Using Reverse Engineering for Building Ontologies with Deeper Taxonomies from Relational Databases

Sara Sbai[1*], Mohammed Reda Chbihi Louhdi[2], Hicham Behja[1], El Moukhtar Zemmouri[3], Chakhmoune Rabab[1]
[1] High National School of Electricity and Mechanics (ENSEM), Hassan II University, Casablanca, Morocco
[2] Faculty of Sciences Ain Chock, Hassan II University, Casablanca, Morocco
[3] High National School of Arts and Crafts (ENSAM), Meknes, Morocco

* Corresponding author: email: sara.sbai4@gmail.com
Manuscript submitted October 30, 2018; accepted December 20, 2018.
doi: 10.17706/jsw.14.3.138-145

Abstract: The relational model is characterized by its high quality and has been widely used by information systems.

However, unlike the conceptual model, the relational model is semantically poor since it doesn't enable the representation of inheritance. In this paper, we present an algorithmic approach to extract generalization/ specialization inheritance hierarchies. We perform a reverse engineering by analyzing stored data. Finally, we evaluated our approach by conducting several experiments on relational databases.

The results were satisfying in terms of recovering the tables lost during the transformation from the entity relationship model to the relational model.

Key words: Relational databases, generalization inheritance, specialization inheritance, reverse engineering, ontology generation.

1. Introduction

The semantic web aims to make data understandable and shareable; it also solves the problem of interoperability between applications. To express semantics and integrate them in the data circulating through the web, we use ontologies.

There are several definitions of ontology in literature, Tom gruber [1] defined an ontology as "an explicit specification of a conceptualization of a domain of interest", as for Swartout and collegues [2], they defined an ontology as "a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base".

Creating ontology manually is a very complex task. Therefore, several researchers are interested in building ontology from relational databases.

The relational model is the most popular model due to its simplicity and ability to process data with storage efficiency, it also contain hidden semantics that can be exploited during the ontology extraction. The relational model lacks expressiveness compared to the conceptual model, that is why some existing approaches [3]-[6] offered to do a reverse engineering to transform the relational model to a conceptual model because the latter is semantically richer.

In this paper, we propose algorithms that enable the recovery of generalization and specialization inheritance hierarchies from the relational model. Our approach is based on a

previous approach [7] for generalization/specialization reverse engineering.

The remainder of this paper is organized as follows. Section two discusses related works in ontology extraction from relational databases. Section three describes the proposed method for detecting multi-level generalization/specialization inheritance. Section four presents the experimentations and the results. And finally, Section five concludes this paper, and discusses the perspectives of this work.

2. Related Works

Many approaches have been proposed for ontology extraction from relational databases. The techniques used in ontology building from relational databases are generally based on [8], [9]: reverse engineering, schema mapping and data mining.

Reverse engineering technique's goal is to extract the conceptual model from the relational model of an existing database. The approach proposed in [3] describes a method on generalization hierarchies reverse engineering. The approach uses a combination of heuristic and algorithmic rules. It takes into account inclusion and existence dependencies, intersection and exclusion constraints and null values.

In [4], an automatic approach on database reverse engineering is proposed; this approach uses the entity relationship schema as an intermediate between the relational schema and the OWL ontology in the engineering process. This approach extracts natural domain semantics from the relational database by doing an in-depth analysis of the conceptual correspondences between the entity relationship model, relational model and the OWL ontology in the forward and reverse engineering processes. Two algorithms were proposed for the extraction process: (1) Table type identification algorithm, which classifies different types of entity tables and relationship tables and (2) Schema translation algorithm, which specifies a set of schema translation rules that enable the translation of a relational database schema into OWL DL ontology.

Another approach [5] proposed to perform a reverse engineering from the relational model to the conceptual model. It consists of the following steps: (1) Extract ER model from relational database by reverse engineering tool or querying for database system tables, (2) Analyze the ER model extracted from step1 then transfer it to an OWL ontology model by schema conversion mechanism, (3) Transfer the database data to ontology instances by using data conversion mechanism, and finally (4) Evaluate the OWL file by existing ontology engineering tools.

In [6], the authors proposed an approach for automatic ontology construction from relational databases. This approach uses as input a relational database written in SQL and its output is OWL ontology. This method is based on three steps. In the first step, metadata is extracted from the relational database and the database tables are classified based on it. In the second step, a conceptual middle model in the form of a graph is produced using the information extracted in the previous step. In the final step, an OWL ontology model is created using the graph model obtained in the last step. Triggers are used in this method; they are used to increase the amount of power and expressiveness of knowledge by presenting part of the knowledge dynamically.

