**International Journal of Electrical and Computer Engineering (IJECE)**
Vol. 9, No. 4, August 2019, pp. 3298~3306
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i4.pp3298-3306

# Onto2DB: towards an eclipse plugin for automated database design from an ontology

**Morad Hajji, Mohammed Qbadou, Khalifa Mansouri**
Laboratory: Signals, distributed systems and Artificial Intelligence (SSDIA), ENSET Mohammedia,
University Hassan II of Casablanca, Morocco

**Article Info**

***Article history:***

Received Jan 21, 2019
Revised Mar 26, 2019
Accepted Apr 6, 2019

***Keywords:***

Eclipse
EMF
GEF
GMF
Ontology
Plug-in
Relational database (RDB)
Semantic web

**ABSTRACT**

Ontologies are spreading more and more in the field of information technologies as a privileged solution allowing the formalization of knowledge. The theoretical model of ontologies is most promising. They are increasingly ubiquitous given the benefits they present. Despite the proliferation of research proposing approaches dedicated to the design of a database from an ontology, the tools to design a database from an ontology are rare or inaccessible. Thus, in this contribution, we present our approach for the development of an Eclipse Plug-in, in order to automatically generate a conceptual model of a relational database from an ontology. To evaluate the usefulness of our approach, we used our resulting Eclipse Plug-in to automatically generate a conceptual model of a relational database from an ontology, customize it, and automatically generate the corresponding SQL script for Data Definition. The results of this experiment showed that our Plug-in constitutes a concretization of our approach and a means of automatic translation from the ontological model to the relational model.

***Corresponding Author:***

Morad Hajji,
SSDIA laboratory, ENSET Mohammedia,
Hassan II University of Casablanca,
Morocco.
Email: morad.hajji@gmail.com

## 1. INTRODUCTION

Recently, the Semantic Web field has attracted increasing interest from researchers because of its innovative features and interactions with other fields. Indeed, ontologies proliferate in several disciplines, namely medicine, biology, geology, environment, business, e-commerce, security, etc. Today, ontologies are a solution to the formal representation of knowledge of a field of study. They promote the sharing and reuse of knowledge [1] and to interoperability [2]. In fact, ontologies make it possible to structure the data in a semantic way, favoring their understanding by machines. In context our contribution and as result of the ontology learning, they constitute the appropriate means of transition from a corpus of textual documents to a semantically structured knowledge base. Our choice is supported by the advent of mature tools for building ontologies from textual resources. We particularly cite the tool Text2Onto [3].

As part of our research, we have proposed a global model for the analysis of a corpus of textual documents that was the subject of our first paper [4], whose objective is to present our approach that aims at exploiting technological advances offered by the architecture of a decision support system to analyze an ontology generated from a corpus of documents related to the field of pedagogy. This ontology, whose generation is based on ontology learning techniques, constitutes a structured source of semantic data that can be analyzed by the suggested system to deduce decisional indicators, which can be exploited in the pedagogic decision-making process. Our global model consists of several steps, which include the design of a Data Warehouse from an ontology. This step was the subject of our second paper [5] whose goal is to propose a new multi-approach model based on the coupling of relational database design approaches from an ontology with Data Warehouse design approaches from a relational database.

Unfortunately, while trying to apply our approach we faced the lack of tools to design a database from an ontology. In the literature, we find several research works that propose approaches for the design of a database from an ontology such as [6-13]. The proposed approaches in the literature introduce transformation rules allowing the automatic design of a relational database from an ontology. However, to our knowledge none of them has produced an exploitable tool to generate de conceptual model of relational database (RDB). This paper aims to overcome the aforementioned limitation and it presents our approach for developing an Eclipse Plug-in based on Eclipse Project Modeling for the automatic generation of a conceptual model of a RDB from an ontology and its implementation within a relational database management system (RDBMS). Our approach can produce more accurate and flexible results by promoting the adaptation and customization of the generated model at the conceptual level.

The flow of this paper is organized as follow: section 2 presents the model driven engineering and Elipse modeling project relationship, section 3 depicts our proposed model and presents proposed Plug-in followed by an example of implementation in section 4. The analysis of the results are presented in section 5 before concluding in section 6.

