World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:8, No:10, 2014

Automatic Generation of Ontology from Data Source Directed by Meta Models

Widad Jakjoud, Mohamed Bahaj, Jamal Bakkas

***Abstract*—**Through this paper we present a method for automatic generation of ontological model from any data source using Model Driven Architecture (MDA), this generation is dedicated to the cooperation of the knowledge engineering and software engineering. Indeed, reverse engineering of a data source generates a software model (schema of data) that will undergo transformations to generate the ontological model. This method uses the meta-models to validate software and ontological models.

***Keywords*—**Meta model, model, ontology, data source.

I. Introduction

ONTOLOGIES are increasingly used which focuses on the medium of storage of its data. To avoid storing data in memory using specific text files (OWL/XML, RDF/XML,…) the use of DBMS seems the best solution in term of resource management. That said, researchers in the ontological field are giving growing interest to data sources (mostly relational), seen that the management systems of the latter offer a mature solution with solid theoretical foundations.

Model Driven Engineering (MDE) represent a change of paradigm of the software engineering, it focuses on the notion of model (higher level of abstraction) as the basis of conception replacing the notion of object in object's approach. A model undergoes transformations to generate other models of the same nature (endogenous transformation) or the different natures (exogenous transformation) that's why we interest in the MDE in this paper:

From a data source we can extract a model by retro engineering; this model would be transformed to a model of ontology.

II. Related Works

Ontology can be created or enriched by various data sources. We quote the not structured data such as text and dictionaries …, the data semi structured or data based on structured schemas.

In this paper, we focus on data sources structured and semi structured. By manipulating ontology, many problems can be detected: merging several models of ontology in the same architecture, enrichment of ontology by several data sources and supplying data sources with ontological data.

We differentiate several families of research works which approach the transition of data sources towards ontology.

Widad Jakjoud, Mohamed Bahaj, and Jamal Bakkas are with University Hassan 1, Laboratory LITEN, 26000 Settat, Morocco (e-mail: jakjoudw@gmail.com, mohamedbahaj@gmail.com, jbakkas@gmail.com).

A. Passage of the Relational Schema to the Ontological Model

We quote the passage from relational database to ontology, [1], [2], this passage is determined by rules of transition of the elements of relational database (relations, attributes, constraints ...) to the equivalent elements in the ontology (classes, properties data, properties object, ...). Another demarche [3] is based on analysis of the relational schema combined with analysis of tuples of the relational database to extract more semantics. Other work [4] are based on the analysis of user views (HTML forms on a web application, for example) in order to extract the semantics that will enrich the relational model. Rules are defined by passages in order to convert the relational schema directly to ontology.

B. Passage of the Class Diagram to the Ontological Model

Several works were done to describe the direct transformation of UML to OWL. We quote works based on the transformation of UML's models to ontology by using serialization XMI with XSLT: [5]-[7], the idea is to transform UML models into XMI and apply the XSLT transformation. Another demarche uses the transformation language ATL (Atlas Transformation language) for passage the model to ontology [8]. We also include works that offer a translation of UML to OWL with preservation of semantics [9]. Other works [10]-[12] are based on UML profiles; they each propose its UML profile and the transition to ontology languages. Reference [13] proposed another approach which is based on UML diagrams annotated while the approach [14], [15] translates the UML in OCL and then in OWL. Reference [16] proposes a solution based on an MDA-defined architecture for ontology development and the Ontology UML Profile (OUP), this approach represent an ontology by a UML Profile and transform into OWL description.

III. Proposition

This work is integrated of an approach to two-way navigation between data sources eventually heterogeneous and ontology.

We propose the generation of a model of ontology from a data source model, this generation is based on a reverse engineering in order to preserve the flexibility of models on data level (persistence level) and functionality level (software level).

As any proposal which based on the MDA approach [17], we use the models as abstract elements for the standardization of the stages of the generation of the ontology from any data source.