In [7], the authors propose a hybrid method to extract ontology from a relational database. This method consists of the following steps: (1) Reverse engineering to identify generalization/specification inheritance case, (2) Classifying tables of the database into six categories, (3) Apply the transformation rules and data analysis to transform the tables of each category to an ontological component, and finally (4) A refinement phase to rename the ontology concepts to make them more expressive. An algorithm is proposed in the first step to extract generalization inheritance and recover the tables lost during the transformation from the entity relationship model to the relational model. This approach consists on using the NULL values in the relational database table to perform a reverse engineering based on the analysis of stored records. As for the specialization inheritance, the authors search tables that have the entire or apart of the primary key

with the same names and types.

In the schema mapping technique, a set of transformation rules is defined to convert relational database schema to ontology. This technique is usually incorporated together with the reverse engineering technique or the data mining technique. In [10], the authors propose new rules for the direct mapping of Relational database schema and data to RDF(S)/OWL ontology automatically. The mapping rules are divided into two parts: Rules for mapping Relational database schema to ontology and rules for mapping Relational database instances into ontology.

Another approach [11] uses the OWL2 language for describing ontologies. It uses mapping rules to define the ontology classes, functional properties, individuals, restrictions, object properties and cardinality.

In [12], the authors proposed an approach for ontology learning from relational databases. This approach is composed of three sub processes that aim to overcome the limits of previous approaches in addressing the peculiarities of real world databases. The sub processes are preceded by a pre-processing phase to clean the database from irrelevant and erroneous information that may affect the learning process. The first sub process transforms the data model of the relational database to an XML document. The second sub process is a semantic enrichment to upgrade the semantic of the relational schema. In this step, additional information can be inserted in the XML document either automatically through an API or manually by a domain expert. The final sub process is the automatic transformation of the enriched relational schema to an OWL2 ontology.

Another approach [13] proposed to transform Relational Database Management System (RDBMS) schema into ontology model. The first step is to represent the relational model as a semantic model. The semantic model is a conceptual data model that uses instances to enable meaning interpretation. The semantic tables are then implemented in an oracle database using SQL (Structured Query Language). The next step is creating a Java program using JDBC API to map and transform the relational database metadata to standard Ontology description through the help of DOM/XML and importing the created Ontology description to an Ontology editor (Protégé) to form a standard Ontology structure.

Data mining technique is used in [14]. The authors propose an approach that uses concept hierarchy as background knowledge in order to accelerate the extraction of ontology from Relational databases process and guide the extraction of knowledge that resides in the database. The approach uses the database schema and instances to construct the ontology, and also uses concept hierarchy as the background knowledge to select the relevant dataset and to specify the kind of knowledge to be extracted. In another approach [15], RTAXON combines schema and data analysis to exploit the content of the database to find deeper class hierarchies.

Recently in [16], the authors used machine learning techniques. They proposed a semantic approach to automatically generate ontology from heterogeneous relational databases. This method uses Wu and Palmer's semantic similarity measure [17] and WordNet as a lexical database to help select the best terms to represent ontology components. This approach provides the possibility to generate an ontology from many relational databases in the alimentation risks domain.

In this work we propose algorithms to recover the tables lost during the transformation from the entity relationship model to the relational model.

3. Motivation

As we mentioned before, the process of building ontologies from scratch is tedious and error-prone, that is why several researchers choose to build ontologies automatically from relational databases. Several previous works use the entity relationship model as an intermediate between the relational model and the OWL ontology.

During the transformation from the entity relationship model to the relational model, different possibilities can be used by the designer to translate inheritance links. Among these possibilities we find generalization and specialization.

In this paper, we use reverse engineering techniques to extract generalization/specialization inheritance hierarchies. Our goal is to recover the tables lost during the transformation from the entity relationship model to the relational model. This will eventually enable us to generate a deep taxonomy for the extracted ontology.

4. Proposed Method

4.1. A Multi-level Generalization Inheritance Hierarchies Extraction Algorithm

The basic idea for generalization is to eliminate all the sub tables and include their attributes in the super table when transforming the entity-relationship model to the relational model. This transformation will introduce null values. Extracting generalization inheritance hierarchies allows building ontologies that are semantically richer. Our algorithm is based on the approach proposed in [7]. The results of the experiment of the latter were satisfying but they only succeeded in extracting one level of the generalization hierarchy. In our approach, we propose algorithms to recover the different levels of hierarchies in the generalization inheritance. It is based on performing a reverse engineering by analyzing the database records. Our algorithms are presented in Fig. 1.