## 2. MODEL DRIVEN ENGINEERING AND ECLIPSE MODELING PROJECT

The conception of tools to design other tools is one of the concepts on which the evolution of civilizations is based. Like blacksmiths who make tools to make other tools, computer scientists develop software to develop other software. Eclipse can be considered as a workshop for the production of new software and tools, since Eclipse is composed of several tools that we call "Plug-ins". The plug-in concept allows Eclipse to be scalable. It is one of its distinctive features. Eclipse is a forge for software development. Like the hammer, anvil and forging bellows that are the starting tools of blacksmiths, a software tool of analysis, a software tool of design, coding, compilation and execution are the starting means of the software geniuses.

### 2.1. Eclipse modeling framework (EMF)

EMF is a Java framework and a code generator for building tools and applications using a structured model approach. Moreover, the software approach recommends starting with the phases relating to conceptual modeling. This Framework concretizes the perception of the implementation of a set of tools providing support for organization and structuring the development of software tools using the Eclipse platform. One of EMF's goals is to free developers of code generation and focus on model design. The fundamental concepts of this Framework are: metadata, code generation and serialization. EMF is an implementation and evolution of the standard MOF (Meta Object Facility), precisely version 2.0 of the MOF specification.

Therefore, EMF is part of the MDA architecture; it is the implementation of this architecture at Eclipse. In the context of EMF, the metamodel is named Ecore. Through EMF, Eclipse offers the necessary tools to exploit the advantages offered by the model-driven software development approach namely the MDA. EMF provides the metamodel layer in the form of Ecore defining the language needed for modeling to describe classes, attributes, data types, and relationships. Ecore is aligned with OMG's Essential Meta-Object Facility [14] which corresponds to layer four (M3) at the level of OMG architecture.

### 2.2. Graphical editing framework (GEF)

The GEF is an Eclipse project that aims to develop the tools needed to develop applications and tools for manipulating information in a graphical form. Indeed, GEF makes it possible to create applications and tools intended for the manipulation of the objects and the interaction of the users with these objects, such as: curves, graphs, diagrams, etc. This framework is based on SWT (Standard Widget Toolkit) and Draw2d to facilitate the implementation of graphic editors. The developers of GEF have adopted the Model-View-Controller (MVC) architecture whose composition comprises the model, the view or the representation. Following the creation or modification of an element at the level of the model, the controller that corresponds to it is responsible for propagating this event at the level of the visual representation by making the changes to the corresponding visual objects. This process is bidirectional, since the changes that the visual objects undergo corresponding to an object of the model, following the interactivity with the user, have repercussions on the model through the corresponding controller. In fact, the model contains the information, the view displays the information contained in the model, and the controller coordinates the exchanges between the model and the view. Figure 1 illustrates its interactions between the different components.

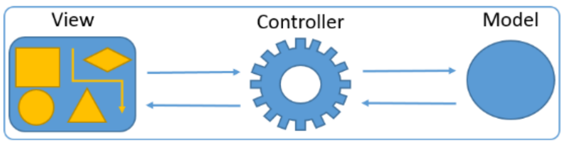


Figure 1. GEF MVC model

### 2.3. Graphical modeling framework (GMF)

The objective of GMF is to provide a generative bridge between EMF and GEF [15]. The GMF sub-project is part of the Eclipse Modeling Project, a collection of sub-projects that are part of model-driven software development. While EMF allows the modeling as well as the generation of the code from the product models and GEF allows the creation of the graphical interfaces for publishers based on models, GMF allows the connection between these two universes to provide an integrated development framework. With GMF, the transition from the model to a graphic editor is systemic profiling the beauty of the benefits brought by the MDA whose main advantage is the contribution of a high-level of abstraction. Thus, relying on EMF and GEF, GMF offers a development environment rich graphical editors based on GEF and generated from a model expressed using EMF. In extremely simplified manner, GMF makes it possible to produce graphic editors from a metamodel expressed using EMF.

