

# Ontology Modeling Profile, an Extension for the Ontology UML Profile

Wafaa Alakwaa  
Computer Science Dept.  
Cairo University  
Giza , Cairo 12613, Egypt

Akram Salah  
Computer Science Dept.  
Cairo University  
Giza , Cairo 12613, Egypt

## ABSTRACT

This paper presents an extension for the Ontology Modeling Profile (OUP). The extended profile together with Ontology Definition Metamodel (ODM), enables the usage of Model Driven Architecture (MDA) standards in ontological engineering. Our profile is based on the recent W3C effort – The Web Ontology Language (OWL). We have shown the role of individuals and slots in building OWL statements. This utilization improves the practical implementation of Ontology models using UML profiles..

## General Terms

Semantic web, Ontology , Ontology Modeling.

## Keywords

Unified Modeling Language (UML), Model Driven Architecture, Ontology (MDA), Semantic Web, Ontology Definition Metamodel(ODM), Web Ontology Language (OWL).

## 1. INTRODUCTION

The semantic web is the next generation of the current web, that allows resources on the web to become machine processable rather than just human processable. The main idea that semantic web relies on to achieve this goal is creating a map for these resources. That is because annotation on semantic web express links between the web resource and its metadata (ontology). Ontologies play an important role in fulfilling semantic interoperability. Due to ontology's formal semantic and consensual terminologies, it facilitates sharing and reusing ontologies, which in turns interweave human and machine understanding [23].

Artificial intelligence (AI) techniques that are used for ontology creation are more related to research laboratories, and are unknown to wider software engineering population. In order to overcome the gap between software engineering practitioners and AI techniques, there are a few proposals for UML use in ontology development[4]. But, UML itself does not satisfy needs for representation of ontology concepts that are borrowed from description logics, and that are included in Semantic Web ontology languages (e.g. RDF, RDF Schema, OWL, etc.). The OMG's has created a Request For Proposal (RFP) metamodel for developing Ontology using MDA standards. Accordingly, as an answer to this request in [6], the authors show a metamodel for ontology modeling language - Ontology Definition Metamodel

(ODM). This metamodel is defined using Meta- Object Facility (MOF), and is based on the Web Ontology Language (OWL). Since Unified Modeling Language (UML) is widely accepted as a modeling language, they also defined a profile that supports ontology design - Ontology UML Profile. Ontology UML Profile (OUP) is intended to be used as a support to ODM, not as a stand-alone solution for Ontology modeling.

In this paper we have adapted the OUP by adding some extensions and modifying some concepts found in the ODM. Our modified profile was standard and based on MOF. We noticed that although the OUP has covered almost all concepts created in the OMG's ODM , it missed some important ODM concepts such as slot definition. One key modification is redefining the ODM statements elements.

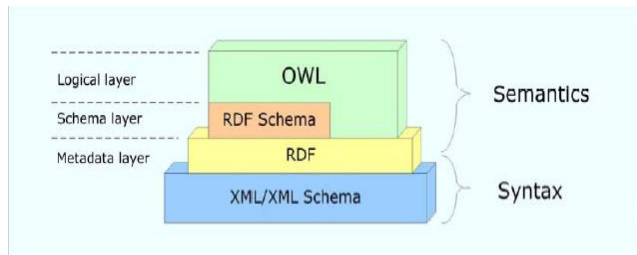
The overview of the Semantic Web languages and OWL is given in the next section, the description of the MDA and MOF is explained in section three. Section four shows in details the need for the ODM over UML, and related work on the use of MDA standards and UML for Ontology Modeling. The OUP is explained there too. In section five, we show in details the extended stereotype and the need for them, with a simple use case in section seven. The last section concludes showing our view for some work that could be done in the near future..

## 2. AN OVERVIEW OF THE SEMANTIC WEB AND OWL

One of the more recent developments on web is an activity known as the Semantic Web. The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. For the semantic, the most important component is Ontology. Ontology can be defined according to the Object Management Group's Ontology Definition Metamodel document as a "specification of a conceptualization." And according to the W3C's Web Ontology Language Overview, an ontology is "the representation of meaning of terms in vocabularies and relationships between those terms." [11].

Semantic Web architecture is a functional, non-fixed architecture [12]. Barnes-Lee defined three distinct levels that incrementally introduce expressive primitives: metadata layer, schema layer and logical layer Languages that support this architecture [7]. Fig.1 shows the main 3 layers of the semantic web architecture,

where each of these layers is based on a technology that plays a distinct role in deploying and reusing learning objects on the Semantic Web. Metadata layer based on XML and RDF, schema layer based on RDF schema and finally the logical layer that is based on OWL [5]



**Figure 1: OWL in the Semantic Web architecture.**

XML is used to markup the structure (syntax) of a resource in a machine readable way by associating metadata with resources. RDF allows the specification of metadata in a more flexible manner, facilitating the discovery and exchange of resources with limited information or more than one metadata specifications. OWL allows the specification of concepts in a domain as well as the terms used to markup content [25].

