definition Domain/Value Domain of Object (Data Type) Attribute                                      |
| Restriction(...) | A constraint in OWL, such as Cardinality, etc.                                                     |
| equivalentClass/ equivalentProperty | Equivalent classes will contain the same individuals/attributes that can be described as equal. Equivalent attributes associate each identical individual with the same set of other individuals |
| Ii(C)           | An instance of a class C in OWL.                                                                   |
| Min(max)Cardinality | Constraint declarations made on an attribute by a specific class are used to illustrate its value. |
| FunctionalProperty | Attributes can be declared to have only one value.                                                   |
| InverseOf        | inversion                                                                                         |

4. Mapping Relational Schema to Ontology

The mapping algorithm can be divided into five parts: table mapping, column mapping, constraint mapping, data mapping and axiomatic mapping.

4.1. mapping of tables

For discerning different semantics from different relational structures, we classify the table types into entity tables (Et), relational tables (Rt), and so on.
Connection table is a product of many-to-many in ER model, which is formally described as:
\[ T \in R \land \forall \text{Attr}(T) = \text{fkey}(T) \land \forall \text{Attr}(T) = \text{pkey}(T) \].
(1) Mapping join tables in relational databases to a pair of reciprocal object properties in OWL.
\[ \forall T_i \in R \Rightarrow R(T_i) \rightarrow \text{Class}(C_i) \]
\[ \text{InverseOf} (\text{ObjectProperty(domain}: T_{11}, \text{range}: T_{12}), \text{ObjectProperty(domain}: T_{21}, \text{range}: T_{22})) \]
\[ T_{11} \text{ and } T_{22} \text{ are reference tables for table } T_i. \]
(2) Mapping relational database entity table Ti to class Ci in OWL.
\[ \forall T_i \in R \Rightarrow E(T_i) \rightarrow \text{Class}(C_i) \]
\[ \forall A_i, A_i \in \text{pkey}(T_i) \text{ and } A_i \notin \text{fkey}(T_i) \text{ then } E(T_i) \rightarrow \text{Class}(C_i) \]
Else \( E(T_i) \rightarrow \text{ObjectProperty}(P_i) \)
A. If \( |\text{pkey}(T_i)| = 1 \) or \( |\text{pkey}(T_i)| > 1 \), \( \exists A_i, A_i \in \text{pkey}(T_i) \text{ and } A_i \notin \text{fkey}(T_i) \text{ then } E(T_i) \rightarrow \text{Class}(C_i) \)
B. If \( T_i \) and \( T_j \) have the same primary key and there are dependencies between them, then \( T_i \) and \( T_j \) are mapped to the same concept.
\[ \forall T_i, T_j, \text{if } \text{pkey}(T_i) = \text{pkey}(T_j) \text{ and } (\text{pkey}(T_i) \subseteq \text{pkey}(T_j), \text{pkey}(T_i) \in \text{pkey}(T_j) \text{ then } E(T_i) \rightarrow \text{Class}(C_i); E(T_j) \rightarrow \text{Class}(C_i) \text{ or } E(T_i) \rightarrow \text{Class}(C_j); E(T_j) \rightarrow \text{Class}(C_j); \text{SubClassOf}(C_i,C_j) \] As example, Table 3 shows a relational database schema for an online examination system, which consists of three columns: Relations, Primary Key and Foreign Key: reference. The first column is the definition of the relationship name, the second column represents the primary key of each relationship, and the last column gives the foreign key of each relationship and its reference relationship.

| Relation     | Primary key | Foreign key : reference |
|--------------|-------------|-------------------------|
| question     | ques_id     | ques_id : ques_type_id  |
| question_type| ques_type_id |                         |
| question_tag | question_id, tag_id | question_id: ques_id, tag_id : tag_id |
| question_point| question_id, point_id | question_id: ques_id, point_id: point_id |
| knowledge_point | point_id | field_id : field_id |
| field        | field_id    |                         |
| user         | user_id     |                         |
| role         | role_id     |                         |
| user_role    | user_role_id, role_id | user_role_id: user_id, role_id: role_id |
| tag          | tag_id      | creator:user_id        |
| tag          | tag_id      |                         |
| news         | News_id     | user_id: user_id       |
| comment      | comment_id  | user_id: user_id, question_id: ques_id |

Example 1: 

\[ <\text{owl:Class rdf:ID}="\text{news"} /> \]
\[ <\text{owl:Class rdf:ID}="\text{role"} /> \]
\[ <\text{owl:Class rdf:ID}="\text{user"} /> \]

……

\[ <\text{owl:onProperty}> \]
\[ <\text{owl:ObjectProperty rdf:about}="#\text{userID"} /> \]

\[ </\text{owl:onProperty}> \]