GMF brings together three sub-projects: GMF Runtime, GMF Tooling and GMF Notation [16]. GMF Runtime supports bridging between EMF and GEF and offers application services (API) for the production of graphic editors. While GMF Tooling plays the role of interfacing between the developer and the GMF Runtime, it is the visible part of the MDA iceberg to define the graphic elements, diagram tooling and the association between two sets and the base domain model. As part of the GMF, the creation of a graphic editor is guided by a well-defined process framed by the MDA approach. This process begins with the creation of a project under eclipse and ends with the generation of a plug-in for a graphic editor while passing through intermediate steps, in this case the development of a model of the domain, the development of a graphic definition, the development of a tool definition, the development of a mapping model and the creation of a generation model.

## 3. TOWARDS PLUG-IN ONTO2DB

The conceptual level of a relational database is represented by the Entity-Relation model whose composition groups the concepts: entity, relation, attribute (identifier or descriptor) and the links between entities and relations (with cardinalities).

### 3.1. Architecture

The Onto2DB Plug-in is a visual modeling environment for generating a conceptual model of a RDB from an ontology. Onto2DB gives designers the ability to customize the conceptual model generated by interacting with graphical user interface widgets. As mentioned above, the development of Onto2DB is based on the Eclipse platform because of the advantages mentioned above, in this case its extensibility and the wide range of tools available as a framework. In fact, Onto2DB is implemented as an Eclipse plug-in, adding capabilities to generate and visualize conceptual models of RDBs from ontologies and manipulate these models using graphical user interface, such as entities, relationships, attributes, etc. Figure 2 shows the stacking relationship between Onto2DB and Eclipse in a layered architecture. Indeed, Onto2DB is based on the exploitation of the services offered by GMF.

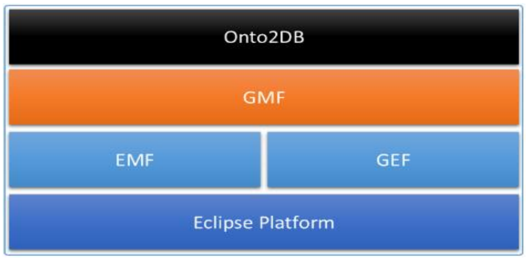


Figure 2. Onto2DB and Eclipse

Onto2DB's architecture is centred around a core that coordinates the process of designing a conceptual models of RDB from an ontology. It uses the algorithm chosen by the user to produce this model. The result is made available to the designer in a visual form representing this conceptual model of RDB as illustrated in Figure 3.

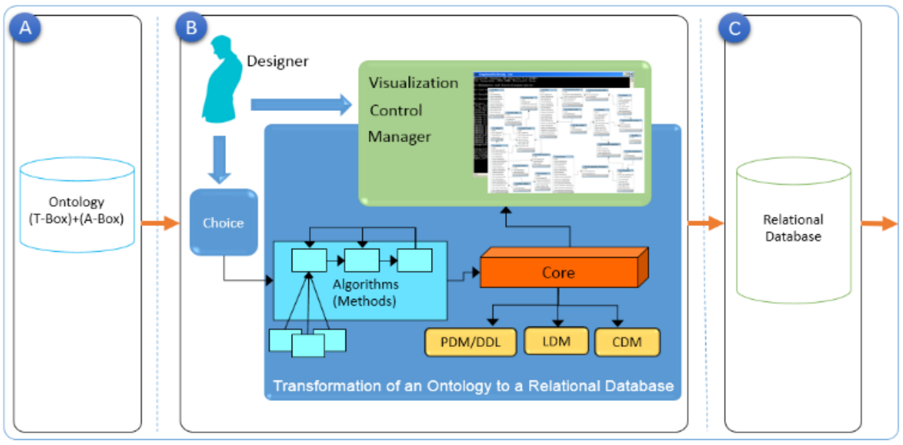
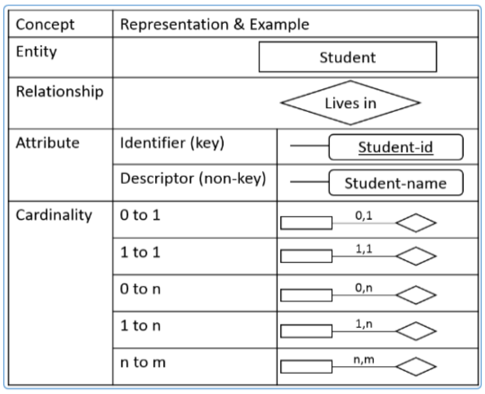


