

Mapping the Semantic Web to SQL Query to Extract Knowledge



Inaya Lahoud¹, Davy Monticolo², Vincent Hilaire¹

¹SET Laboratory, University of Technology of Belfort-Montbeliard
Belfort, France

²ERPI Laboratory, Polytechnic National Institute of Lorraine
Nancy, France

{Inaya.Lahoud@utbm.fr, Vincent.Hilaire@utbm.fr}, {Davy.Monticolo@ensgsi.inpl-nancy.fr}

ABSTRACT: *Among the goals of Knowledge Management (KM) we can cite the identification and capitalization of the know-how of companies in order to organize and disseminate them. KM is recognized as a nontrivial task. In the context of extended enterprises it is even more difficult because they are geographically distributed and use heterogeneous information systems. Indeed, information, from which knowledge is derived from, is stored in different databases, distributed in different sites across the entire network of the extended enterprise. This paper proposes a semi-automatic approach for knowledge extraction from several databases. This knowledge will be stored in an organizational memory (OM). The approach is based upon the definition of ontologies for knowledge exchange and Model Driven Engineering concepts such as meta-models and transformation to process requests.*

Keywords: Ontology, Semantic Web Service, Knowledge Management, Model Driven Engineering, Interoperability, Mapping

Received: 23 January 2012, Revised 19 March 2012, Accepted 18 March 2012

© 2012 DLINE. All rights reserved

1. Introduction

The process of designing manufactured products (called products from now on) frequently requires the use of several heterogeneous business tools (calculation tool, CAD tool, a tool for production management, PLM, PDM, etc.). Each of these tools deals with specific aspects of product design such as: CAD, project management, resource planning and PLM. In the nowadays context, these tools are distributed on different sites across the entire network of the extended enterprise.

Moreover, to survive in an increasingly competitive business environment, manufacturing enterprises are under unprecedented pressure to become leaner and more agile. The product range must be updated permanently and production costs the lowest possible. Product leadership companies must continue to enter new market with innovative products. This requires optimizing process and methodologies used by engineering department. The design process has to be rationalized in capitalizing knowledge, know-how and technological patrimony. A solution to this problem consists in using Knowledge Management techniques.

The goal of this paper is to propose an approach that allows knowledge extraction for the actors of the extended virtual enterprise. The underlying idea relies on ontologies created by experts that specify the concepts and relationships that conceptualize the specific knowledge of a business tool. These ontologies are then used as a mean to create requests that are

executed by web services to query business tools.

Model-Driven Engineering (MDE) techniques are used to transform ontologies into query languages. This approach provides a great modularity and reusability of models.

The paper is organized as follows: After a presentation of some works on MDE and ontologies in section 2, the section 3 introduces our knowledge management cycle which is the heart of our approach. Our method of extracting knowledge is presented and explained in section 4. Eventually, we give an example of our approach and an overview of technologies and solutions our approach contributes to.

2. Related Work

2.1 Model-Driven Engineering and Knowledge Management

Designing a product requires the collaboration between various business actors and among different countries in certain cases. This problem of collaboration was the studied in several works. One solution consists in having a synchronization methodology for the interoperability between business tools based on the principle of Model Driven Engineering (MDE) [9].

Model Driven Architecture (MDA) [17] (which is the first effective proposition of the MDE defined by the object Management groups [18]) and ontologies are two different approaches developed by different communities. However, they have common points and issues. There are many researches today who analyze these points and each one present a new method on how we can use these two technologies together.

Roser & Bauer [19] presented an approach of ontology-based model transformation to achieve interoperability in modeling enterprises and application systems by semantic enrichment, thus, the MDA can be realized more efficient by extending specifications of models and model transformations with ontologies. While Alakwaa & Salah [1] present an automatic generation of Content Analysis Model from the Ontology Model by making a straightforward mapping between their Metamodels.

To achieve the targets of MDE, Goknil [10] used the ontology technical space in model transformations. This will enable them to model not only the Meta-concepts but also the semantic context which can be used in model inferencing. Within this context, they define Meta-models of object oriented models ontologically. Another research made by Cao [6] shows us how to do semantic aggregations of semantic relationships and ontologies in one or multiple domains, and the ontological transformation from one domain to another.

2.2 Semantic Web Services

Our approach of KM is based on the use of semantic web services. Semantic web services (WSS) play an important role in the automation of applications based on web services thanks to the contribution of semantics. This approach was the subject of several research works.

In the field of knowledge management, semantic web service approach is used by Brunel University [23] for designing applications for web businesses that provides a convenient and scalable way for sharing knowledge. They propose a Semantic Web Service-oriented model, where resources and services are described by an ontology [11], and processed through Semantic Web Services, enabling an integrated administration, interoperability and automated reasoning. In the same context, Nemirovskij [16] applied this approach to manage knowledge in the field of education. This work aimed to model a platform for the common area of education based on Semantic Web Services Approach and a Framework for SOA-deployment and treatment. We can also cite the work of Che Cob [7] who proposed a structure of the semantic web service ontology based for a knowledge management system. His framework consisted of four main layers: User Interaction, Interface, Mediator, and Ontology. We present in the next section our overall approach of knowledge management.

3. Knowledge Management Cycle

Maier [12] presented a Knowledge Management System (KMS) that supports the functions of knowledge creation, construction, identification, capturing, acquisition, selection, valuation, organization, linking, structuring, formalization, visualization, distribution, retention, maintenance, refinement, evolution, accessing, search, and application. This KMS uses a variety of technologies designed to enhance knowledge storage and knowledge communication/transfer.

Grundstein proposed a knowledge management life-cycle, where, according to him, *“in any operation of knowledge capitalization, it is important to identify the strategic knowledge to be capitalized”*. His cycle is divided into four facets which

are: identify, formalize, value and update [15].

We will present in this section our knowledge management system which is based on these two systems to fit with our objective.

We created our cycle of knowledge capitalization from four facets which are: Definition, extraction, validation and update/ reuse. In the first phase, “*definition of knowledge*”, the business expert identifies the knowledge needed for the project by creating a light ontology (LO). We define LO as a set of concepts and relationships.

We chose this kind of ontology firstly because it is relatively easy to construct. Secondly, business experts need just to define the concepts and the relationships that match their needs, thus, it is not necessary for a complete ontology with axioms and inference machinery.