The OWL language provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users. The OWL Lite supports the primarily classification hierarchy and simple constraint features. OWL DL supports the maximum expressiveness without losing computational completeness and decidability of reasoning systems. OWL Full is meant for maximum expressiveness and the syntactic freedom of RDF with no computational guarantees [17]

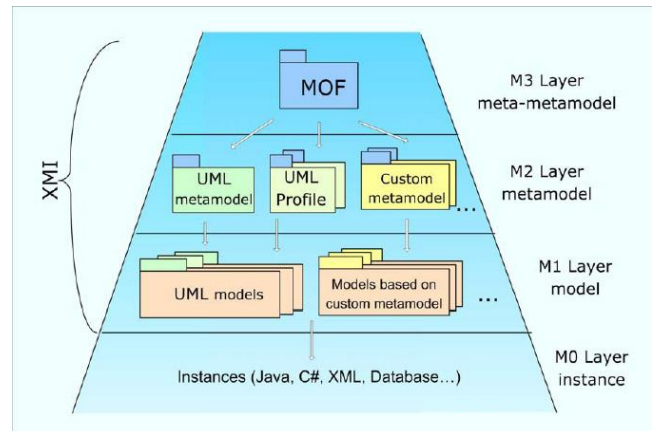
### 3. AN OVERVIEW OF MDA AND MOF

Model driven development ( MDD ) constructs systems by means of models, which make systems seen from an abstract view. Many models are created for the same system , each addresses different view [22] They (models) solve problems away from the technical details, making world more reusable and easy to create by domain experts, requiring less knowledge of specific computer systems [1].

The Object Management Group (OMG), defined a framework called Model Driven Architecture (MDA) for model driven development. This framework (MDA) separates the system functionalities from the implementation details [15]. In MDA three types of viewpoints on models are distinguished. The computationally independent (CI) viewpoint sees the system from the customer's point of view, and manifests it in a computation-independent model (CIM). This model is a typical analysis model, since it is expressed in terms of the problem domain. The platform-independent (PI) viewpoint sees the system from the designer's point of view, abstracts from all platforms a system may run on, and results in a platform independent model (PIM). Finally, the platform-specific viewpoint adds platform specific extensions and results in a platform-specific model (PSM). MDA is supported by a series of OMG standards, including the UML(Unified Modeling

Language), MOF (Meta-Object Facility), XMI (XML Metadata Interchange), and CWM (Common Warehouse Metamodel). MDA also includes guidelines and evolving supporting standards on model transformation and pervasive services. The standards in MDA collectively define how a system can be developed following a model driven approach and using MDA compatible tools. Each MDA standard has its unique role in the overall MDA picture [16].

MDA is based on the four-layer metamodelling architecture, as shown in Fig.2, which are :meta-metamodel (M3) layer, metamodel (M2) layer, model (M1) layer and instance (M0)layer [6] .



**Figure 2: MDA four-layer MOF-based metadata architecture.**

Meta-Object Facility (MOF), the M3 layer, is the responsible for defining abstract languages and frameworks for specifying, constructing and managing technology neutral metamodels. It is the foundation for defining any modeling language; such as UML, a graphical modeling language for specifying, visualizing and documenting software systems. The models of the real world, represented by concepts defined in the corresponding metamodel at M2 layer (e.g. UML metamodel) are at M1 layer. Finally, at M0 layer are instances of concepts modeled at M1 layer [6].

Extensions of the UML can be either heavyweight or lightweight. The difference between lightweight and heavyweight extensions comes from the way in which they extend the UML metamodel. Heavyweight extensions are based on a modified UML metamodel with the implication that the original semantics of modeling elements is changed and therefore the extension might no longer be compatible with UML tools. Lightweight extensions are called UML profiles and are based on the extension mechanisms provided by UML(stereotypes, tag definitions, and constraints) for specializing its metaclasses, but without breaking their original semantics. UML profiles may impose new restrictions on the extended metaclasses, but they should respect the UML metamodel, without modifying the original semantics of the UML elements (i.e., the basic features of UML classes, associations, properties, etc., will remain the same, only new constraints can be added to the original elements). One of the

major benefits of profiles is that they can be handled in a natural way by UML tools [18].

#### 4. USING UML AND MDA-BASED STANDARDS IN ONTOLOGICAL ENGINEERING

The importance and use of ontology was expanded from being a basic building block of the Semantic web [14], to participate in many software applications and the critical semantic foundation for many rapidly expanding technologies such as software agents, e-commerce and knowledge management [10]. This importance caused for many new tools to be developed to accelerate and aid in building , representation, design and construction of domain ontologies [20]. Most of the current Semantic Web ontologies are developed in AI laboratories. Because of this, the use of ontologies by Software engineers professionals and researchers can be seen as an additional learning experience, and in some cases, of considerably great effort. Another aspect is visualization for complex ontologies that need to be represented in a way that uses formal notation with known tools.

Researchers have investigated that a strong coupling exists between the knowledge engineering and software engineering phases of a knowledge-based system. These researches tried to converge between MDA standards and ontology developments. Applying MDA techniques in developing ontologies has been discussed, focusing on what is common among them [24].