Figure 3. Onto2DB internal architecture

There are two types of ontology relational coupling methods depending on the type of generated schemas. The first kind of methods aims to design the Conceptual Data Model (CDM) from an ontology as mentioned by Y. Lv and C. Xie in [17], while the second kind of methods is focused on the design of Logical Data Model (LDM) from an ontology as stated by I. Astrova, N. Korda and A. Kalj in [8]. In the first case, the generated CDM will be transformed into a LDM. The logic model is then translated into a physical data model (PDM) according to the chosen RDBMS. The physical model is used to generate the SQL DDL (Data Definition Language) script whose execution produces the physical structure of the relational database within the RDBMS.

## 3.2. Symbols

The symbols used and their meanings for modeling a schema of an Entity-Relationship conceptual model are shown in Figure 4. In fact, a conceptual model of a RDB includes entities with attributes, relationships with or without attributes in addition to links between entities and relationships, and links between attributes and their entity or relationship. An entity is represented by a rectangle, a relationship is represented by lozenge, an attribute is represented by a rounded rectangle, an attribute forming part of the identifier of an entity is represented by a rounded rectangle with the underlining of the text and finally a link between an entity and a relation by a line labeled with cardinality.



| Concept | Representation & Example | |
|---|---|---|
| Entity | | Student |
| Relationship | | Lives in |
| Attribute | Identifier (key) | Student-id |
| | Descriptor (non-key) | Student-name |
| Cardinality | 0 to 1 | 0,1 |
| | 1 to 1 | 1,1 |
| | 0 to n | 0,n |
| | 1 to n | 1,n |
| | n to m | n,m |

Figure 4. Basic ER model symbols

### 3.3. Metamodel

We advocate for the use of EMF to define the underlying model. As illustrated in Figure 5, we use the Ecore graphical editor in order to specify this model. We consider that a conceptual model of a RDB is composed of elements and links. Entities, relationships, attributes, and attributes that are part of an identifier are elements. There are two types of links: the links that link an attribute to its entity or relationship, and the links that link entities and relationships.

To programmatically manipulate an ontology, we used OWL2API [18] to generate the conceptual model of a RDB and to leverage EMF support for resource management for the persistence of this conceptual model and the generation of the DDL SQL script. In this context, it should be taken into account that an ontology can be divided into two parts: TerminologicalBox (TBox) and AssertionBox (ABox) [19]. TBox corresponds to the conceptual model of an ontology representing the concepts linked together by semantic relations. To each concept are attached specific properties, which uniquely identifies it in a domain [20]. ABox regroups all the instances of the concepts in addition to the instances of the relations as well as the values of the properties [19].