4.2. Rebuilding Multi-level Hierarchies for Specialization Inheritance

In the enhanced entity-relation model, specialization is the process of defining a set of subclasses of an entity type; this entity type is called the superclass of the specialization. The set of subclasses that forms a specialization is defined on the basis of some distinguishing characteristics of the entities in the superclass [18].

During the conception of a database, designers can build multi-level hierarchies, exploiting the power of the inheritance mechanism.

When mapping entity-relation model into the relational model, the inheritance cases can be mapped in different ways, among them, we can use specialization: we remove the super-type and duplicate all of its attributes in each of the sub-tables. This transformation can be done if there is totalness (every entity in the super-type must be a member of at least one subtype in the hierarchy) in the inheritance relation and disjointness (an entity can be a member of at most one of the subtypes of a hierarchy) between subtypes.

In the entity-relation model, each component must have a name with unique meaning. To detect specialization cases in the relational model, we must search tables that have the entire or a part of the primary key with the same names and types. In our previous work [7], we proposed to rebuild the hierarchy by analyzing the column's names and types of tables, and creating a new table (the super-type) that regroup all the common columns, these columns are removed from their original tables (subtypes). However, the proposal can only deal with one level hierarchy.

In this paper, we propose a recursive method that can rebuild multi-level hierarchies from a set of specialized tables. This method contains 5 steps and takes as input a JSON Array containing a subset of specialized tables (each table with its name and columns):

Step 1: finding common columns (CC) from specialized tables (ST)

Step 2: building a new table containing the CC

Step 3: removing the CC from the ST

Step 4: build a new JSON Object containing the new table with its columns (CC) and subtypes (ST)

Step 5: analyzing the subtypes, if they have CC, return to step 1 and apply the method to the subtypes,

else the treatments are stopped.

```
BinaryVectorAlgo
{Binary Vector construction Algorithm}
Input: Relational Database table T
Output: List V
Let val(ind)a function retrieving the value of the attribute with the index ind for the current record
Let attr(ind)a function that returns the attribute with the index ind for the current table
Let P: Positive integer
Let C: String
FOR each record E of T
        C←""
    FOR each attribute A of E
        IF val(A)<>NULL THEN
                C ← C + "1"
        ELSE
                C ← C + "0"
        ENDIF
    ENDFOR
    ADD C to V
ENDFOR
P ←The number of the attributes of T
Remove all duplicates from the list V
RETURN (V)
END
```

```
Generalization
{Generalization Inheritance Hierarchies Detection Algorithm}
Input:
- List V : A list of binary vectors
- P :Number of attributes of table T
- Key : Integer to indicate the current level of hierarchy
Output: M : A map that contains the attributes of each table in the different levels of hierarchy
Let attr(ind) a function that returns the attribute with the index ind for the current table
Let val(ind) a function that returns the value of the attribute with the index ind for the current table
Let extract(□,i,P-1) a function for extracting a subvector from vector □ from the index i to the index P-1
Let setData(M,key,Table_attr_list) a function to set the value of the elements of M
Let n, S: Positive integers
Let Table_attr_list: A list of attributes
Let n the size of the list V
IF n >1, THEN
  S ← 0
  FOR each element □ of V
      FOR i=0 TO P – 1
          S ←S + □[i]
      END FOR
  END FOR
  Table_attr_list←NULL
  IF S contains n THEN
    FOR each column of S
      IF S[i] equals n THEN
          ADD attr(i) TO Table_attr_list
      END IF
    END FOR
    FOR each □ of V
        Replace val(i) by '0'
    END FOR
    Replace n in S by '0'
     CASE S IN
       0: Continue
       Else:
           Let R a list
           Let L a list
           FOR each element □ of V
              IF(□[i]==1) THEN
                  ADD Extract(□,i,P-1) TO R
              ELSE IF(□[i]==0) THEN
                  ADD Extract(□,i,P-1) TO L
              END IF
           END FOR
           IF Table_attr_list is not empty THEN
                  setData(M, Key, Table_attr_list)
           END IF
           Generalization(R, P, Key+1)
           Generalization(L, P, Key+1)
     END CASE
  ELSE
      {Each attribute belongs to a different table in the current level of hierarchy}
  END IF
ELSE
```

	{This table didn't undergo a generalization}

Fig. 1. Proposed algorithm for binary vector construction and detecting multi-level generalization inheritance.