OMG ,as a consortium which develops standards for various aspects of software engineering which are widely used in industry, including UML, has published a RFP (Request for Proposal) that tries to define a suitable language for modeling Semantic Web ontology languages in the context of MDA. This RFP was responsible for modeling Web Ontology Language, which is a W3C effort. This metamodel will make ontology being used in a computing application. Ontology then could be represented as some sort of computer-readable data structure [3]. Although an ontology is a kind of data model, the UML metamodel, which is a rich in the purpose of representing data, widely used, and well supported with software tools, was not being suitable for developing ontologies. The Web Ontology language has many shared concepts with the UML metamodel. But these concepts have many deep differences, that make each has unique representation manner.

The most important element is Class . A class in UML is a kind of classifier whose features are attributes and operations. In UML, instances (Objects) usually have more meaning than do classifiers because classifiers represent at least the first level of abstraction. Instances under the same classifier help people understand what classifiers are. But class is not a collection of instances (group of objects sharing the same description). Saying that "a class is a set of objects" is simply incorrect. Class is just a mold defining all the properties of instances [2].

While in ODM ,the concept Class refers to any concept that could be modeled as a resource. The concept could be instantiated to create an individual or specified by having properties. A Class is a set of individuals (in contrast to UML

class), which by themselves belong to special concept (class). A list of ontology classes exist each for different purpose. From describing a naturally occurring concepts(OWL:Class) to list of existing individual(enumeration) to concepts that is deduced from other concepts either by union, intersection or complement. OWLRestriction class is responsible for setting values restriction or cardinality restrictions if exists. Individuals which are different from UML objects (instances) are members of a classes. Another and important difference between Individual and Object is that, individuals may be specified independently and might not have the same set of properties that the class has, such as (sameAs, DifferentFrom) [17].

The third shared concepts between UML and ODM, but has completely different representation is property. Properties in UML , which are either attributes or association's ends. As seen in Figure 3 properties appears either in the class as an attribute, or in the association as an association end. So Property can never be by itself in a model. It should be owned by some kind of class or association [2].

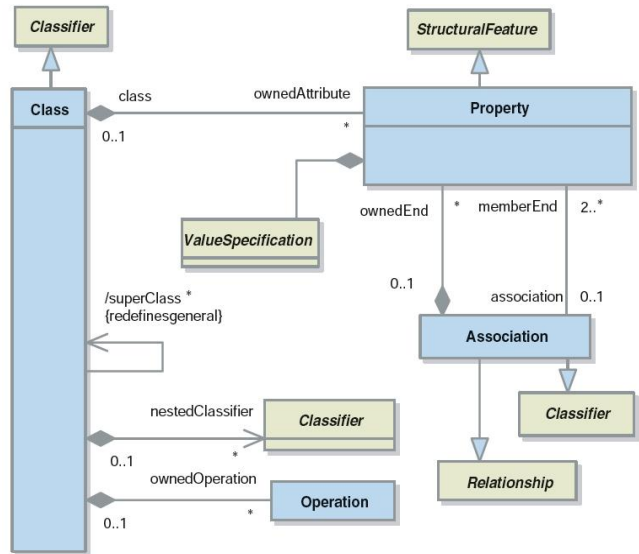


Figure 3: Class and Property concepts of the Kernel package.

Unlike properties in ODM, properties are independent, stand alone concepts. There exists two types of properties, DatatypeProperty and ObjectProperty. The difference between them is in the type of the range they link individual with. DatatypeProperties link individuals with primitive types value (e.g Integer, Boolean, String ...), while ObjectProperties link individuals with individuals (e.g colleagueOf, hasParent..). Generally one can map datatypeproperty to UML attribute while objectProperty could serve as UML association [13].

In order for ODM to support well-known Semantic Web ontology languages, it has two separated metamodels, namely OWL and RDFS Metamodels. These languages are W3C standards which form a basis for the Semantic Web, thus are the central part of the ODM. Other metamodels have two-way mappings to and from RDFS/OWL. The other metamodels are Common Language (CL) metamodel, Topic Maps (TM) metamodel, and the Description Logics (DL) metamodel [3].

Below we will have an overview of the main ODM concepts .

#### 4.1 The OMG's Ontology Definition Metamodel

The Web Ontology Language is built on the top of RDF(S), using RDF(S) as both a meta-metamodel (M3) and a metamodel that is a base for extension (M2). In the MOF MS, the first dependency (the use of RDFS as a metamodel) is replaced by using the MOF as a meta-metamodel. The second dependency means that the concepts of the OWL metamodel extend the concepts of the RDFS metamodel. Most OWL concepts inherit the RDFS concepts RDFSResource, RDFProperty and RDFClass. RDFClass is a base concept of the OWL concepts that represents classes(OWLClass, OWLRestriction ), RDFProperty is inherited by many concepts that represent properties in OWL (OWLObjectProperty, OWLDatatypeProperty and so on). OWLClass which is a set of individuals, inherits RDFClass. OWLClass can be defined in OWL in several other ways. Enumeration is defined by exhaustively enumerating its instances. A class can be constructed in OWL as a complement of another class or as a union or intersection of other classes. OWLRestriction is a special kind of OWLClass, and thus it inherits the OWLClass concept. It is not a “real” class, but a concept that enables constraints in OWL. OWL refines the concept of rdf:Property by distinguishing two basic kinds of properties, owl:ObjectProperty and owl:DatatypeProperty. Properties in OWL cannot have a Datatype as a domain, but only as a range. A data range represents a range of data values. It can be either a datatype or a set of data values. Data ranges are used to specify a range of datatype properties. They are modeled as the OWLDataRange MOF Class [8].