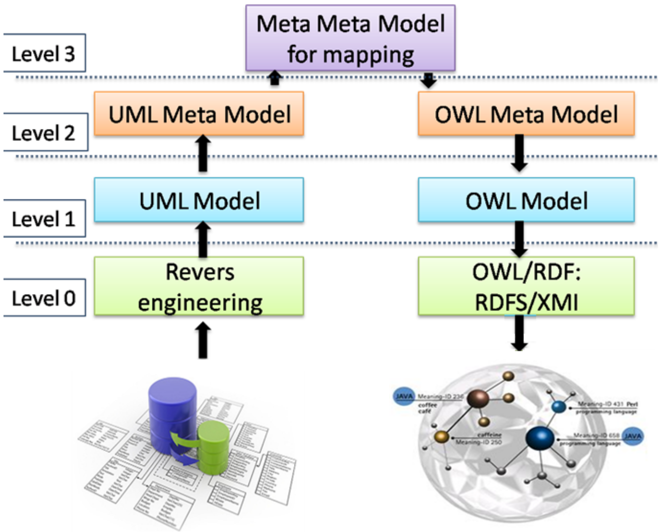

Fig. 1 General schema of the proposed approach

Indeed, the MDA (Model Driven Architecture) [18] is based on techniques of modelisation and techniques of models transforming. It creates a computation independent model that transforms at a Platform Independent Model and eventually transforms it into a specific model (implementation).

The originality of our proposal is that it is based on Meta models perfectly tailored to the nature of the problem (convert conceptual representations, expressed as diagrams graphics classes into Ontology Language OWL represented with XMI, RDF and OWL). The mapping between Meta models is expressed by an operational language "QVT Operational"

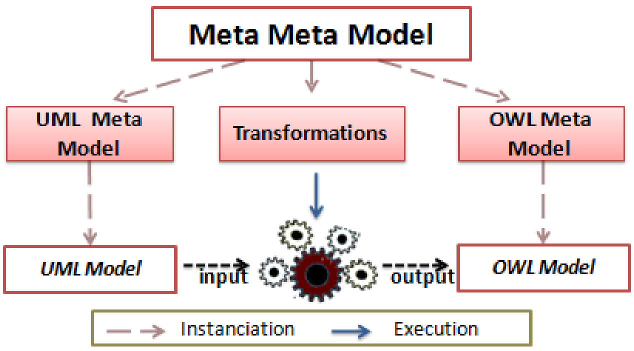

Fig. 2 Detailed schema of the proposed approach

The UML model results of reverse engineering from data source. Indeed, whatever the data source it is based on a single model which will undergo a reverse engineering to generate a UML model (class diagram).

A. Meta Meta Model Level

In Meta Meta model level, we use the MOF (Meta Object Facility) [19] standard for manipulate the lower level Meta model. Several standards exist, namely MOF and BNF (Backus–Naur Form) we choose MOF because it is the OMG (Object Management Group) standard. Indeed, the MOF is used to define Meta models; it is a cyclical language for Meta models which means it describes itself:

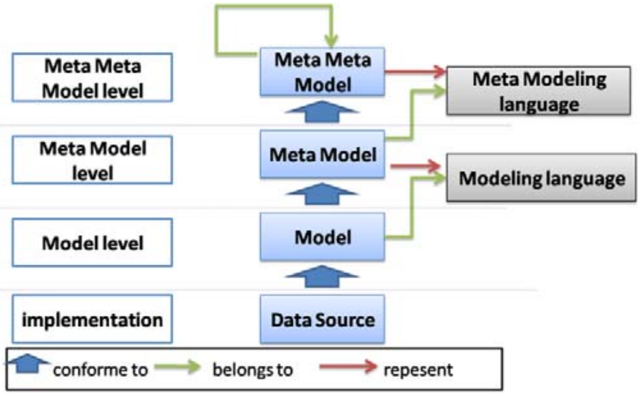

Fig. 3 Relations between levels

B. Meta Model Level

1) The Proposed Meta Model of Class Diagram

At Meta model level, we propose the UML Meta model elaborated with the UML Language.