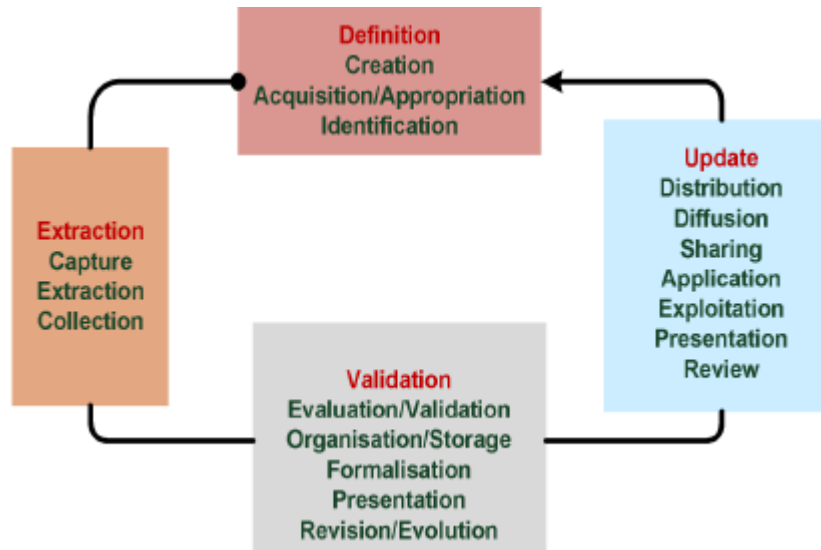
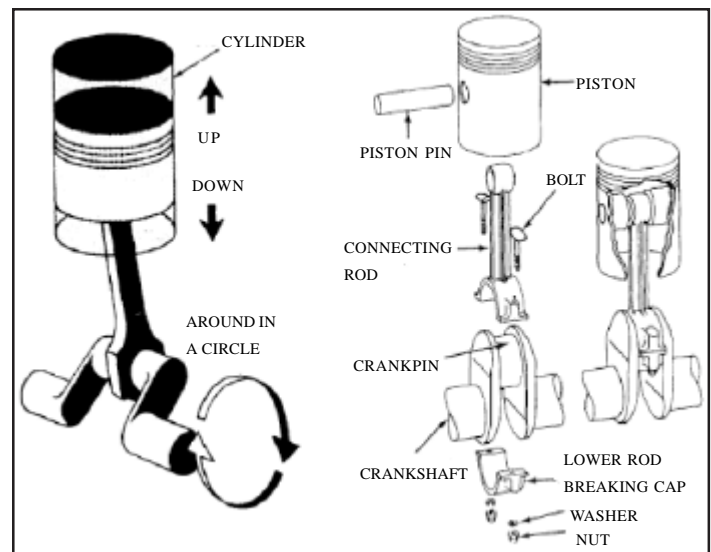


Figure 1. KM cycle



Bike . Ultimate CF SLX 9.0 SL



Car Engine

We give now an example of knowledge identification. Let's suppose that a business expert wants to create two projects. The first one consists to design a bicycle Ultimate CF SLX 9.0 SL and the second one to design a car engine. Thus, He creates in OCEAN system a LO describing the Meta-model of bicycle and a second LO for the engine as shown in the figure 7.

In this example business experts define LO of each project and he wants to get as result value of the parameters defined in this LO as seat tube length, top tube length, etc. for Bicycle project and cubic inch displacement, Diameter crank pin, distance between centre lines, etc. for car engine project.

In the second step, “*extracting knowledge*”, the LO created will be sent to the SWS responsible of extraction to apply it on corresponding business applications. Thus, each LO is translated to query by applying several rules, which are detailed in section 6. The transformation from a LO model to a query model will be made by MDE transformations described in section 5 and 6.

Data resulting from the process of extraction are then annotated. Annotating data means adding informations, details, comments and other semantics concerning data in order to contextualize it and transform it into Knowledge.

In the third phase, “*validation knowledge*”, the knowledge is already stored in an OM. This OM is accessible to business actors under the form of a semantic wiki. The semantic wiki allows: the structuration of knowledge, his diffusion and the obtention of feed-back from business actors. Indeed, each actor can confirm, reject, or modify Knowledge using the semantic wiki. If some knowledge is rejected by the majority of actors then it will be destroyed and will no more appear in our knowledge base.

In the fourth phase, “*update/ reuse knowledge*”, any business actor can consult or search knowledge in the OM.

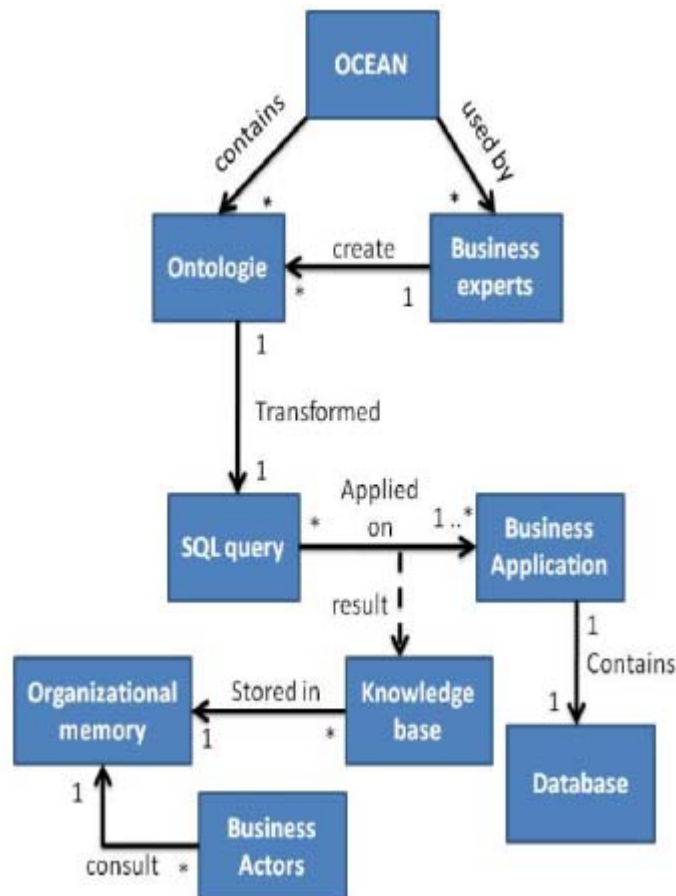


Figure 2. Architecture of OCEAN [14]

EADS. They have different business areas and wish to apply the same methodology for multi-sources knowledge management. The carrier of the project is the industrial society DPS (Digital Product Simulation).

4.1 Knowledge Extraction Approach

Our main objective is to design and develop a knowledge management system for the extended virtual enterprise that allows extracting knowledge by using MDE techniques and semantic web services. The overall approach of this system can be summarized by the figure above (Figure 2).

- 1) A business expert builds its own ontologies.
- 2) The ontologies are then transformed into queries using MDE transformations. These queries are executed through semantic web services. The SWS can then extract data from several databases (figure 3).
- 3) Data are semantically annotated and become knowledge which is stored in an OM
- 4) Business actors can consult this OM and obtain knowledge through searches.

So figure 3 explains our approach of transformation from ontology to query. We present here 3 cases.

Case 1: If we have ontology “O1” and this one describes concepts from the PLM business application (step 1), so O1 will be transformed into SQL query syntax (step 2). This query will be taken by web service (step 3) which will connect to the database