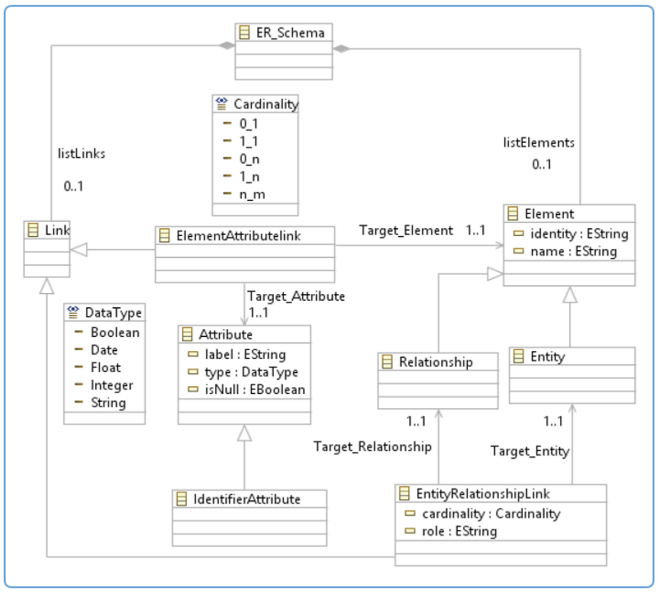


Figure 5. Onto2DB metamodel

### 3.4. Onto2DB Design

The constitution of the user interface will comprise, as is illustrated in Figure 8, a graphic editor, an explorer view, a Properties view and Outline view. The graphic editor will display the schema resulting from the automatic generation of the conceptual schema of a database from an ontology, and will also allow users to edit this schema by making modifications according to needs. As for the explorer view, it will allow users to browse the resources of a project and possibly navigate through a model (schema) and its components. The properties view is used to edit the properties of a model (schema) and its elements (entities, relationships, attributes, identifiers and links). Thus, the elements of a model are represented graphically at the editor level and their properties are manipulable at the level of the property view. The editor has a palette that groups all the types of graphic elements for the design of database models, such as Entity, Relation, Attribute, Link between Attribute and its Entity/Relation. A user can therefore modify a schema automatically generated from an ontology. After checking/modifying the schema and its properties as well as those of its elements, the user can proceed to the generation of a corresponding DDL SQL script which he can implement within a RDBMS.

## 4. EXAMPLE CASE STUDY

In order to show how our Plug-in works, we have opted for a simplistic ontology representative of the field of purchases and sales. This ontology has three basic concepts: customers, products and orders. Figure 6 illustrates the creation of this ontology using Protégé ontology manipulation tool.

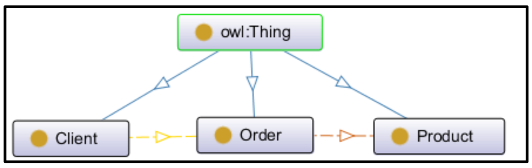
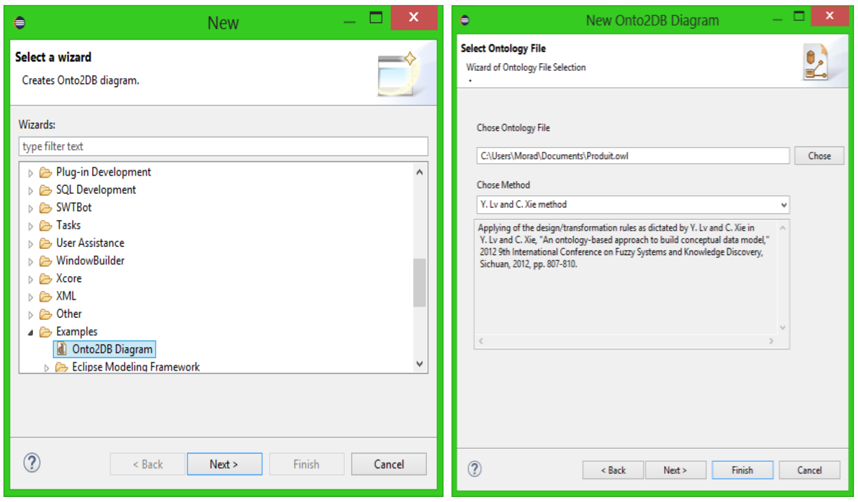


Figure 6. Ontology example

After running our Plug-in and creating a new project under the name "TestOnto2DB", we proceed to the creation of a new Onto2DB Diagram as shown on left side in Figure 7. After this action, the wizard offers a window for selecting the ontology source file in addition to the generation method. This wizard allows us to complete the action by clicking on the "Finish" button or to continue by clicking on the "Next" button to customize the name of the file to be generated. This is shown on right side in Figure 7.



C:\Users\Morad\Documents\Produit.owl

Y. Lv and C. Xie method

Applying of the design/transformation rules as dictated by Y. Lv and C. Xie in Y. Lv and C. Xie, "An ontology-based approach to build conceptual data model," 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, Sichuan, 2012, pp. 807-810.