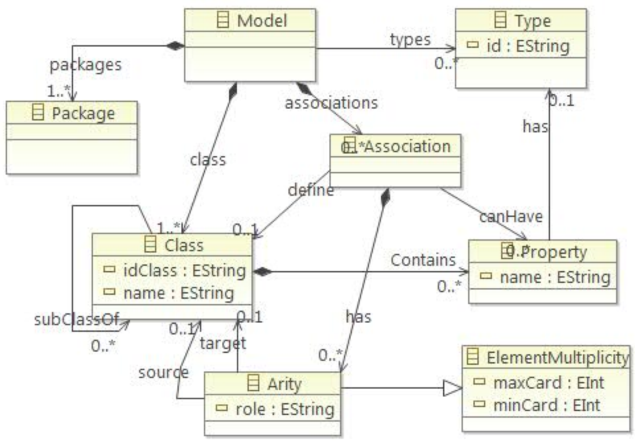

Fig. 4 The proposed Meta Model of Class Diagram

UML Meta model is a class diagram that specifies the structure of all UML diagrams. Indeed, class diagram is UML Diagram which defines concepts through classes and the relations between concepts through associations.

To define a Meta model, with all types of elements, constraints and relationships, several approaches are possible:

- Define a new Meta model which is not based on Meta model already exist
- Define a Meta model based on an existing Meta model by modifying, deleting items unnecessary for business context or technical context in question, or even adding elements considered essential.
- Be based on Meta models provided by Meta Meta models like Ecore which propose a UML Meta model.

we opted for the definition of our Meta model UML based on existing Meta model by adding some simplifications and modifications to adapt it to our context of study: Our Meta model is then a class diagram consisting of packages, types, associations, each association is a set of arities, each arity connects a source class with a target class . Arities are in number of participating classes in the association. An UML class can inherit from another UML class.

2) The Meta Model OWL Proposed

We also propose the OWL Meta model developed in UML:

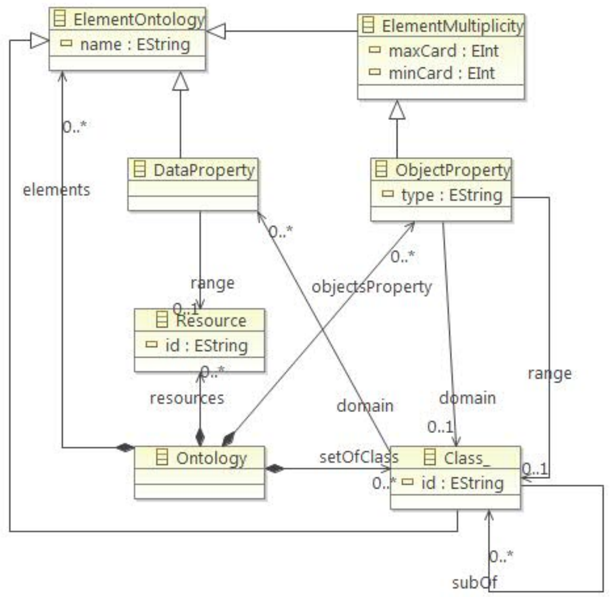

Fig. 5 The Meta Model OWL proposed

We consider ontology as a set of classes, resources, object properties and data properties. A class is an ontology element which can have other sub classes

An ElementMultiplicity is an element of ontology which can be multiple; we define a maximal cardinality and a minimal cardinality.

An Object property is an ElementMultiplicity; it is connected to a source class (domain) and a target class (range)

A data property is an element of ontology which has a resource (range) it is member of a class (domain).