4.2. Column mapping
Map non-foreign key columns in tables to dataTypeProperty in OWL.
\[ \forall \text{Attr}(T_i) \land \neg \text{fkey}(T_i) \Rightarrow \text{DataProperty(domain}: T_i, \text{range}: \text{typeof(Attr}(T_i))) \]
(1) \( \forall T_i \), if \( |\text{fkey}(T_i)| \geq 1 \) \( \land \) \( T_i \subseteq T_j(A_i) \in \text{pkey}(T_i) \) \( \land \) \( \text{E}(T_i) \rightarrow \text{Class}(C_i) \), \( A_i \rightarrow \text{ObjectProperty}(P_i). \) \( \text{Domain}(P_i) \subseteq \text{Class}(C_i) \) and \( \text{Range}(P_i) \subseteq \text{Class}(C_j) \).
(2) \( \forall T_i \), if \( |\text{fkey}(T_i)| \geq 1 \) \( \land \) \( T_i \subseteq T_j(A_i) \in \text{pkey}(T_i) \) \( \land \) \( \text{Attr}(T_i)-(\text{pkey}(T_i) \cup \text{fkey}(T_i)) \) \( \land \) \( |A| \geq 1 \), then \( A_i \rightarrow \text{DatatypeProperty}(P_i)(A_i \in A) \).
∀ Tk, Ti, Tj, Tk, Ti, and Tj are associated through Tk, if pkey(Ti) = pkey(Tj) ∧ (pkey(Ti) ∈ pkey(Ti)), then Et(Ti) → Class(Ci) ∧ Et(Tj) → Class(Cj) or Et(Ti) → Class(Ci) ∧ Et(Tj) → Class(Cj) ∧ SubClassOf(Ci, Cj). ∧ pkey(Ti) → ObjectProperty(Pi, Domain:Ci, Range:Tj), pkey(Tj) → ObjectProperty(Pj, Domain:Cj, Range:Ti), ∧ Inverse(Pi, Pj).

Example 2: <owl:ObjectProperty rdf:about="#userID">
<rdfs:domain rdf:resource="#news"/>
<rdfs:range rdf:resource="#user"/>
</owl:ObjectProperty>

4.3. Data type mapping
Because OWL uses RDF's datatype schema, RDF refers to XML Schema's datatype. These data types may be identified through URIs.

4.4. Constraint mapping
(1) Non-null constrained mappings: ∀ notNull(Ti) → restriction(DataProperty, minCardinary = 1);
∀ ¬ notNull(Ti) → restriction(DataProperty, minCardinary = 0)
(2) Unique constraint mapping: ∀ unique(Ti) → restriction(InverseFunctionalProperty)
(3) Primary key constraint mapping:
∀ pkey(Ti) → restriction((DataProperty, minCardinary = 1), InverseFunctionalProperty)
(4) Foreign key constraint mapping: ∀ pkey(Ti) → ObjectProperty(domain:Ti, range:Tj)

4.5. Row mapping
When a table in a relational database is mapped to a class in OWL, all rows (tuples) belonging to the data table are mapped to an instance of the class. The formal representation is as follows:
∀ (Ti → Ci) ⇒ ∀ tj ∈ Ti → Individualj(Ci)
(1) If Et(Ti) → Class(Ci) then Ti.t → Individual(Ci) and t[Ai] → property(Pi). (Ai ∈ attr(Ti))
(2) ∀ (Ti → Rj). then Ti → Individual(Rj) and t[Ai] → property(Pi). (Ai ∈ attr(Ti)).

5. The process of transformation realization
The mapping process enhances the semantics of the database by providing additional ontology entities. The proposed mapping rules includes the following steps:
(1) Metadata extraction technology of schema information in relational databases is very mature. Metadata can be extracted including relationships, attributes, attribute types, primary keys, foreign keys/inclusion dependencies, etc.
(2) Applying the mapping rule of Part 4, the database entity is mapped to the ontology entity according to the obtained information. The specific steps are as follows:
A. Tables in relational schema are extracted in turn and converted into OWL ontology class identifiers and class axioms according to mapping rule 4.1. The identifier of class is the same as the table name. If exists table Ti, the description of class axiom is: <owl:Class rdf:ID = "#Ti">.
B. Judge the primary and foreign key situation of all tables step by step, if the number of primary keys is 1, then the entity table, continue to judge the columns in the table, if not, then rule 4.2 (1); if listed as a foreign key, and refer to another table Tj, and for the primary key of Tj, then mapping rule 4.2(2) is run, and according to mapping rule 4.3(3), (4) the transformation of primary and foreign key constraints is carried out; The number of primary keys in a table is 2, and each primary key refers to the foreign keys of other tables separately. Rule 4.2 (3) is executed, and the transformation of primary and foreign key constraints is carried out according to Rule 4.3 (3), (4).
C. Judge the foreign key situation of each table successively, if there is no foreign key, the identification table has been converted; if the table has foreign key, judge whether the tuple set of the reference table has been mapped to the individual of OWL, and convert the null value and uniqueness constraints according to Rule 4.3 (1), (2).
(3) Evaluate, validate and refine these mapping rules. Check whether all relational entities are mapped to the corresponding ontology entities. In addition, the implicit semantics of the relational model must be mapped to the explicit ontology structure.

(4) Forming a knowledge base based on OWL ontology, mapping the non-foreign key attributes of each tuple to the values of the data type attributes of the individual, and mapping the foreign key attribute columns of the tuple to the values of the object attributes of the individual.