Figure 7. New Onto2DB diagram, ontology and method choise

Figure 8 shows the result of the generation of a conceptual model of a RDB from the example ontology and choosing the method dictated by Y. Lv and C. Xie [17]. In addition to the generation of the conceptual model, our Onto2DB Plug-in gives us the hand to handle this model closely. Indeed, our plug-in provides us with the ability:

- To analyze the data,
- To verify the validity of the data,
- To filter the data,
- To select the useful data,
- To delete the uselessdata,
- To ensure the coherence of the data,
- To ensure the integrity of the data,

In addition, through the relational model, we are able to act on the model itself. Consequently, we can add or remove entities, attributes, and relationships. After customizing the generated conceptual model, the Plug-in allows us to generate the DDL SQL script whose execution produces the physical structure of the relational database within the chosen RDBMS. The generation of the DDL SQL script is illustrated in Figure 9. However, the result of the execution of this script is shown in Figure 10.

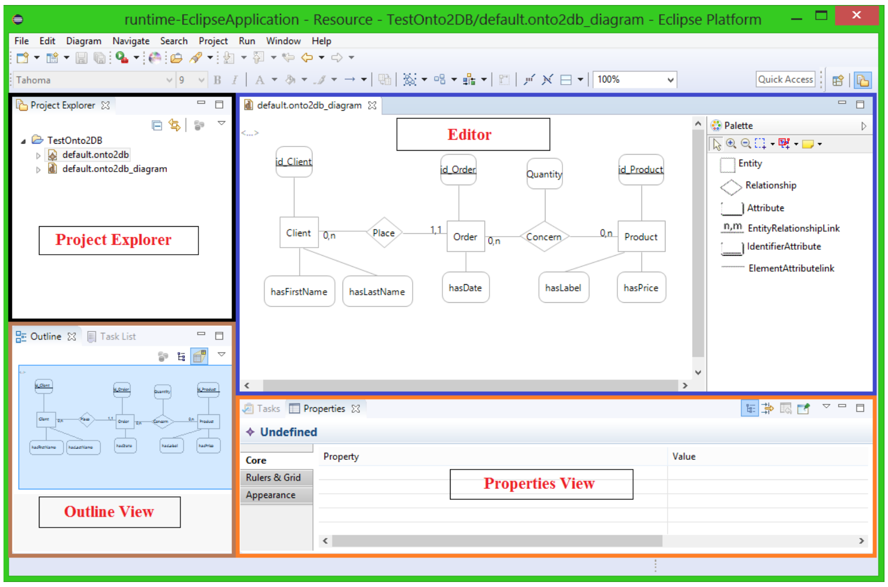


Figure 8. Onto2DB design preview

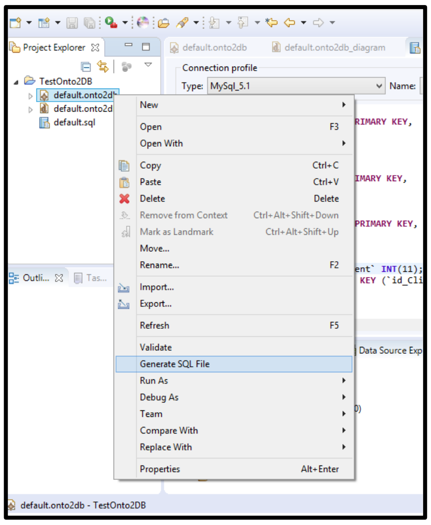
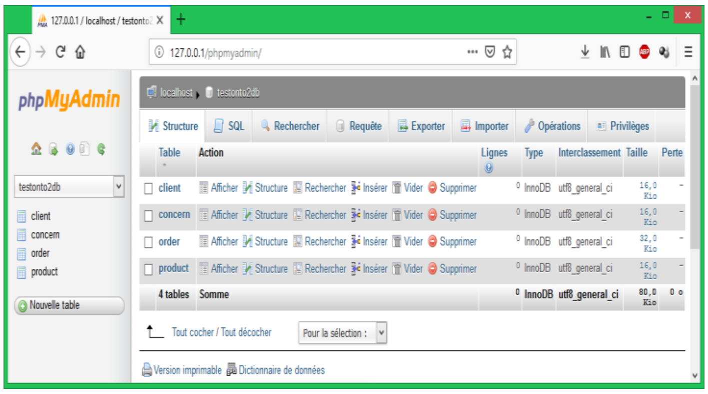