3) Transformations

Transformations are seen as inference rules parametrisables. A transformation is used to switch from a conceptual representation to another by translating the elements of the first to find elements of the other. Indeed, a transformation can take upstream [1 ... *] source models that will transform in [1...*] target model, each model must be validated against a Meta model.

The academic projects under this approach are many, such as: AndroMDA [20], ATL (Atlas Transformation Language) [21], MTL (Model Transformation Language) [22] and QVT (Query View Transformation) [23] which is an OMG standard.

A model (view), instance of other models, will undergo transformation (transformation). A transformation can modify a model or create a new one.

To express the transformation rules, we opt for the QVT Operational [24] language, an OMG standard, which is based on queries (query) on model.

A model (view), instance of other models, will undergo transformation (transformation) .

A transformation can modify a model or create a new one.

The QVT standard defines three languages for expressing model to model transformations, such as:

- QVT Relations and QVT Core are declarative, the relations established between elements of source and target models are specified using constraints. They are purely declarative and their specification is not executable,
- QVT Operational mapping that has the privilege of being imperative; the transformations will be expressed in the form of executable instructions.

Qvt Operational may specify particular aspect of a transformation in a declarative languages (core or relation) then implement equivalent rules in the imperative language.

IV. Implementation

The implementation of our proposal is based on the following standards OMG:

- UML language for describing UML Meta model and models
- MOF as Meta modeling language
- QVT Operational mapping language for transforming models.

A. Meta Meta Model Level

In our approach, Meta Meta model level is assured by the standard MOF.

B. Meta Model Level

The application generates each model in XMI (Xml Metadata Interchange) format which is a standard for exchanging models and Meta models.

The transformation rules consist in defining mappings between different components of the source Meta model to the target Meta model, we define:

mapping Model :: ModelToOntology(): Ontology : We express the necessary treatments to transform a component "Model" of UML Meta model to component "Ontology" of Meta model OWL.

mapping Class :: ClasseToClasse() : Class : We transform a component "Class" of Meta model UML to a component "Class" of Meta model OWL.

mapping type :: TypeToRssource() : Resource : It consists to transforming a component "type" of Meta model UML to a component "Resource" of Meta model OWL.

*mapping Arity:: ArityToObjectProperty() : ObjectProperty :*It allows transforming a component "Arity" of Meta model UML to a component "ObjectProperty" of Meta model OWL.

*mapping Property :: PropertyToDataProperty() : DataProperty :*We transform a component "Property" of Meta model UML to a component "DataProperty" of Meta model OWL.

mapping Class :: SubClassOfToSubClassOf():Class : we transform the UML inheritance relation to the inheritance relation of OWL.

C. Model Level

We consider the class diagram generated from a data source:

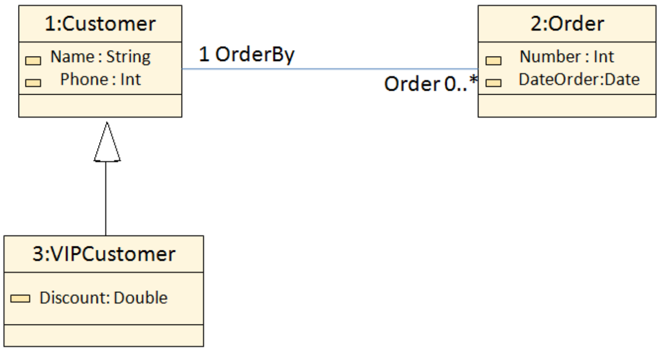

Fig. 6 The class diagram generated from a data source

For example, the mapping ModelToOntology generate the following trace:

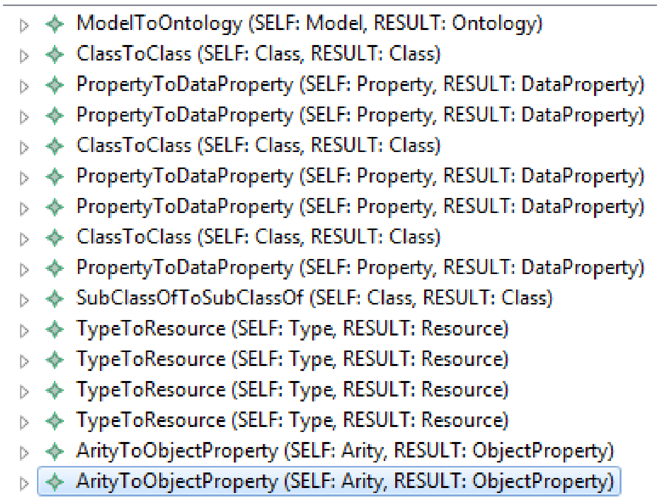

Fig. 7 Trace of mapping ModelToOntology

The result expressed in XMI:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.0"
xmlns:xmi="http://www.omg.org/XMI"
xmlns:owlmodele="http://owlmodele/1.0"> <owlmodele:Ontology>
    <setOfClass name="owl_customer">
  <domain name="Name" range="/0/@resources.0"/>
  <domain name="Phone" range="/0/@resources.1"/>
 </setOfClass>
  <setOfClass name="owl_Order">
  <domain name="DateOrder" range="/0/@resources.2"/>
  <domain name="Number" range="/0/@resources.1"/>
  </setOfClass>
  <setOfClass name="owl_VIPCustomer" subOf="/1">
<domain name="Discount" range="/0/@resources.3"/>
  </setOfClass>
  <resources id="String"/>
  <resources id="Int"/>
  <resources id="Date"/>
  <resources id="Double"/>
   <objectsProperty name="Order" cardMax="1" cardMin="1"
domain="/0/@setOfClass.1" range="/0/@setOfClass.0"/>
   <objectsProperty name="orderBy" cardMax="10" cardMin="1"
domain="/0/@setOfClass.0" range="/0/@setOfClass.1"/>
   </owlmodele:Ontology>
  <owlmodele:Class/>
  </xmi:XMI>
```

The generated ontology:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:genOnto="file://D:/genOnto.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
 <rdf:Description rdf:about="genOnto:owl_order">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
 </rdf:Description>
 <rdf:Description rdf:about="genOnto:orderBy">
  <rdfs:range rdf:resource="genOnto:owl_customer"/>
  <rdfs:domain rdf:resource="genOnto:owl_order"/>
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
 </rdf:Description>
 <rdf:Description rdf:about="genOnto:discount">
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="genOnto:owl_VIPCustomer"/>
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
 </rdf:Description>
 <rdf:Description rdf:about="genOnto:DateOrder">
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
  <rdfs:domain rdf:resource="genOnto:owl_Order"/>
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
 </rdf:Description>
 <rdf:Description rdf:about="genOnto:owl_VIPCustomer">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
 </rdf:Description>
 <rdf:Description rdf:about="genOnto:order">
  <rdfs:range rdf:resource="genOnto:owl_order"/>
  <rdfs:domain rdf:resource="genOnto:owl_customer"/>
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
 </rdf:Description>
 <rdf:Description rdf:about="genOnto:name">
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="genOnto:owl_customer"/>
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
 </rdf:Description>
 <rdf:Description rdf:about="genOnto:number">
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
  <rdfs:domain rdf:resource="genOnto:owl_order"/>
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
 </rdf:Description>
 <rdf:Description rdf:about="genOnto:phone">
```

```
   <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
   &lt;rdfs:domain rdf:resource="genOnto:owl_customer"/&gt;
   <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  &lt;/rdf:Description&gt;
  &lt;rdf:Description rdf:about="genOnto:owl_customer"&gt;
   &lt;rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/&gt;
  &lt;/rdf:Description&gt;
  &lt;rdf:Description rdf:about="file://D:/genOnto.owl#"&gt;
   <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#Ontology"/>
  &lt;/rdf:Description&gt;
&lt;/rdf:RDF&gt;
```