6. Conclusion
Ontology learning is the pivotal technology for efficiently building a knowledge base system. At present, the data structure of a large number of existing knowledge base systems is mainly relational model. It is of great significance to study ontology extraction from relational database. To solve the problem of model differences between relational database schema and ontology, this paper proposes a mapping rule based on relational schema to learn OWL ontology. This paper mainly studies and implements the rules of mapping to OWL ontology for data structure and integrity constraints in relational schemas. As for the structure of relational data, all kinds of relationships between entities are described by relationships (tables). Therefore, in the concept mapping rule, the hierarchical relationship of the OWL ontology is extracted according to the semantic information implicit in the primary key, the foreign key and its correlation. The mapping rules for integrity constraints are complex, including rules for entity integrity, referential integrity, and domain integrity. Mapping rules combine similar cases to simplify the processing. The ontology model is described by the OWL language, and the ontology axioms are represented by special labels. Finally, a prototype system tool based on Java platform is used to map the case database, which proves the correctness of the mapping rules. The significance of this paper is to provide an effective method for building a new ontology-based knowledge base from the existing knowledge base system, so as to achieve maximum knowledge reuse.

Acknowledgments
This work was financially supported by the Natural Science Research of Jiangxi University of traditional Chinese Medicine (2013ZR0073), Teaching reform project of Jiangxi University of traditional Chinese Medicine(2013jzjs-15), 2017 Jiangxi Provincial Department of Education Science and Technology(GJJ161671).

References
[1] Chunhua Liao, Yanqiu He, Guangyao Xiong. Studies on ontology-based TCM knowledge base construction mode. Proc of the 9th International Economics, Management and Education Technology Conference(IEMETC 2017):AEBMR, vol.48. Paris:Atlantis Press: 2017(12):392-397.
[2] Duineveld A J, Stoter R, Weiden M R, et al. WonderTools? A comparative study of ontological engineering tools[J]. International Journal of Human-Computer Studies, 2000, 52(6):1111-1133.
[3] Noy NF, Fergerson RW, Musen MA. The knowledge model of protégé-2000: Combining Interoperability and Alexibility[J]. In: Dieng R, Corby O, eds. Proc. of the EKAW 2000. Heidelberg: Springer-Verlag, 2000. 17−32.
[4] Sure Y, Angele J, et al. OntoEdit: Collaborative ontology engineering for the semantic Web. In: Horrocks I, Hendler JA, eds. Proc. of the ISWC 2002. Heidelberg: Springer-Verlag, 2002. 221−235.
[5] Bozsak E, Ehrig M, Handschuh S, et al. KAON - Towards a large scale semantic web. International Conference on E-Commerce and Web Technologies. Heidelberg:Springer-Verlag, 2002:304-313.
[6] Lenat DB. CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM, 1995,38(11):33−38
[7] Nirenburg S, Beale S, Mahesh K, et al. Lexicons in the MikroKosmos Project. University of Sussex, 1996:26-33.
[8] Shamsfard M, Barforoush A A. Learning ontologies from natural language texts. International Journal of Human-Computer Studies, 2004, 60(1):17-63.
[9] Agirre E, Ansa O, Hovy E, Martinez D. Enriching very large ontologies using the WWW. In: Staab S, Maedche A, eds. Proc. of the ECAI 2004 Workshop on Ontology Learning. 2000. http://ol2000.aifb.uni-karlsruhe.de/.

[10] Missikoff M,Navigli R, Velardi P. Integrated approach for web ontology learning and engineering. IEEE Computer, 2002, 35(11): 60–63.

[11] Papatheodorou C, Vassiliou A, Simon B. Discovery of Ontologies for Learning Resources Using Word-Based Clustering. [J]. Educational Resources, 2002, 2002(1):1523--1528.

[12] Modica, Giovanni, A. Gal, and H. M. Jamil. "The Use of Machine-Generated Ontologies in Dynamic Information Seeking." Cooperative Information Systems, International Conference, Coopis, Trento, Italy, September DBLP, 2001.

[13] Kashyap V. Design and Creation of Ontologies for Environmental Information Retrieval[C]// Proc. of the 12th Workshop on Knowledge Acquisition, Modeling and Management. 1999.

[14] Rubin D L, Hewett M, Oliver D E, et al. Automating data acquisition into ontologies from pharmacogenetics relational data sources using declarative object definitions and XML. Pac Symp Biocomput, 2002:88-99.

[15] Baader F, Calvanese D, McGuinness D, et al. The description logic handbook. Theory, implementation, and applications. Kybernetes, 2003, 32(9/10):43-95.

[16] XU Zhuo-Ming, DONG Yi-Sheng, LU Yang. Semantics-Preserving Translation from ER Schema to OWL DL Ontology. CHINESE JOURNAL OF COMPUTER, 2002(10):1786-1796.

[17] Hainaut JL., Henrard J., Englebert V. et al. Database Reverse Engineering. In Encyclopedia of Database Systems. LIU L., ÖZSU M.T. (eds) New York: Springer, US, 2009, pp. 723-728.

[18] DEAN M, SCHREIBER G. OWL Web ontology language reference[EB/OL]. (2004-02-10). https://www.w3.org/TR/owl-ref.