#### 4.2 Related Work

Several different approaches were proposed for ontology representation in UML. The main major trends among them:

- Extending UML with new constructs to support specific concepts of ontologies (Property for example) [13].
- Using standard UML and defining a UML profile for ontology representation. This approach is mostly applied in the recent proposed solutions.

Many approaches suggested using UML for Ontology Development. In 1999, Stephen et al. [21] , investigated the use of UML and OCL for the representation of information system ontologies. They examined the potential for object-oriented standards to be used for ontology modeling, and in particular presents an ontology representation language based on a subset of the Unified Modeling Language together with its associated Object Constraint Language.

In [13] the authors proposed to predefine several stereotypes so that a more detailed mapping from UML to the primitives offered by the DAML+OIL description logic can be achieved. In [13], the authors argued that the UML metamodel should be extended with elements such as property and restriction so that UML becomes more compatible with KR languages like OWL. We

believe that ontology representation in UML can be achieved without nonstandard UML extensions. Models created with UML profiles are supported by standard UML tools, since they do not add nonstandard concepts to UML, and thus they are UML models.

Recently, Brockmans et al. in [19], introduced a visual, UML-based notation for OWL ontologies. They provide a standard MOF2 compliant metamodel which captures the language primitives offered by OWL DL. They also created a UML profile, which allows to visually model OWL ontologies in a notation that is close to the UML notation. This allows to develop ontologies using UML tools. The main drawback that we noticed in this work was on how to represent ObjectProperties and datatypeproperties. Object properties are represented as UML n-ary associations, while datatype properties are represented as UML attributes. This is not comfort to OWL, which uses the term property(either object or Datatype) as an independent concept that is related to individuals via domain or range links. The other drawback was that they completely neglect the Statement concepts.

In [7], D. Djuri et al. proposed a profile named "Ontology UML Profile (OUP)"; which, together with Ontology Definition Metamodel (ODM), enables the usage of Model Driven Architecture (MDA) standards in ontological engineering. What make the OUP the best is their intend to represent OWL ontology , the recent W3C effort. Other similar UML profiles are based on ontology representation languages, such as RDF(S), DAML+OIL, etc. Below is a brief description of this approach.

##### 4.2.1 The Ontology UML Profile (OUP)

In answer to the OMG RFP an architecture was created in [6], that proposes several specifications to be defined, as shown in Figure 4 :

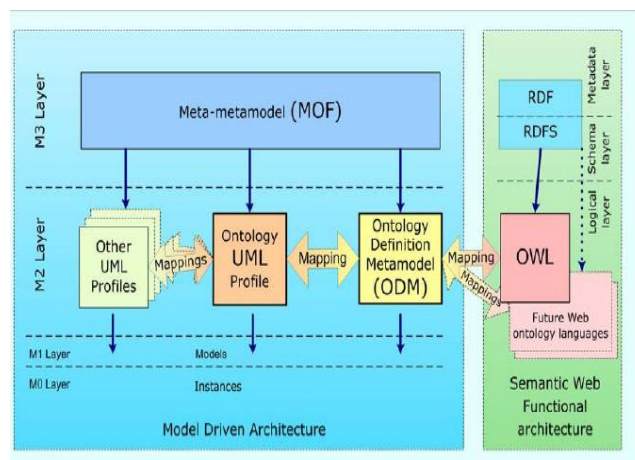


Figure 4: Ontology modeling in the context of MDA and Semantic Web.

- Ontology Definition Metamodel (ODM)
- Ontology UML Profile - a UML Profile that supports UML notation for ontology definition.

- Two-way mappings between OWL and ODM, ODM and Ontology UML Profile and from Ontology UML Profile to other UML profiles

The OUP developer firstly defined the place of ODM and OUP in the context of the MDA four layer architecture and identify the main OWL concepts. Then, to support ODM, they defined the OUP and describe its details. The proposed UML profile enables usage of the well-known UML notation in ontological engineering more extensively. The purpose of OUP is to enable the use of the standard UML graphical notation for developing ontologies. You can refer to [7] for more information.

Due to the difference between UML and ontology class, as it is defined in OWL (owl:Class), OUP defined a stereotyped `<<OntClass>>` UML classes to model ontology classes. And many subclasses of the class `OWL:Class`, stereotyped `<<Enumeration>>`, `<<Union>>`, `<<Intersection>>`, `<<Complement>>`, `<<Restriction>>` and `<<AllDifferent>>` for classes namely Enumeration, Union, Intersection, Complement, Restriction and AllDifferent respectively. Also, there are many differences between UML objects and `OWL:Individual`, OUP has stereotyped UML objects as `<<OntClass>>`.