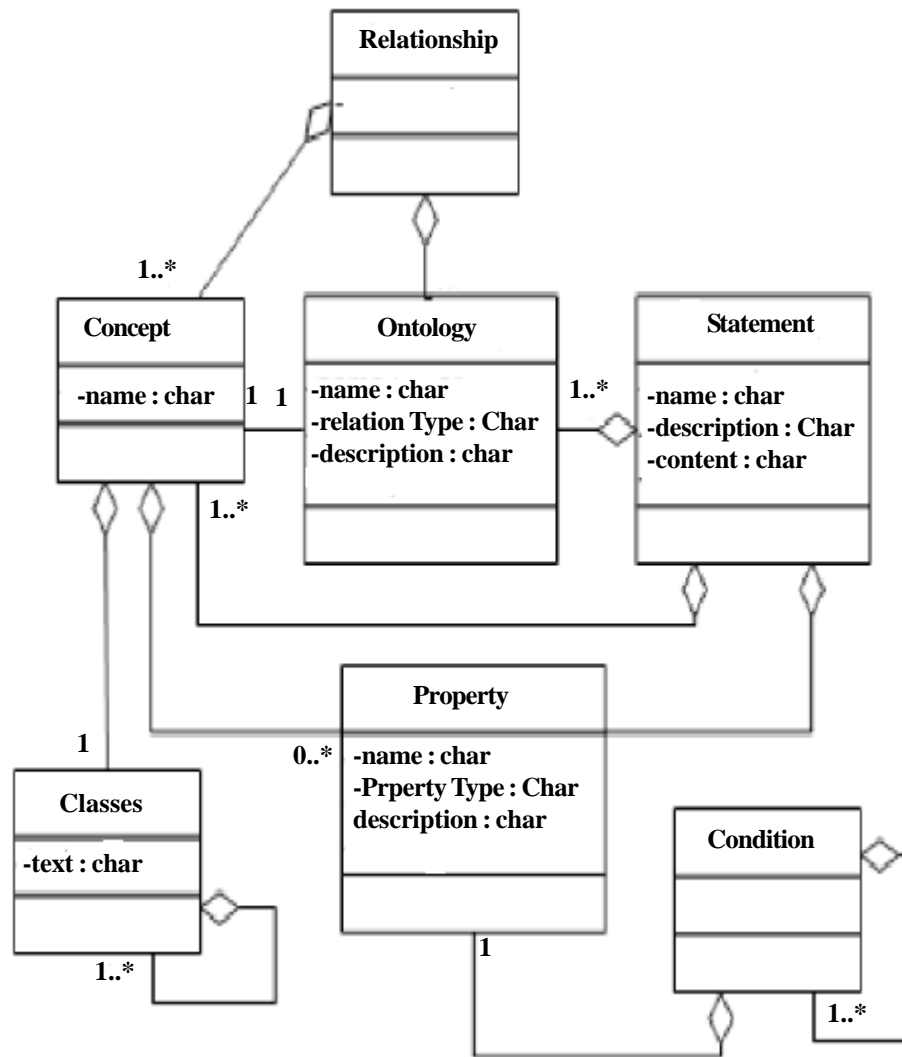


Figure 4. Simplified ontological metamodel

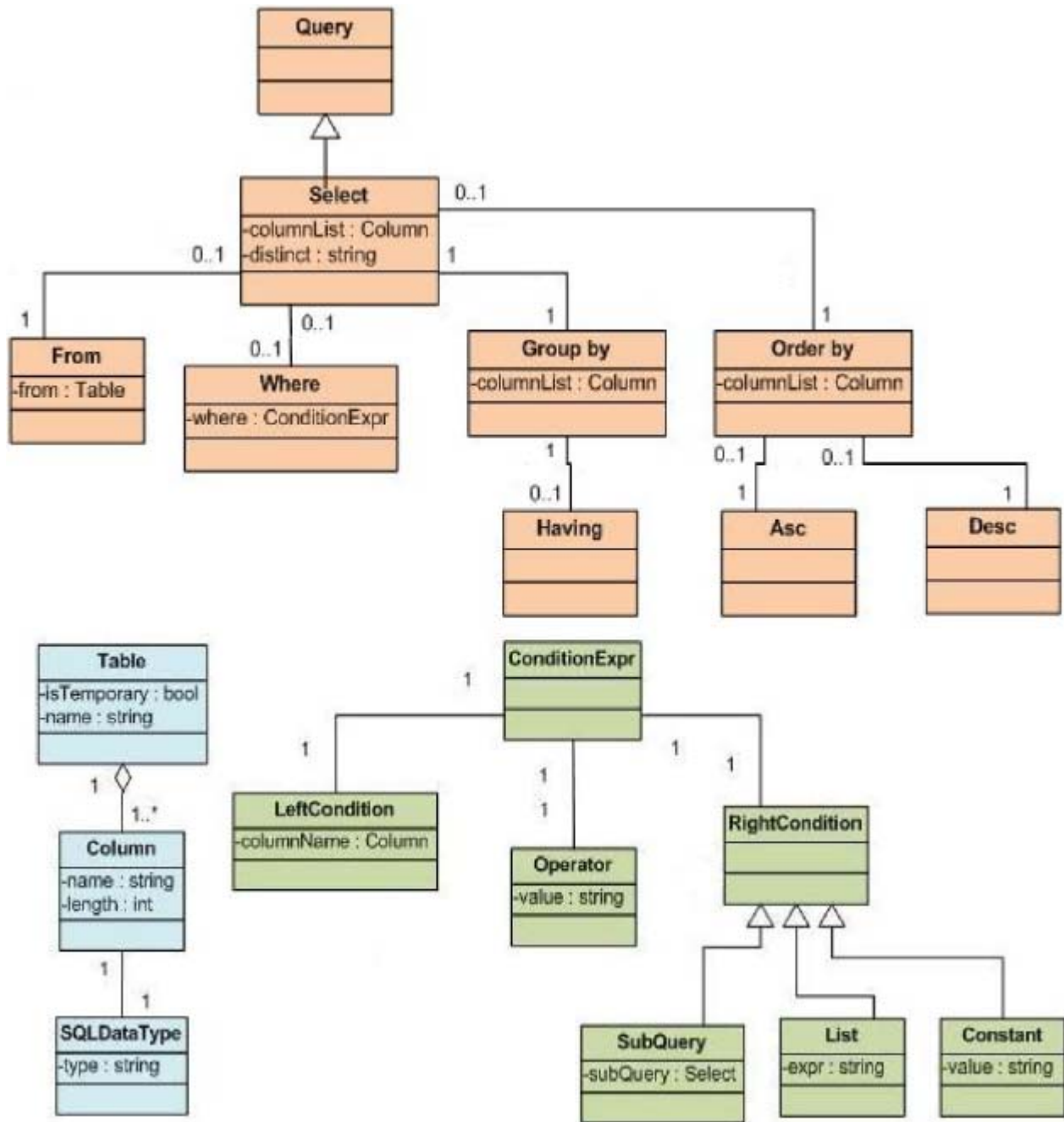


Figure 5. Simplified SQL query metamodel

of PLM business application (step 4) and execute it. The result of the execution is a set of data. These data (step 5) will be annotated to become information and stocked in OM.

Case 2: same steps but this times the ontology O2 will be transformed into excel query (step 2) and the connection will be with the CAD business application (step 4).

Case 3: same steps but O3 will be transformed into xml query (step 2) and the web service will connect with the database of management tool business application (step 4).

We already mentioned that knowledge is distributed in different databases on the entire network. Our system has thus to query several databases. These queries are generated from the different ontologies describing business tools and transformations described in the next subsection.

4.2 Methodology of mapping the Semantic Web (Ontology Model) to SQL Model

In order to obtain information from the different business tools, the ontologies created by business experts are transformed into queries.

```

Select [distinct]{list of columns}
From {tables}
[Where conditions]
[Group by {list of columns} [having condition]]
[Order by {list of columns} [asc|desc]]
    
```

To define such transformations we need first a meta-model for our ontologies and a meta-model for the used query language. Once these meta-models are defined, respectively source and destination, we can define a transformation that takes as input an instance of the ontology meta-model and produces an instance of query for the database meta-model.

Our LO is composed of many concepts which are linked by relationships. Concepts and relationships are both a sort of statement as properties. Concepts are a group of classes and subclasses. Each concept can have 0 or more properties which can contain conditions, as shown in Figure 4.