The generated ontology visualized by Protégé:

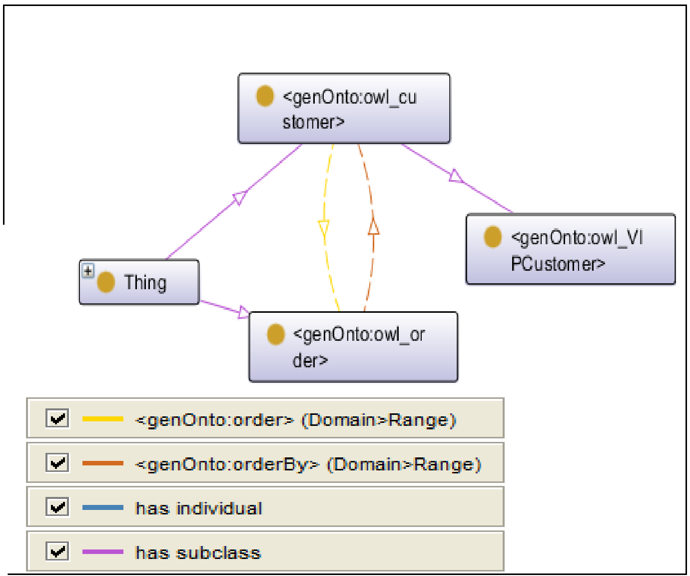

Fig. 9 The resultant ontology visualised by Protege

V. Conclusion and Perspectives

The paper proposed concerns generating ontology from a data source. This proposal is generic and in accordance with the MDA approach. Indeed, from any data source we can extract a unique model that will undergo transformations through a process managed by the models to generate automatically the ontology. The Meta models proposed are appropriate with respect to the posed problem.

In perspective we will extend the proposal to discuss the transposition at the data level.

References

[1] L. Stojanovic, N. Stojanovic and R. Volz, Migrating Data-intensive Web Sites into the Semantic Web, In: Proceedings of the 17th ACM Symposium on Applied Computing (SAC) (2002) 1100–1107.

[2] G. Dogan, and R. Islamaj, (2002). Importing relational databases into the semantic Web. *2002). (2006-11-05). http://www.-mindswap, org/webai/ 2002/fall/Importing_20Relational_20Databases_20into.20the__2OSemantic_20 Web.html*.

[3] I. Astrova, (2004). Reverse engineering of relational databases to ontologies. In*The Semantic Web: Research and Applications* (pp. 327-341). Springer Berlin Heidelberg.

[4] S.M Benslimane, D. Benslimane, M. Malki, Y. Amghar, and H. Saliah-Hassane. (2006, July). Acquiring owl ontologies from data-intensive web sites. In*Proceedings of the 6th international conference on Web engineering* (pp. 361-368). ACM.

[5] S. Cranefield. (2006). Networked knowledge representation and exchange using UML and RDF. *Journal of Digital information*, *1*(8).

[6] D. Đurić. (2004). MDA-based ontology infrastructure. *Computer Science and Information Systems*, *1*(1), 91-116.

[7] S. Leinhos . (2006). OWL ontology extraction and modelling from and with UML class diagrams-a practical approach. *University of the Federal Armed Forces of Germany*.

[8] M. Milanović, D. Gašević, A. Giurca, G. Wagner and V. Devedžić. (2006, February). On interchanging between owl/swrl and uml/ocl. In *Proceedings of 6th Workshop on OCL for (Meta-) Models in Multiple Application Domains (OCLApps) at the 9th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MoDELS), Genoa, Italy* (pp. 81-95).

[9] M. Bahaj, and J. Bakkas, Automatic Conversion Method of Class Diagrams to Ontologies Maintaining Their Semantic Features.