Property in OWL is a standalone concept, different from the property concept used in UML. Because of that, OUP used the stand alone concept in UML, class, stereotyped by `<<objectProperty>>` for properties among individuals and `<<datatypeProperty>>` for properties among individuals and primitive data type.

OWL statements are represented via concrete links between ontology instances, individuals. In OUP, Statement is represented as an Object, "ObjectProperty" or "DatatypeProperty", with two Links - the subject Link and the object Link. Unlike other MDA-based approaches to ontology development both ODM and OUP support modeling of body of knowledge (i.e. class instances) [9].

We have chosen this profile and then updated it because of many reasons:

- OUP is specialized for modeling the recent W3C effort (OWL).
- It has covered almost all the practical part of building ontology, (e.g. Statements).
- It shows in details how to use UML modeling tools in developing Ontology using their profile. For more details you can refer to [8].

## 5. THE PROPOSED ONTOLOGY MODELING PROFILE (OMP)

From the above explanation of the OUP, we have noticed that the OUP was a complete and worth work not to be neglected. The effort spent their could be accompanied with our extra and modified stereotypes to generate a valid and practice oriented profile.

### 5.1 OWLDataRange

Through the analysis of the ODM, Enumeration are represented in two manner. The class called enumeration, which is

stereotyped `<<Enumeration>>` in the OUP. Enumeration is a class where its elements are individuals. This is due to the fact that in Ontology, a class is a set of individuals. The second type of Enumeration mentioned in the ODM was, `DataRange`, in which elements are all primitive literals (e.g Integer, String, Boolean ..)

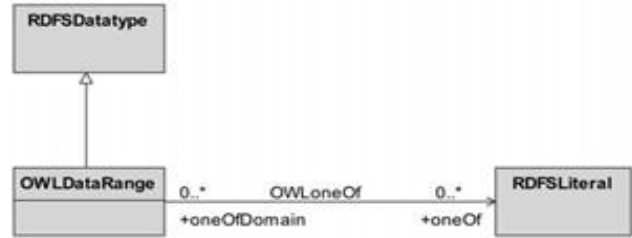


Figure 5: OWLDataRange in the OWL metamodel.

### 5.2 OWLStatement

Statement in OWL, is inherited from `RDFSStatement`. From the definition of `RDFSStatement`, we can see that all three associations, subject, object and predicate, link `RDFSStatement` to `RDFSResource` as shown in Figure 6. This design is wide open to any kinds of statement, even one that does not mean anything. Therefore, the predicate should usually be a resource that represents a verb, such as "be" or "memorize", or some characteristic of a resource, like "name". In the RDFS MS, such a resource is an `rdf:Property`, a concept that represents a type of relationship between resources [17].

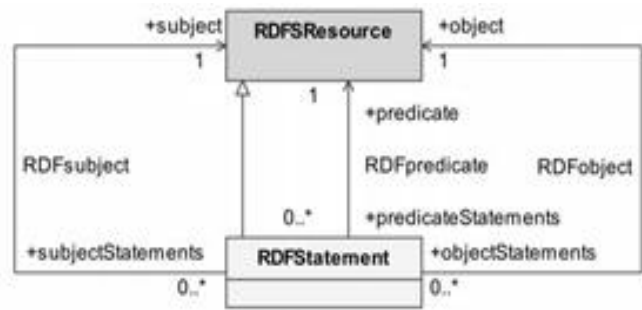


Figure 6: OWLStatement in the OWL metamodel.

From the above definition of a statement in the ODM metamodel, we think that statement in ODM is not supposed to have a similar concept in UML. What is being represented is its elements, Subject, object and Predicate. The OUP has preferred to use link, and link ends. In our opinion, Subject, object and predicate are not links, they are individuals and Object property instances. A statement in OWL has to link 3 instances, each plays a role in this statement. If subject is an Individual, predicate could be an instance of an object property, and the object could be another individual or the same subject individual in case of symmetric object properties. So the proposed OMP define stereotyped instance specification named `<<predicate>>` of a class stereotyped `<<ObjectProperty>>` and a stereotyped instance specification named `<<subject>>` and `<<object>>` for a class `<<ontClass>>` or `<<individual>>`.

Figure 7 shows the InstanceSpecification metaclass with all new defined stereotypes, the one with gray background was recently defined.

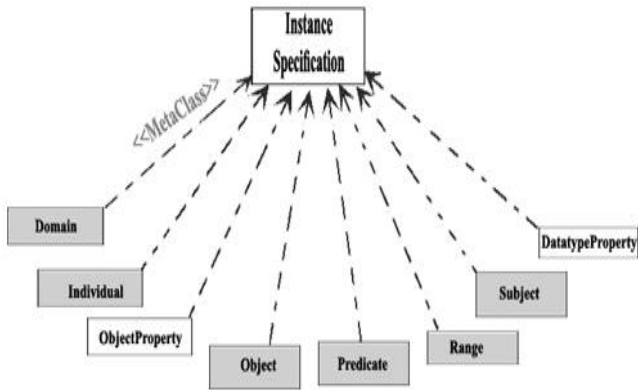


Figure 7: MetaClass InstanceSpecification extended concepts in the OMP.