Similarly, a relational database is considered to be the implementation of a SQL query metamodel (figure 5). This model includes

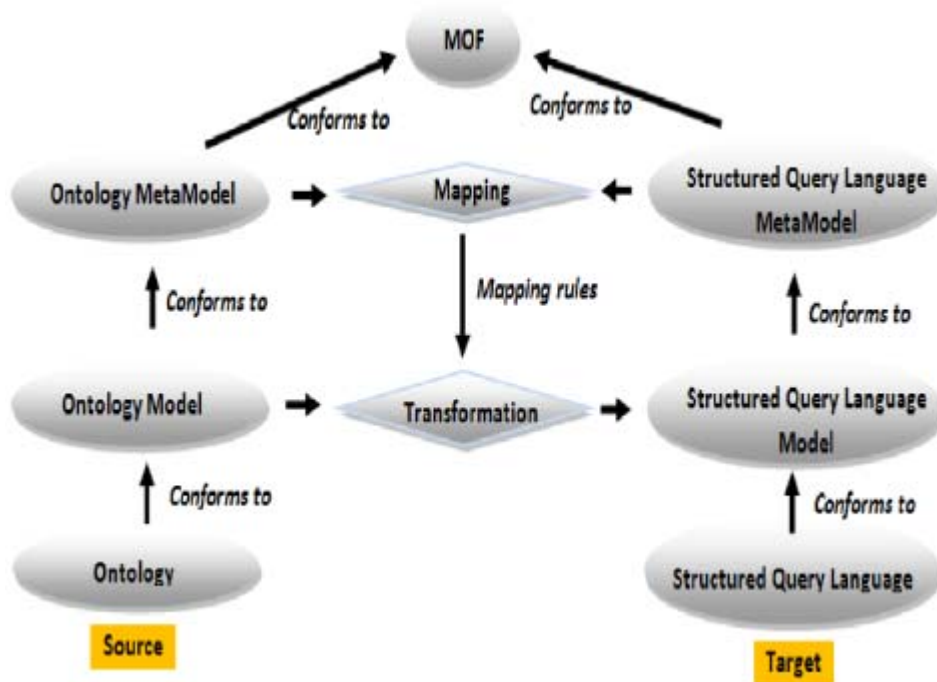


Figure 6. Simplified SQL query metamodel

constructs for specifying select, from, where, group by, order by, tables, columns, and conditions.

We have defined our sql query metamodel by using the sql query syntax:

We can define the condition syntax by three elements:

- The left side of the condition
- The right side of the condition
- The operator that links the 2 parts

The operator can be: =, !=, >, >=, <, <=, between, and, in (list), like, is null ...

The right side of the condition can be: a column, constant, list, or select sub query.

Figure 6 shows the basic idea behind our approach. It represents the transformation between the source model and the target model. Thus, transformation of ontologies to SQL queries is based on a set of rules called mapping rules that specify how to map constructs of the ontological metamodel to the SQL metamodel.

4.3 Mapping Rules

There is a lot of research today which talk about extracting ontologies from relational databases as [2, 4, 5, 8, 13, 20, 21, 22], but as far as I know, there is no research in the reverse order except the article of Astrova [3], which transforms an ontology to relational database. Thus we based on her work to create our own rules for converting an ontology to SQL query.

Our approach maps an ontology to a SQL query, applying the following rules:

R1 The name of the ontology is the name of a project

R2 A class is a condition on the name of product or element of product

R3 If a class has one or more DatatypeProperty so we the condition is ParametersName in ('DatatypeProperty1', 'DatatypeProperty2', 'DatatypeProperty3' ...)

R4 Forming conditions: We look at the range of each DatatypeProperty

- If it is an attribute positiveInteger then the condition is check > 0
- If it is a DataRange which includes a list so check attribute in [value1,value2 ...]
- If there are restrictions on DatatypeProperty and it is not a cardinality restriction as HasValue so the condition is: check attribute = value
- Inverse functional property is the "Distinct" constraint
- Required property mean check fields not null (not 0 or not = "")

R5 Store conditions in the list X.

R6 We have to mention that we have a table "Matching table" which stores all tables, columns of all databases and the equivalent variable. For instance, variable '\$ParametersName' corresponds to column 'nom_variable' in the table 'variables' in the database 'ACSP', the same variable '\$ParametersName' corresponds also to column 'nom_parametre' in the table 'parametres' in the database 'KrossRoad'.

As we have in a table above the list of tables, we can search the tables of the database on which we applied our ontology and store them in the list Y.

R7 Search the primary and foreign keys of the tables in the list (the relations between them) and store these relations in a table.

R8 We search all times the values of the following variables: \$ParametersName, \$ParametersValue, \$ProjectName, \$ProjectDescription, \$ProjectStartDay, \$Project-EndDay, \$PersonName, \$PersonRole. So, we search in the "Matching table" the equivalence of these variables in the database we work on and store them in the list Z.

R9 Build the query:

- **Select** the columns in the list Z proceeded by the first letter of the domain table. E.g.: p.nom, And followed by a comma
- **From** all tables stored in the list Y followed by the first letter as naming the table by a letter
- **Where** all the relationship stored in a table to make the connections between the tables + all conditions sored in the list X separated by "and".

R10 we repeat these rules for each class int the ontology who have datatypeproperty and we do "Union" between these queriesto get one result at the end.

- ▶ Ultimate CF SLX 9.0 SL (Bike)
 - ▶ Brakes
 - ▶ Chainrings
 - ▶ Canyon One Four SLX (Fork)
 - ▶ Frame
 - ▶ Seat tube length
 - ▶ Top tube length
 - ▶ Head tube length
 - ▶ Head tube angle
 - ▶ Seat tube angle

- ▶ Chainstay length
- ▶ Wheel base
- ▶ Stack
- ▶ Reach
- ▶ Handlebar
 - ▶ Weight
 - ▶ Width
 - ▶ type
 - ▶ Material
- ▶ Headset
- ▶ Wheel
 - ▶ Hub
 - ▶ Axle
 - ▶ Bearings
 - ▶ Hub shell
 - ▶ Hub brakes
 - ▶ Gears
 - ▶ Mavic R-Sys SL (Rims)
 - ▶ Size
 - ▶ Weight
 - ▶ Spokes
 - ▶ Material
 - ▶ Number of spokes
 - ▶ Nipples
- ▶ Selle Italia SLR (Saddle)
- ▶ Rear Derailleur
- ▶ Front Derailleur
- ▶ Pedals

- ▶ Car Engine
 - ▶ Power train
 - ▶ Connecting rod
 - ▶ Cubic inch displacement
 - ▶ Diameter crankpin
 - ▶ Distance between centre lines
 - ▶ Thickness wings
 - ▶ Thickness rod
 - ▶ Width crankpin
 - ▶ Mass connecting rod
 - ▶ Workshop engine

- ▶ Bore
- ▶ Height combustion chamber
- ▶ Height cylinder
- ▶ Lambda
- ▶ Width crankpin
- ▶ Cylinder Pressure
- ▶ Power
- ▶ Volume up room
- ▶ Crankshaft
 - ▶ Half race
 - ▶ Diameter crankpin
 - ▶ Width crankpin
 - ▶ Diameter trunnion
 - ▶ Shifting center counterweight
- ▶ Piston
 - ▶ Outside diameter hole
 - ▶ Inner diameter hole
 - ▶ Diameter piston
 - ▶ Thickness Face Up Piston
 - ▶ Segment spacing
 - ▶ High compression
 - ▶ Piston height
 - ▶ Segment number