[10] S. Brockmans, R. M. Colomb, E. F. Kendall, E. Wallace, C. Welty, G.T. Xie and P. Haase (2006). A Model Driven Approach for Building OWL DL and OWL Full Ontologies. 5th International Semantic Web Conference, vol. 4273 of LNCS, pp. 187-200, Athen, USA.

[11] J. Evermann. (2009). A UML and OWL description of Bunge's upper-level ontology model. *Software & Systems Modeling*, *8*(2), 235-249.

[12] M. Mehrolhassani and E. L. Ç. İ. Atilla. (2008). *Developing Ontology Based Applications of Semantic Web Using UML to OWL Conversion* (pp. 566-577). Springer Berlin Heidelberg.

[13] C. Faucher, F. Bertrand, and J. Y. Lafaye. (2008). Génération d'ontologie à partir d'un modèle métier UML annoté. *RNTI*, *12*, 65-84.

[14] D. Gasevic, D. Djuric, V. Devedzic, and V. Damjanovi. (2004, May). Converting UML to OWL ontologies. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters* (pp. 488-489). ACM.

[15] J. Zedlitz, and N. Luttenberger. (2012, November). Transforming Between UML Conceptual Models And OWL 2 Ontologies. In *Terra Cognita 2012 Workshop*(Vol. 6, p. 15).

[16] Dragan Gasevic, Dragan Djuric, Vladan Devedzic, Violeta Damjanovi Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, pp. 488-489, ACM New York, NY, USA ©2004

[17] J. Bézivin and X. Blanc, (2002). MDA: Vers un important changement de paradigme en génie logiciel. *Développeur référence v2*, *16*, 15.

[18] J. M. Jézéquel, S. Gérard, and B. Baudry. (2006). Le génie logiciel et l'IDM: une approche unificatrice par les modèles. *L'ingénierie dirigée par les modèles*.

[19] A. G. Kleppe, J. Warmer, W. Bast and M. D. A. Explained. (2003). The model driven architecture: practice and promise.

[20] AndroMDA Model Driven Architecture Framework", http://galaxy.andromda.org/docs-3.2/, 2007.

[21] F. Jouault, F. Allilaire,J. Bézivin and I. Kurtev. (2008). ATL: A model transformation tool. *Science of computer programming*, *72*(1), 31-39.

[22] R. Silaghi, F. Fondement, and A. Strohmeier. (2005). "Weaving" MTL model transformations. In *Model Driven Architecture* (pp. 123-138). Springer Berlin Heidelberg.

[23] Object Managment Group.Request for proposal :Mof 2.0 query/view/transformation rfp.Request for proposal ad/2002-04-10, OMG, Octobre 2002.

[24] Object Managment Group. Mof qvt final adopted specification. Technical Report ptc/05-11-01, OMG, Novembre 2005.

Widad Jakjoud was born in 1979, in Marrakech, Morocco. She got her diploma of computer science engineer after completing her license in computer sciences from University Cadi Ayyad of Marrakech; she is PhD student in the Department of Mathematics and computer sciences, Faculty of Sciences & Technology of Settat, University Hassan I, Settat, Morocco. In her research she is interested in the ontology, semantic web and Model Driven Engineering.

Mohamed Bahaj was born in 1964, in ouezzane, Morocco. He got his PhD in Applied Mathematics, from University of Pau, France, in 1993. He is now working as a Professor at the Department of Mathematics & Computer Sciences, University of Hassan 1er, Faculty of Sciences & Technology Settat, Morocco. His research interests include pattern recognition, Semantic web & Ontology in MAS, Controls of mobiles agents.

Jamal Bakkas was born in 1979, in Marrakech, Morocco. He got his special higher studies degree in engineering of information systems after completing his license in computer sciences from University Cadi Ayyad of Marrakech; He is phd student in the Department of Mathematics and computer sciences, Faculty of Sciences & Technology of Settat, University Hassan I, Settat, Morocco. His area of interest includes web ontologies and semantic web.