As a statement is a coherent instances connected together, we used links (instanceSpecification with classifier association) to connect those three elements. We defined two stereotype link, <<domain>> to connect subject and predicate and <<range>> to connect predicate and object.

Slots are instances of Properties, either used in Class as attributes or in association as Association End. Slots in our profile plays an important role. Although the OUP has identified individuals, it did not shows how to create instances of data type properties or object properties. According to ODM we defined three stereotyped slots, as shown in Figure 8. The first is <<DatatypeSlot>> which links individuals with datatypeproperties instances value. The second and the third types are <<subjectSlot>> and <<objectSlot>> which define the subject and object of a statement respectively.

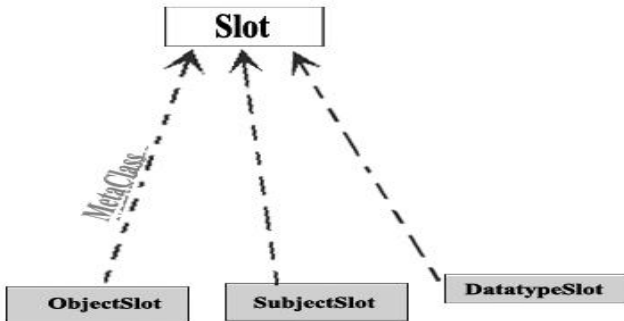


Figure 8: MetaClass Slot extended concepts in the OMP.

Another light addition is the adding of some tagged values to the object Properties. The one who notices the mapping between OWL:ObjectProperty and UML:Association, will know how importance to know if such relation is composition or aggregation. So we added two main boolean Tagged values to the ObjectProperty Stereotype. ISAggregation and IsComposition to define if the relation such as (has\_a) is composed or aggregated.

The next section describe a simple case study representing part

of the museum ontology and its properties. The museum is being shown in its class diagram in Figure 9.

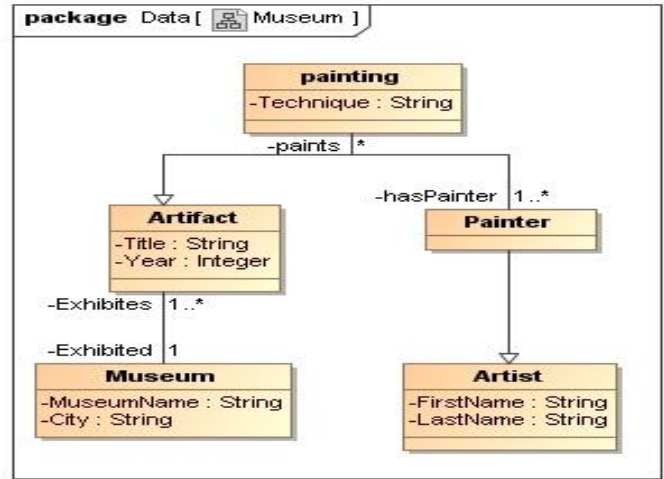


Figure 9: Part Of The Museum Class Diagram .

## 6. CASE STUDY

The case study below shows how OMP is applied on the museum class diagram. The UML classes we used in the use case are "museum ,painting, painter". Using the OUP, the owl:classes are painting, painter and museum, each represent a concept in the domain. Following the OUP method to define property we see that for example, class Museum is the domain of two Data type properties , MuseumCity and MuseumName which are stereotyped <<Domain>>, as shown in Figure 10.

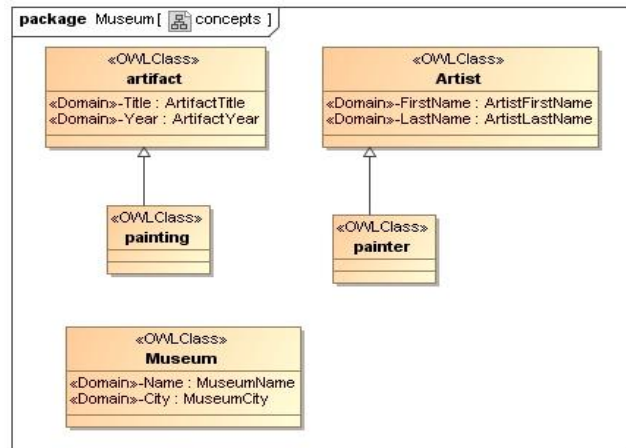


Figure 10: OWLClasses from the Museum Class Diagram .

To represent Data type properties, we stereotyped UML classes as << Data Type Property>>, as shown in Figure 11. The property stereotyped <<range>> within the MuseumName, for example, link it with the range, which is in this case , The primitive data type String.