Figure 7a. An extract of the bike ontology

Figure 7b. an extract of the engine ontology

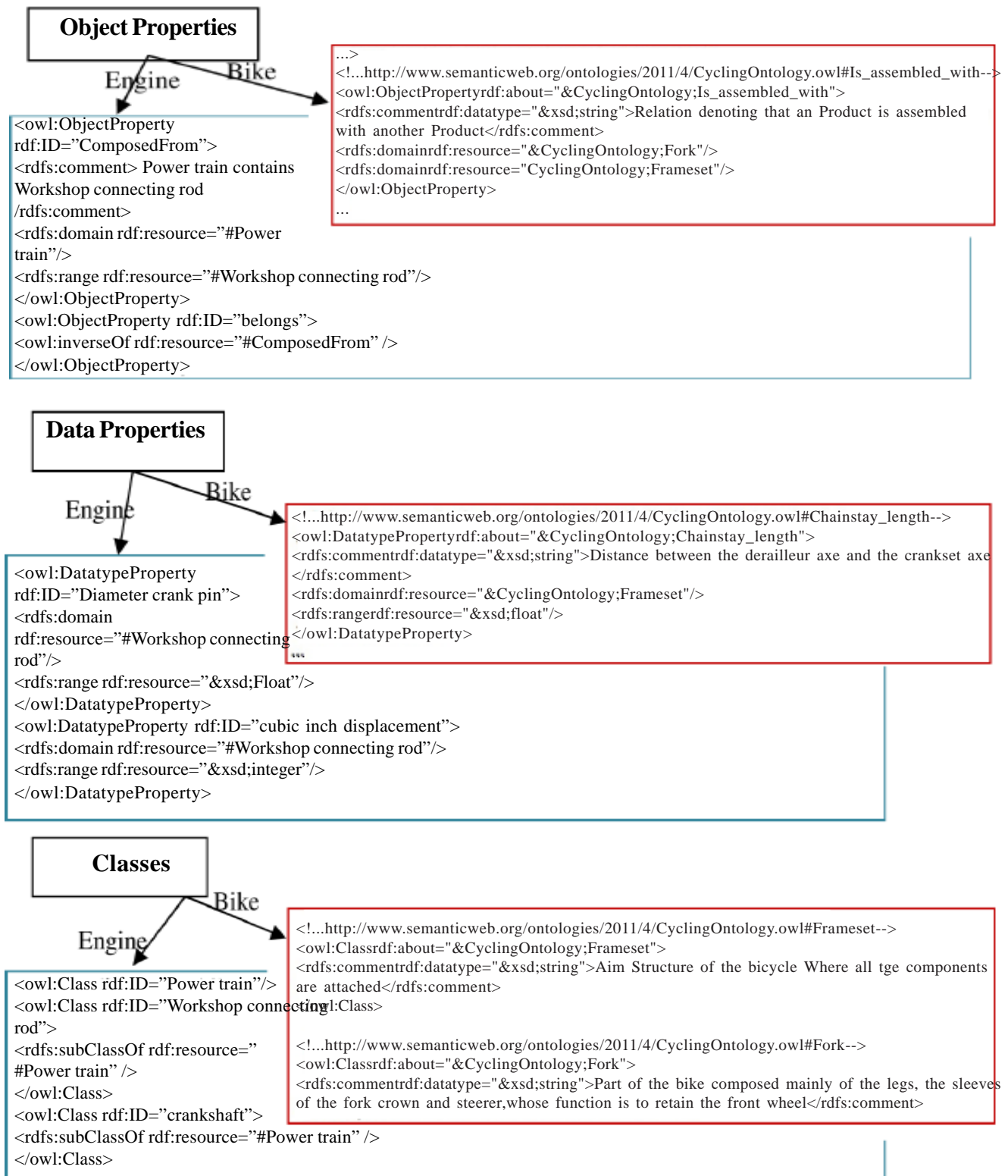


Figure 8. An extract of knowledge in OWL file

4.4 Example of Ontologies

We will present in this section an example of two ontologies created by business expert where he defined his needs. Figure 7

```

select v.nom_variable, v.valeur_variable, pc.nom, pc.description, pc.datedebut, pc.datefin,
pp.nom, pp.role
from variables v, produit_element pe, produit p, projet pc, personne pp
where v.id_produit = pe.n_produit
and v.id_element = pe.n_element
and pe.n_produit = p.id_produit
and p.id_projet = pc.id_projet
and pc.id_personne = pp.id_personne
and v.nom_variable in ('size', 'weight')
and pe.nom_pe like '% Mavic R-Sys SL%'
and p.nom_produit like '%ultimate%'
and pc.id_projet like '%bike%'
UNION
select ...
from ...
where ...
and v.nom_variable in ('Weight', 'Width', 'type', 'material')

and pe.nom_pe like '% Handlebar%'
and p.nom_produit like '%ultimate%'
and pc.id_projet like '%bike%'
UNION ...

```

Figure 9. Extract of the generated SQL query

```

<rdf:description rdf:about="https://acsp.utbm.fr/Bicycle_Ultimate CF SLX 9.0 SL -50cm">
<BicycleOnto:PersonRole>Concepteur</BicycleOnto:PersonRole>
<BicycleOnto:CreatedBy>Iahoud</BicycleOnto: CreatedBy>
<BicycleOnto: ProjectStartDay>23/10/2006</BicycleOnto:ProjectStartDay>
<BicycleOnto: ProjectEndDay>08/01/2007</BicycleOnto:ProjectEndDay>
<BicycleOnto:Maturity>10</BicycleOnto:Maturity>
<BicycleOnto: EvaluationPourcent>25</BicycleOnto:EvaluationPourcent>
<BicycleOnto:ProductName> Ultimate CF SLX 9.0 SL </BicycleOnto:ProductName />
<BicycleOnto:ProductDescription> The Speedmax CF 9.0 is a top of the line aero bike. Manufactured by Canyon, the
Speedman 9.0 was conceived to give triathletes the supreme cycling experience when running against the clock.
Founded in 1985 Canyon has over 25 years of craftsmanship and engineering experience and was able to translate
this experience into the Speedmax CF 9.0 frame.</BicycleOnto:ProductDescription />
<BicycleOnto:SeatTubeLength>550</BicycleOnto:SeatTubeLength />
<BicycleOnto:TopTubeLength>570</BicycleOnto:TopTubeLength />
<BicycleOnto:HeadTubeLength>125</BicycleOnto:HeadTubeLength />
<BicycleOnto:HandlebarName>Alu X</BicycleOnto: HandlebarName/>
<BicycleOnto:HandlebarWeight>927g< BicycleOnto: HandlebarWeight/>
<BicycleOnto:HandlebarMaterial>Aluminium< BicycleOnto: HandlebarMaterial/>
</rdf:description>

```

Figure 10. Example of knowledge annotated in RDF file

presents an extract of bike and engine ontology. These ontologies define a vocabulary and a semantic to structure, organize, and detail:

- all the characteristic of a bike or engine,

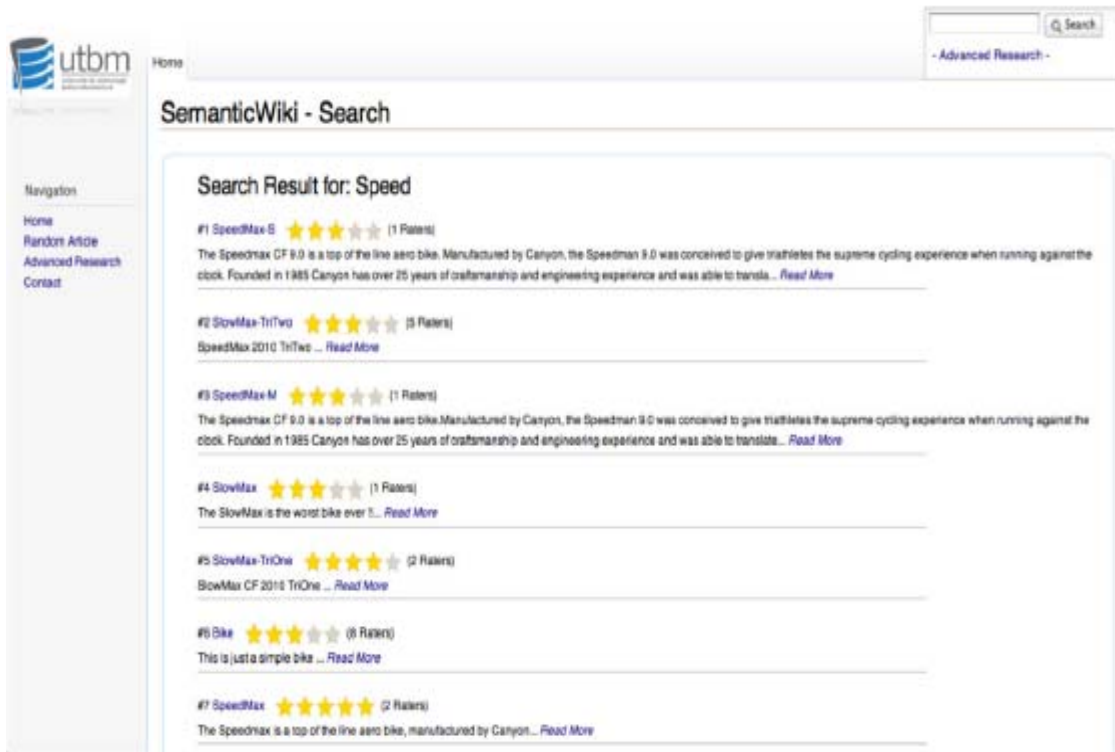


Figure 11. The search results for the query

- all the roles of the professional actors in a development project of a new bike or engine,
- all the processes used to develop and industrialize a bicycle or engine.

By considering the bike ontology (figure 7a), business expert can add in his ontology general concepts. He can add also instances of concepts in order to filter the results of the ontology. So, instances in this ontology play the role of conditions. For instance, in the bike ontology business expert wants to know the right values of parameters to design a new bike. But he does not want any bike, so he specifies his search by indicating the instance “*Ultimate CF SLX 9.0 SL*” as a name of the bike. In addition, he specifies the instance “*Canyon One Four SLX*” which means that the bike must have this name as fork, also a “*Mavic R-Sys SL*” as rims and “*Selle Italia SLR*” as saddle. Thus these conditions can be considered as filter of the result as expected. On the other side, he can create general concepts in his ontology as the concept “*brakes*”, “*chainrings*”, “*frame*”, etc. without specifying conditions on these concepts.

In the proposed ontology there are relations between existing concepts. For example, the relation “*is_stopped_by*” links the instance “*Ultimate CF SLX 9.0 SL*” to the concept “*brakes*”, the relation “*is_composed_by*” links the instance “*Ultimate CF SLX 9.0 SL*” to the concept “*frame*”, the relation “*is_fitted_into*” links the concept “*wheel*” to the concept “*frame*”, etc.

If we take the second example (figure 7b) where the business expert expects to design an engine, we can see that he defines all the concepts in his ontology as general concepts. There is no instance in this ontology which means that business expert expects to get as result the parameters with their values of every kind of engine existing in the database.

These ontologies in the figure 7 are saved as owl format as shown in the figure 8. This figure shows an extract of OWL files where there are knowledge defined by the business expert.

In the rest of this paper we will detail the example of the bike ontology. As we can see in the figure 9, this ontology was transformed into an SQL query which will return information from business application.

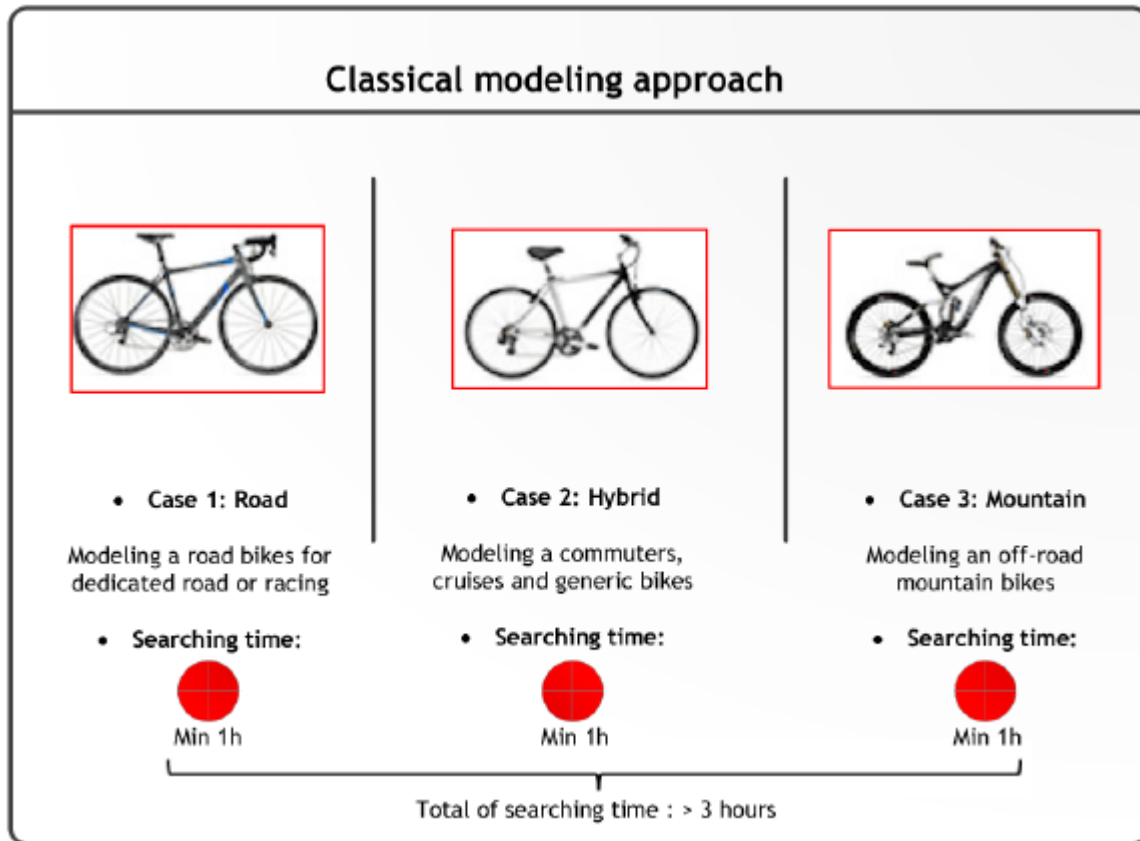


Figure 12. Classical modeling approach

The information returned with this query becomes knowledge and stored in RDF files as shown in the figure 10. The results for the query are given in the figure 11 as search results.