5. Experiments and Results

To evaluate the efficiency of our proposed algorithms, we implemented them with java. We conducted experiments using a normalized database called SAKILA[1] and that is available on the official MySQL website (where the detailed relational model can be found). This database was intended to provide a standard schema that can be used for examples.

After applying our algorithms on SAKILA, we successfully recovered four new tables through our algorithm of detecting multi-level generalizations, (1) a sub table of the table "*Address*", (2) a sub table of the table "*payment*", (3) a sub table of the table "*rental*" and (4) a sub table of the table "*staff*". We also recovered a super table through the proposed method for rebuilding hierarchies from specialization inheritance for the 'tables "*film*" and "*film_text*". We used the same transformation rules used in [6] to obtain the result ontology (Fig. 2).

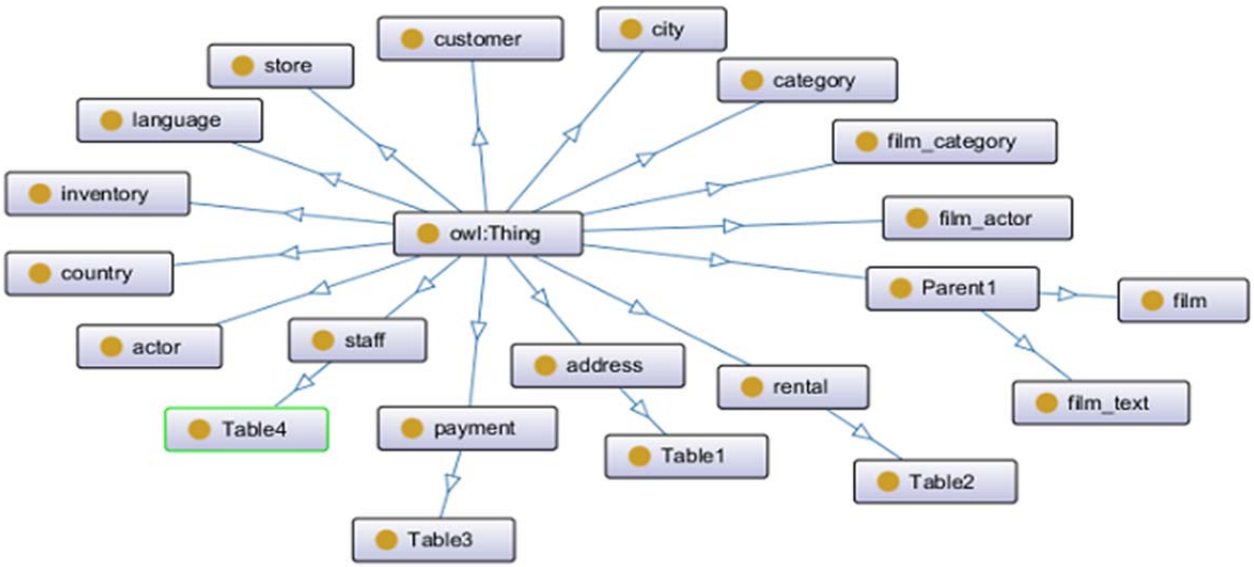

Fig. 2. The obtained ontology from Sakila database.

6. Conclusion

In this paper, we have proposed an approach to extract multiple levels of the generalization/specialization inheritance hierarchies by doing a reverse engineering based on analyzing the database records. The approach has allowed us to recover lost tables during the transition from the conceptual model to the relational model in all the different hierarchy levels of the inheritance.

We have tested our approach using a relational database example. The obtained results were satisfactory in terms of extracting of the generalization/specialization hierarchy tables.

As we mentioned before, we use the null value to do tests and make the decisions. However, sometimes default values are used instead of null values. We intend to improve our approach to handle this kind of situation.

[1] SAKILA - https://dev.mysql.com/doc/sakila/en

Acknowledgment

This work is supported by the project "Knowledge Management for Development in the Context of OCP Group (KM4Dev – OCP Group)" granted by OCP Group, Morocco.

References

[1] Gruber, T. (2009). Ontology. *In the Encyclopedia of Database Systems*, Ling Liu and M. Tamer Özsu (Eds), Springer-Verlag, 2009.