Figure 9. DDL SQL script generation



| | Table | Action | Lignes | Type | Interclassement | Taille | Perte |
|---|---|---|---|---|---|---|---|
| ☐ | client | Afficher Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8_general_ci | 16,0 Kio | - |
| ☐ | concern | Afficher Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8_general_ci | 16,0 Kio | - |
| ☐ | order | Afficher Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8_general_ci | 32,0 Kio | - |
| ☐ | product | Afficher Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8_general_ci | 16,0 Kio | - |
| | 4 tables | Somme | 0 | InnoDB | utf8_general_ci | 80,0 Kio | 0 o |

Figure 10. Resulting database

## 5. DISCUSSION OF RESULTS

The proliferation of Semantic Web projects over the modern era is impressive. The notion of ontology is the fundamental pillar of this discipline. As a result, ontologies are gradually invading the field of information systems. They are used for the standardization, structuring and formalization of the Web, Web Service, e-learning systems, etc. In fact, ontologies have become widely used in many areas and their number and size are growing.

However, the representation and persistence of data in today's information systems is dominated by the relational model and to a lesser degree by the object model. Despite this domination, the adoption of the ontological model as part of the representation of knowledge proves to be in perpetual growth. In this contribution, we provide a new approach for the development of a practical and exploitable tool developed according to the MDA approach in its version implemented by Eclipse. Thus, we touch on a relatively recent and innovative field by opening the way to the development of tools allowing the coexistence between the ontological databases and the relational databases.

For the validation of our contribution, we carried out an example of a representative case study for evaluating the functionalities of our developed Eclipse Plug-in. We admit that the example is simplistic, but we have seen the efficiency and the good functioning of this Plug-in. It fulfils the gap between the theory and practice of coupling an ontology with a relational database.

The results show that our approach allows to create an Eclipse Plug-in for the generation, customization and implementation of a relational conceptual model from an ontology. In fact, the generation of the conceptual model is automatic, the customization is carried out through the graphic edition and the generation of the DDL SQL script is automatic. In addition, our approach demonstrates the effectiveness of the tools developed by Eclipse as part of the Eclipse Modeling Project. In particular, the GMF sub-project allows the creation of graphic editors according to a highly developed level of abstraction based on the implementation of the MDA approach at the Eclipse level.

## 6. CONCLUSION

In this paper, we presented our approach for the development of Onto2DB an Eclipse Plug-in aimed at the automatic generation of a conceptual model of a relational database from an ontology with the possibility of editing this model. In this approach we have developed a DSL based on EMF offering a high level of abstraction. In addition, we reviewed the different concepts and tools needed for the implementation of this visual modeling tool based on the Eclipse GMF Framework.

Through this work, we have proven the feasibility of the automated generation of the conceptual model of a relational database from an ontology and thus filled our need for such a tool in order to implement our global approach that aims to produce decision making indicators from an ontology generated from a corpus of documents [4]. Additional work is needed to improve this plug-in by enforcing validation rules through Object Constraint Language (OCL) and implementing other relational database methods from an ontology. Another perspective is to develop another Plug-in aimed at generating the conceptual model of a Data Warehouse from a database.

## REFERENCES

[1] T. R. Gruber, “The Role of Common Ontology in Achieving Sharable Reusable Knowledge Bases,” *Proc. KR '91 Second Int'l Conf. Principles of Knowledge Representation and Reasoning*, pp. 601-602, 1991.

[2] C. Roussey, F. Pinet, M. A. Kang, and O. Corcho, “Ontologies for Interoperability,” in *Ontologies in Urban Development Projects*, London: Springer London, pp. 39-53, 2011.