These RDF files are stored in OM. This one will be used later in the semantic wiki to search projects, and to evaluate, modify or delete knowledge of the project searched. In the example below we can see that user search projects who talk about speed, so he get as result a list of projects listed by the most important for the user depending by his role. Furthermore, user can select a project from the list and :

- consult all knowledge included in the project with the good values.
- evaluate these knowledge by putting a note
- modify these knowledge

If the global note of the knowledge is less than 20%, this knowledge will be considered as bad knowledge and deleted from the project.

5. Experimentation Results

In the classical modeling approach (figure 12), a designer had to design each type of bike with correct parameters. To search correct values for each parameter to design his bike he needs more than one hour and the more the information is distributed on different sites across the entire network of the extended enterprise, the more the search is difficult.

Unlike the classical modeling approach we present here a new method to model different types of bike in a short time (figure 13). Thus, business expert create generic bike ontology. This LO contains concepts describing the bike and the relations between them.

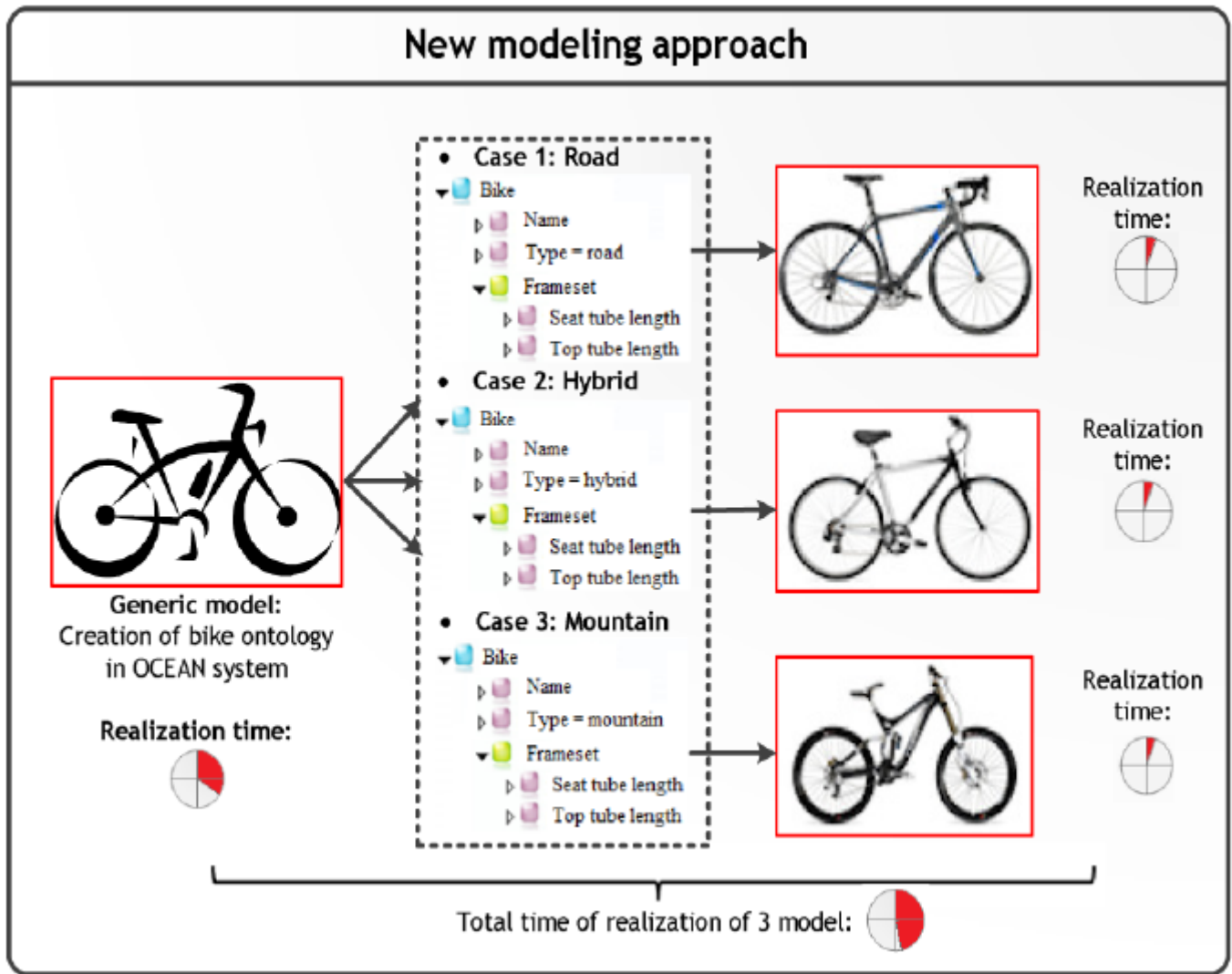


Figure 13. New modeling approach

Figure 13 presents a generic ontology and 3 cases of bike ontology. For instance, in case 1 business expert wants to design a road bike. To achieve this goal he use the generic bike ontology and define “*type = road*” as shown in the figure below. In this way we reduce the time of realization of different type of bikes more than 60%.

We have made our experiments by requesting three existing databases of collaborative platforms which manage mechanical projects. OCEAN allows transforming the bike ontology and the connection with the databases.

According to our tests, we saw that there is some loss of semantic concepts during the transformation as we realize it between two different models: ontology and SQL. The first is related to knowledge that involves not only syntax but also semantics and inferences, and the second is a language related to databases.

We firstly recognized that we have a loss of structure because not all concepts of ontology can have equivalences in SQL and therefore the quality of transformation should be analyzed. Second, we have done until now a certain number of transformation rules and we will continue to find more. Hence, we can conclude that we have really a loss of semantic if the ontology created contains concepts that are not taken into account in the transformation rules.

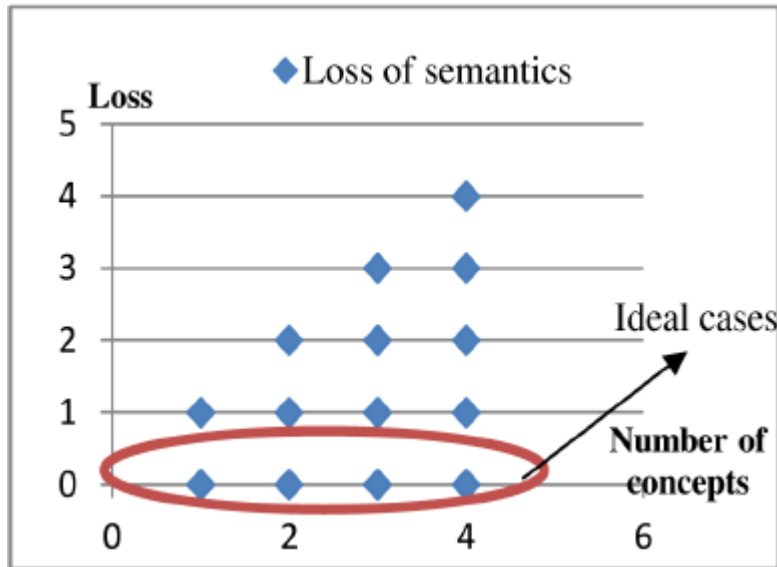


Figure 14. Loss of semantics in our transformation models

Figure 14 shows the loss of semantics that we can have on a sample of four concepts. For example, we have four concepts in the ontology and which have equivalences in SQL managed by our rules. If the ontology created by the user includes four concepts and all these concepts have equivalent then there is no loss. Else, if two of the four concepts have equivalences and two not so we lost two semantic and so on.