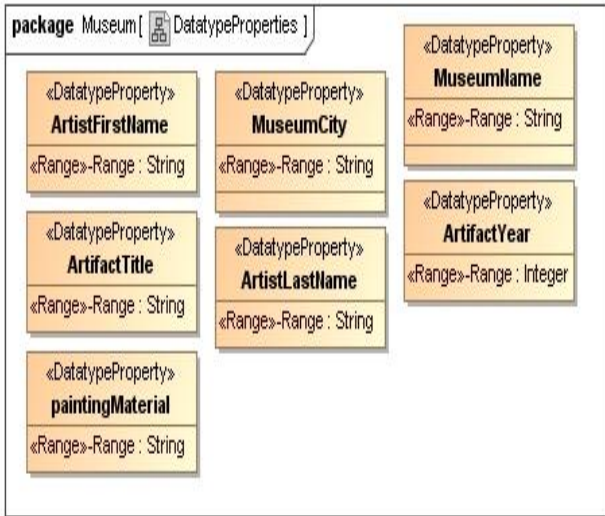


Figure 11:DatatypeProperties from the Museum Class Diagram .

Unlike datatypeProperties, object properties are connected with their domains and ranges, using associations stereotyped <<domain>> and <<range>> respectively, as shown in Figure 12.

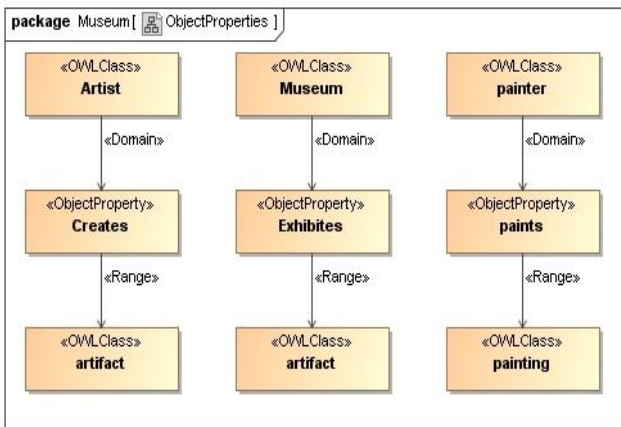


Figure 12:ObjectProperties from the Museum Class Diagram.

Individuals in OWL represent instances of owl:classes. They are either subject or object as shown in the Figure 13. Object properties instances are predicates(if they participate in a statement) that connect subject with object.

Note that in our approach data type properties do not participate as a part in the OWL statement constructions. They are used to fill in the slots value of owl class. For example, the instance of owl:class painting named Autoportrait has 2 data type slots, one is title which was given the value of an instance of a data type Property class named AutoportraitTile , and the second is year which was given a value of the instance of data type property named year as shown in Figure 13.

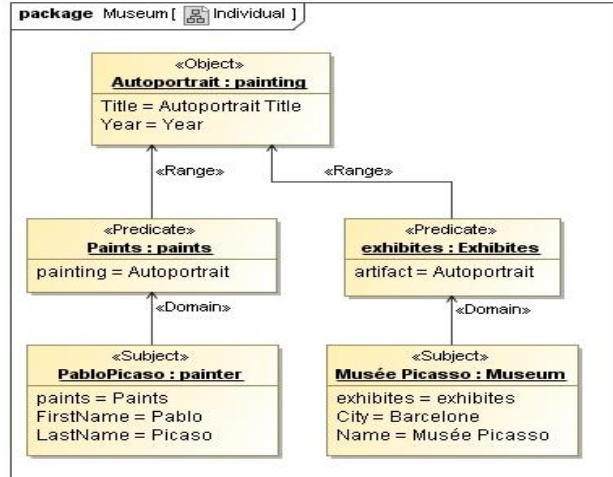


Figure 13: Individual from the Museum Class Diagram.

For more clarifying lets take the statement <painter - paints - painting> as an example. Using Individuals it will be interpreted as <Pablo Picasso - Paints - Autoportrait>. For subject slot, the navigable link stereotyped <<Domain>> between the subject Pablo Picasso Painter has an association end property referring to the subject of the statement, which is Pablo Picasso. We stereotyped this owned end property with <<SubjectSlot>> . The object slot is not owned by the link, rather, it was owned by the predicate, in this case Paints. This slot that has the same type as the object , which is the painting Autoportrait is stereotyped <<ObjectSlot>>.

## 7. CONCLUSION

The Ontology Modeling Profile, that was extended from the OUP, together with ODM are in accordance with the OMG’s RFP initiative for ontology modeling. We concern the individual modeling with statement constructing elements. This profile enables using ontologies in the way that is closer to software engineering practitioners. Also, since the UML and ODM are defined as MOF-compliant languages it is possible to store ontologies in MOF-based repositories, to store ontology diagrams in a standard way (UML2 XMI), as well as to share and interchange ontologies using XML. With the Ontology UML Profile, the ODM concepts can be used as stereotypes in the UML models.