[3] P. Cimiano and J. Volker, “Text2Onto–A Framework for Ontology Learning and Data driven Change Discovery,” *Natural language processing and information systems: 10th International Conference on Applications of Natural Language to Information Systems*, NLDB 2005, pp. 227-238, 2005.

[4] M. Hajji, M. Qbadou, and K. Mansouri, “Proposal for a new systemic approach of analitical processing of specific ontology to documentary resources case of educational documents,” *Journal of Theoretical and Applied Information Technology*, vol. 89, no. 2, pp. 481-488, 2016.

[5] M. Hajji, M. Qbadou, and K. Mansouri, “Toward Multi-Approach Model for Semi-Automating a Data Warehouse design from an Ontology,” *Trans. Mach. Learn. Artif. Intell.*, vol. 5, no. 4, 2017.

[6] A. Gali, C. Chen, K. Claypool and R. UcedaSosa, "From ontology to relational databases," *in Conceptual Modeling for Advanced Application Domains*, LNCS, vol. 3289, pp. 278-289, 2005.

[7] E. Vysniauskas, and L. Nemuraite, “Transforming ontology representation from OWL to relational database,” *Information Technology and Control*, vol. 35A, no. 3, pp. 333-343, 2006.

[8] I. Astrova, N. Korda and A. Kalja, “Storing OWL Ontologies in SQL Relational Databases,” *International Journal of Electrical, Computer and System Engineering*, 2007.

[9] C. Fankam, L. Bellatreche, H. Dehainsala, Y. Ait Ameur and G. Pierra, “SISRO : Conception de bases de données à partir d’ontologies de domaine,” *TSI Volume 28*, pages 1-29, 2009.

[10] J. Trinkunas and O. Vasilecas, “A Graph Oriented Model for Ontology Transformation into Conceptual Data Model,*” Information Technology and Control*, Vol. 36(1A), pp. 126-132, 2007.

[11] I. Astrova, N. Korda and A. Kalja, “Storing OWL ontologies in SQL Relational Databases,” *Engineering and Technology*, Vol. 23, pp. 167-172, 2007.

[12] E. Vysniauskas and L. Nemuraite, “Mapping of OWL ontology concepts to RDB schemas*,”* in *Information Technologies’ 2009, Proceedings of the 15th International Conference on Information and Software Technologies*; pp. 317-327, 2009.

[13] M. Mahfoudh and W. Jaziri, “Approche de couplage de BD et d’ontologie pour l'aide à la décision sémantique: contribution pour la satisfaction des requêtes SQL et SPARQL,” *Techniques et Sciences Informatiques, volume 32 of Hermes*, pages 863-889, 2013.

[14] S. R. Chaudhuri Patwari, A. Banerjee, S. Natarajan, S. Pandey, “A complementary domain specific design environment aiding SysML,” *2016 IEEE International Symposium on Systems Engineering (ISSE)*, pp. 1-8, 2016.

[15] C. Aniszczyk, “Using GEF with EMF,” *Eclipse.org*, 2019. [Online]. Available: http://www.eclipse.org/articles/Article-GEF-EMF/gef-emf.html. [Accessed: 10-Jan-2019].

[16] N. Richard C. Gronback, “Graphical Modeling Framework | The Eclipse Foundation,” *Eclipse.org*, 2019. [Online]. Available: http://www.eclipse.org/modeling/gmp/. [Accessed: 10-Jan-2019].

[17] Y. Lv and C. Xie, “An ontologybased approach to build conceptual data model,” *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, Sichuan, pp. 807-810, 2012.

[18] M. Horridge and S. Bechhofer, “The OWL API: A Java API for OWL ontologies,” *Semant. Web*, vol. 2, no. 1, pp. 11–21, 2011.

[19] I. Horrocks, “Ontologies and the semantic web,” Commun. ACM, vol. 51, no. 12, pp. 5867, 2008.

[20] J. Charlet, B. Bachimont & R. Troncy, “Ontologies Pour Le Web Semantique," In Le Web Semantique, Charlet J., Laublet P. & Reynaud C. (Ed.), Hors Serie De La Revue Information Interaction Intelligence (I3), 4(1), Cepadues, Toulouse, pp. 21-43, 2004.