[2] Swartout, B., Patil, R., Knight K.., & Russ, T. (1997). Towards distributed use of large-scale ontologies. *Spring Symposium Series on Ontological Engineering*. Stanford University, 138-148.

[3] Lammari, N., Comyn-Wattiau, I., & Akoka, J. (2007). Extracting generalization hierarchies from relational databases: A reverse engineering approach. *Data and Knowledge Engineering*.

[4] Lin, L., Xu, Z., & Ding, Y (2013). OWL ontology extraction from relational databases via database reverse engineering. *Journal of software*, *8(11)*.

[5] Ping, H., Lu, H., & Bin, C. (2008). Research and implementation of ontology automatic construction based on relational database. *Proceedings of the International Conference on Computer Science and Software Engineering*.

[6] Dadjoo, M., & Kheikhah, E. (2015). An approach for transforming of relational databases to OWL ontology. *International Journal of Web and Semantic Technology*.

[7] Chbihi, L. M. R., Behja, H., & Ouatik, E A. S. (2013). Hybrid method for automatic ontology building from relational database. *International Review on Computers and Software*, *8(8)*, 1801-1813.

[8] Gomez-Perez, A., & Manzano-Macho, D. (2003). A survey of ontology learning methods and techniques. *The IST Project IST-2000-29243-OntoWeb Consortium*.

[9] Maedche, A., & Staab, S. (2005). Ontology learning for the semantic web. *IEEE Intelligent Systems and Their Applications.*

[10] Hazber, M., Li, R., Zhang, Y., & Xu, G. (2015). An approach for mapping relational database into ontology. *Proceedings of the 12th Web Information System and Application Conference* 2015.

[11] Kaulins, A., & Borisov, A. (2014). Building ontology from relational database. *Information Technology and Management Science*.

[12] Idrissi, E. B., Baïna, S., Baïna, & K. (2015). Bringing flexibility to ontology learning from relational databases. *Proceedings of the 12th IEEE/ACS International Conference of Computer Systems and Applications*.

[13] Ugochukwu, A. C. (2017). Mapping of relational schema to ontology model. *Proceedings of the fourth International Conference on Artificial Intelligence and Pattern Recognition*.

[14] Santoso, H. A., Haw, S., & Ziyad, T. A. M. (2011). Ontology extraction from relational database: Concept hierarchy as background knowledge. *Knowledge-Based System*.

[15] Cerbah, F. (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*.

[16] Aggoune, A. (2018). Automatic ontology learning from heterogeneous relational databases: Application in alimentation risks field. *Proceedings of the 6th IFIP TC 5 International Conference* on *IFIP Advances in Information and Communication Technology*.

[17] Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*.

[18] Elmasri, R., & Navathe, S. (2011). Fundamentals of database Systems sixth Edition.

Sara Sbai received the MSc degree in computer engineering from the Faculty of Science, Oujda, Morocco. She is now a Ph.D. candidate at the High National School of Electricity and Mechanics, Casablanca, Morocco. Her research interests are:ontology building from structured and semi-structured data Knowledge management.

Mohammed Reda Chbihi Louhdi received in 2013 his PhD degree in computer sciences from Dhar Mahraz Sciences Faculty, Fez, Morocco. He is currently a professor in Ain Chock Sciences Faculty, Casablanca, Morocco. His research interests include ontology building from structured and semi-structured data.

Hicham Behja received in 1999 his first Doctorate degree in computer sciences from Mohamed V University, Rabat, Morocco. He received in 2007 his second Doctorate degree in Computer Sciences from Hassan II University, Mohammedia, Morocco. Graduated in 2013 with HDR in Computer Sciences. He is currently the vice director of the High National School of Electricity and Mechanics, Casablanca, Morocco. His research interests include Unified Modeling language, KDD, e-learning, data processing techniques, data mining, Knowledge Engineering and Management.

El Moukhtar Zemmouri received in 2013 his PhD degree in computer sciences from the National School of Arts and Crafts, Meknes, Morocco. He is currently a professor in the High National School of Arts and Crafts, Meknes, Morocco. His research interests include machine learning, data mining and knowledge discovery, text mining, knowledge representation, web mining, clustering algorithms.

Chakhmoune Rabab received in 2015 her PhD degree from the National School of Arts and Crafts, Meknes, Morocco. She is currently a post-doc at the High National School of Electricity and Mechanics, Casablanca, Morocco. Her research interests include knowledge engineering and management, ontologies, KDD, data mining, cloud computing.