A practical implementation for mapping between Ontology model and Requirement model will be carried out using the OMP to show how this profile was able to converge between MDA standards and the Web Ontology Language. We plan to extract the requirement analysis model from the domain model (ontology), accordingly we aim to use model transformation from the Ontology Model (OM) to the Requirement Analysis Model (RAM). We also aim to make automatic generation of the Web Ontology Language (OWL) from UML model based on our profile (OMP).This conversion transforms an ontology from its OMP into OWL description. Accordingly, this generated OWL model can be shared with ontological engineering tools (i.e. Protge).

## **8. REFERENCES**

- [1] S. Bran. The pragmatics of model-driven development. *IEEE Software*, vol. 20, no. 5, pp. 19-25, 2003.
- [2] M. D. BUI. *Real-Time Object Uniform Design Methodology with UML*. Springer, 2007.
- [3] C. Coral, R. Francisco, and P. Mario. *Ontologies for Software Engineering and Software Technology*. Springer-Verlag Berlin Heidelberg, Francis, 2006.
- [4] S. Cranefield. Uml and the semantic web. In *Proceedings of the International Semantic Web Working Symposium*, Palo Alto, 2001.
- [5] D. Djuric, Gasevic D., Damjanovic V., and Devedzic V. *MDA-Based Ontological Engineering*, pages 203{231. World Scientific Publishing Co., Singapore, 2005.
- [6] D. Djuric, D.and Gasevic and V. Devedzic. *Ontology modeling and mda*. *Journal of Object Technology*, vol. 4, no. 1, pp. 109-128, 2005.
- [7] D. Dragan, G. Dragan, D. Vladan, and D. Violeta. A uml profile for owl ontologies. In *Proc. of the Workshop on Model Driven Architecture: Foundations and Applications*, pages 138-152, Linkoping University, Sweden, 2004.
- [8] D. Dragan, G.and Dragan and D. Vladan. *Model Driven Architecture and Ontology Development*. Springer-Verlag Berlin Heidelberg, 2006.
- [9] D. Gasevic, D. Djuric, and Devedzic V. Bridging mda and owl ontologies. *Journal of Web Engineering*, 4(1):119-135, 2005.
- [10] Hans-Jo"rg Happel and Stefan Seedorf. Applications of ontologies in software engineering In *International Workshop on Semantic Web Enabled Software Engineering (SWESE'06)*, November 2006.
- [11] Thomas C. J. Just what is an ontology, anyway? *IT Professional*, 11:22-27, 2009.
- [12] Z. P. Je\_ and H. Ian. Metamodeling architecture ofweb ontology languages. In *Proc. of the Semantic Web Working Symposium*, pages 131-149, July 2001.
- [13] K. K. Kenneth, B.and Mieczyslaw, A. K Paul, H. Lewis, S. Jefferey, S. H. William, Jerzy Letkowski, and L. Aronson Michael. Extending the unified modeling language for ontology development. *Software and System Modeling (SoSyM)*, vo. 1, no. 2, pp. 142-156, 2002.
- [14] D. Li, K. Pranam, D. Zhongli, A. Sasikanth, and J. Anupam. Using ontologies in the semantic web: A survey. Technical report, UMBC, July 2005.
- [15] G. Matthias. What is mdd/mda and where will it lead the software development in the future? May 2007.
- [16] K. S. Michael, W. Chris, and L. M. Deborah. *Mda guide version 1.0.1*. <http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf>, 12 June 2003.
- [17] K. S. Michael, W. Chris, and L. M. Deborah. *Owl web ontology language guide*. <http://www.w3.org/TR/owl-guide/>, 10 Feb. 2004.
- [18] N. Moreno, J. R. Romero, and A. Vallecillo. An overview of model-driven web engineering and the mda. pages 353-382. 2008.
- [19] B. Sara, V. Raphael, E. Andreas, and L. Peter. Visual modeling of owl dl ontologies using uml. In *The Semantic Web ISWC 2004*, pages 198-213. Springer, 2004.
- [20] S. C. Seria, C. B. Sabin, C. Liliana, and C. N. Ovidiu. Survey on web ontology editing tools. <http://thor.info.uaic.ro/~busaco/publications/articles/web-ontology-tool-survey.pdf>, 2006.
- [21] C. Stephen and P. Martin. Uml as an ontology modelling language. In *Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 46-53, 1999.
- [22] J. M. Stephen, N. C. Anthony, and F. Takao. Model-driven development. *IEEE Software*, pages 14-18, Sep./Oct. 2003.
- [23] F. van Harmelen. The semantic web: what, why, how, and when. *IEEE Distributed Systems Online*, vol. 5, no. 3, pp. 1-4, 2004.
- [24] X. Wang and C.W. Chan. Ontology modeling using uml. In *Proc. of the Seventh International Conference on Object Oriented Information Systems (OOIS)*, pages 59 -70, Calgary, Canada, Aug. 2001.
- [25] V. Siricharoen Waralak. Ontology modeling and object modeling in software engineering. *International Journal of Software Engineering and Its Applications*, vol. 3, no. 1, pp. 43-60, 2009.