So the ideal is to have an ontology that includes equivalent concepts to avoid losses. For these reasons we limit the user at the moment to create his ontology in OCEAN using just the concepts managed by our rules and in this way we neglect the losses. At the same time we continue our research on the transformation rules to allow user to use all the concepts of the ontology.

6. Conclusion & Perspectives

The goal of this paper was to contribute to an heterogeneous and distributed knowledge management system which will uses formal ontologies to extract information from business applications and annotate that information to build a knowledge base in RDF format. But to reach this goal we are faced to a problem of transformation from ontology model to SQL model as described in this paper. That's why we explained why we need to do a transformation between these two models, how we did it and the results of our tests.

Furthermore, we are limited today to a certain number of relationships that do not take into account all the complexity of the specifications of business knowledge, but it will be necessary to give more opportunities for business experts to describe the relationships between the concepts of their domains of knowledge which will be our researches in the future..

7. Acknowledgments

We thank our industry partners who regularly test our prototypes and guide our advancements in specifying their industrial needs. We particularly thank the project teams of the company Faurecia who have accepted to work on formalizing their knowledge areas using the Web Service Semantics OCEAN.

References

- [1] Alakwaa, W., Salah, A. (2010). Model Transformation from Ontology Model to Content Analysis Model. *International Journal of Computer Applications* 7, 5- 12
- [2] Arens, A., Chee, C.Y., Hsu, C., & Knoblock, C. (1993). Retrieving and Integrating Data from Multiple Information Sources. *International Journal on Intelligent and Cooperative Information Systems* 2 (2) 127-158.

- [3] Astrova, I., Korda, N., Kalja, A. (2007). Storing OWL Ontologies in SQL Relational Databases. *International Journal of Electrical, Computer, and Systems Engineering* 1, 242-247
- [4] Barrasa, J., Gomez-Perez, A. (2006). Upgrading relational legacy data to the semantic web. *In: Proceedings of 15th International World Wide Web Conference*. ACM. 1069-1070
- [5] Buccella, A., Penabad, M. R., Rodríguez, F. J., Fariña, A., Cechich, A. (2004). From Relational Databases to OWL Ontologies. *In: Proceedings of the 6th Russian Conference on Digital Libraries*. Pushchino, Rusia.
- [6] Cao, L., Luo, D., Luo, C., Liu, L. (2004). Ontology Transformation in Multiple Domains. *In: Australian Conference on Artificial Intelligence*, 985-990. Australia.
- [7] Che Cob, Z., Abdullah, R. (2008). Ontology-based Semantic Web Services Framework for Knowledge Management System. *In: International Symposium on Information Technology*, 1-8. Malaysia.
- [8] Chen, H., Wang, Y., Wang, H., Mao, Y., Tang, J., Zhou, C., et al. (2006). Towards a Semantic Web of Relational Databases: a Practical Semantic Toolkit and an In-Use Case from Traditional Chinese Medicine. *In: Proceedings of 5th International Semantic Web Conference*. Athens, GA, USA.
- [9] Chettaoui, H., Noel, F. (2010). Synchronisation of heterogeneous design models of manufactured products by the formulation of business knowledge. *J. International Journal of Computer Applications in Technology*, 37, 230-243.
- [10] Goknil, A., Topaloglu, N. Y. (2005). Ontological perspective in metamodeling for model transformations. *In: 2005 symposia on Metainformatics*, 73-84. Denmark.
- [11] Gruber, T. (1993). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies* 43, 907-928
- [12] Grundstein, M., Barthès, J.-P. (1996). An industrial view of the process of capitalizing knowledge. *In: 4th International Symposium on the Management of Industrial and Corporate Knowledge*, 258-264. Rotterdam.
- [13] Sequeda Juan, F., Syed Hamid Tirmizi, Oscar Corcho., Daniel P. Miranker (2011). Survey of directly mapping SQL databases to the Semantic Web. *The Knowledge Engineering Review*, 26, 445-486
- [14] Lahoud, I., Monticolo, D., Gomes, S. (2010). OCEAN: A Semantic Web Service to Extract Knowledge in E-Groupwares. *In: 6th International Conference on Signal- Image Technology and Internet-Based Systems*, 354-362. Kuala Lumpur.
- [15] Maier, R. (2002). Knowledge management systems. Information and communication technologies for knowledge management. Springer-Verlag, Berlin.
- [16] Nemirovskij, G., Wolters, M., Heuel, E. (2008). Distributed Study: a Semantic Web Services Approach for Modelling a Common Educational Space. *In: World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 2007-2014. Chesapeake.
- [17] Object Management Group, <http://www.omg.org/docs/omg/03-06-01.pdf>
- [18] Object Management Group, <http://www.omg.org/>
- [19] Roser, S., Bauer, B. (2006). Ontology-based Model Transformation. *In: Bruel, J.- M. (eds.) Satellite Events at the MoDELS 2005*, LNCS, 3844, 355-356, Springer, Heidelberg.
- [20] Shen, G., Huang, Z., Zhu, X., Zhao, X. (2006). Research on the Rules of Mapping from Relational Model to OWL. *In: Proceedings of the Workshop on OWL: Experiences and Directions*. Athens, GA, USA.
- [21] Tirmizi, S. H., Sequeda, J. F., Miranker, D. P. (2008). Translating SQL Applications to the Semantic Web. *In: Proceedings of the 19th International Conference on Database and Expert Systems Applications*, September 01-05, 2008, Turin, Italy. S. S. Bhowmick, J. Küng, and R. Wagner, Eds. Lecture Notes In Computer Science, 5181. Springer-Verlag, Berlin, Heidelberg, 450-464.

[22] Xu, Z., Zhang, S., Y. D. (2006). Mapping between Relational Database Schema and OWL Ontology for Deep Annotation. *In: Proceedings of the IEEE/WIC/ACM international Conference on Web intelligence. Web Intelligence.* IEEE Computer Society, Washington, DC, 548-552.

[23] Yang, H., Qingping, Y., Xizhi, S., Peng, W. (2008). Applying semantic web services to enterprise web. *In: 6th International Conference on Manufacturing Research, 9-11.* Brunel University, UK.

Author Biographies



Inaya Lahoud is born in 1986 in Maghdouche, Lebanon. She got his two-year technical degree in Computer Science from the Lebanese University, his Engineer diploma in computer science (major in: Software and Knowledge Engineering) from the University of Technology of Belfort Montbeliard in 2010 (UTBM, France) and his Master degree in embedded and communicating systems from the Haute Alsace University(North-East of France). Today Inaya is a PhD candidate at the SET laboratory. His research concentrates on designing and developing a multi-agents system to manage heterogeneous and distributed knowledge in the case of extended enterprises.



Vincent Hilaire received his doctorate in computer science and his position as senior lecturer in the “*Université de Technologie de Belfort-Montbéliard*” in 2000. He received his Research Direction Habilitation, at the “*Université de Franche- Comté*” in 2008. Since 2008, he is responsible of the Multi Agent Systems team of the SeT Laboratory. His research interests include: formal specification and methods for engineering multi-agent systems, knowledge management based on multiagent systems and design architectures for agents.



Davy Monticolo received his doctorate in computer science at the “*Université de Technologie de Belfort-Montbéliard*” in 2008. Till 2010, Davy Monticolo was an assistant professor in the Systems and Transports Laboratory University of Technology UTBM, and he is currently a senior lecturer at the ERPI Laboratory in Polytechnic National Institute of Lorraine in the domain of Multi-Agent Systems.