

# Building an Ontology for the Metamodel ISO/IEC24744 using MDA Process

**Mehdi Mohamed Hamri**

EEIDIS Laboratory, Computer science department, Djillali Liabes University, Algeria  
Email: hamri.m.m@gmail.com

**Sidi Mohamed Benslimane**

Ecole Supérieure en Informatique de Sidi Bel Abbes, Algeria  
Email: s.benslimane@esi-sba.dz

**Abstract**—The idea of using ontologies in the field of software engineering is not new. For more than 10 years, the Software Engineering community arouse great interest for this tool of semantic web, so to improve; their performance in production time and realisation complexity on the one hand, and software reliability and quality on the other hand. The standard ISO / IEC 24744, also known as the SEMDM (Software Engineering – Meta-model for Development Methodologies), provides in a global perspective, a conceptual framework to define any method of software development, through the integration of all methodological aspects related to the followed procedures, as well as, products, people and tools involved in the conception of a software product. The purpose of this article is to create domain ontology for ISO / IEC 24744 using an MDA process. This ontology will serve as semantic reference in order to assist for a better interoperability between the different users of the standard (human, software or machine).

**Index Terms**—ISO/IEC 24744, Ontology, MDA, OWL, Software Process, Metamodel, UML.

## I. INTRODUCTION

The development of means of transportation, telecommunication and other technological means, relies on the use of more and more complex software, which has to respect very demanding requirements furthermore, according to Rida Noor, “The success of software industry lies in developing a defect free and quality product within reasonable time and budget.”[1].

Software Engineering had to meet these requirements while increasing the level of abstraction of the methods, which has, for example, led to the emergence of Model Driven Engineering (MDE). Software engineering gives response to problems involving information share and exchange between great numbers of entities. The representation of such information has to be consensual, not only in order to facilitate exchange and communication (and so permit those entities to collaborate), but also to avoid issues of both integration and interoperability provoked by the use of

heterogeneously represented structures in which every entity has a specific meaning.

The MDE approach is presented as a solution to interoperability problems. It tries to give a conceptual meaning to software artefacts. However, due to often reached complexity levels, it is difficult to obtain a thorough conceptualisation of the artefacts. Accordingly, Jin and Cordy [2] proposed an ontological approach to represent Software Engineering knowledge.

As Uschold has asserted in [3], ontologies consist of three main categories of use: Communication, Interoperability and Systems engineering, this latter, is used to assist conception processes, and to maintain software systems, either the systems are based on knowledge or not.

Ontologies are interesting while facing exchange and interoperability problems, they proved to be a solution for conception and modelling of complex systems. Ontology is a philosophical branch that studies what exists [4]. Aristotle employed it to describe the existence of beings. In another context, the domain of artificial intelligence broaches models of the world and so, it seized the term ontology to describe all what can be represented from the real world in a program, and thus facilitate knowledge share and reuse.

In addition to the artificial intelligence research community, ontologies interest numerous domains relative to knowledge. Therefore, a high interest towards ontologies is found in Information Systems (IS) and Knowledge Based Systems (KBS). KBS and IS are concerned by the mastery of the domains and the resolution of knowledge problems, for both, ontologies are means of analysis, modelling and knowledge domains implementation [5][6].

The consensual aspect of an ontology is central. Every ontology has to be the result of a process in which all the operators of the domain cooperate and understand each other regarding the meaning given to every concept relative to the domain, so that ontology can be adopted by everyone and used as a common basis for the mastery of the domain. According to Isotani [S. Isotani] ontologies are applied to address three major challenges in software engineering, namely, difficulty in communicating and sharing information at first, secondly, effective

management of software development phases and finally, development of techniques and environments in order to support the production of semantic software through an interdisciplinary approach.

In the domain of Software Engineering, the norm ISO/IEC 24744 [8][9] has been strongly inspired by the works of Henderson-Sellers on OPF meta-model (OPEN Process Framework) [10]. These works have emerged as result to strong critique of the first versions of OMG's Software Process Engineering Meta-model (SPEM) [11], especially shifting mechanisms from a modelling level to another, for example from M2 to M1 to M0 [12][13]. To reach the limits of SPEM, the normalised meta-model ISO/IEC 24744, presents a new modelling approach, based on three aspects that are Process, Products and work Producers, which implies two kinds of users, developers and method engineers. Though this norm presents several advantages, it may seem intimidating because of the adoption of new concepts that constitute a paradigm change, which implies a new way of thinking and perceiving meta-modelling processes. The idea of using ontologies as semantic references for the norm ISO/IEC 24744, can be a good solution for helping users to surpass the important learning curve of this new paradigm. It permits better comprehension and communication between different users of the norm. In this article, we will propose an MDA-based approach [14] that will permit the elaboration of a domain ontology for the standard ISO/IEC 24744.

The rest of the present article is organized as follows: the next section presents a brief review of the literature that is most related to our work. The third section gives a global view of the norm ISO/IEC 24744 and details the most important concepts of our work. The fourth section introduces our construction architecture of the norm ISO/IEC 2744. In section five, we present the profiles construction steps and transformations in OWL. Section six illustrates the experimentations realised to demonstrate the feasibility of our approach, it is followed, in section seven, by a comparison with a related work. Finally, we conclude and mention perspectives for this work.

## II. RELATED WORKS

Using ontologies in Software Engineering is not a new idea. During the last decade, several works have been undertaken. We will mention the most relative ones to our work.

[15] Defined a generic ontology for process domain considering, it resulted in a contribution of the recently built process ontology to the semantical enrichment of the terms for measurement and evaluation domain ontology by means of stereotypes.

Gazel et al [16] presented an ontology-based tool for software process evaluation, this tool was developed in order to support the phase of collecting data of the evaluation process and to assess software processes, in conformity with CMMI [17]. Rungratri and Usanavasin

[18] proposed «CMMI-GAAF v.1.2» a framework for an automatic analysis of the gaps relative to the lack of conformity with CMMI, using the work of Soydan et.al. [19] Soydan and Kokar. In this context, Project Assets Ontology was developed, so to merge CMMI process and project assets. Lee et al [20] have proposed an ontology-based multi-agent system for the evaluation of CMMI, this system permits the establishment of evaluation links on both processes and product quality in conformity with CMMI. Liao et al [21] aimed to create a generic Software Process Ontology (SPO), with the objective to make it meet the requirement of both CMMI and ISO/IEC 15504, SPO represented by atomic practices, was used in order to represent models of process. Liška and Návrat [22] brought an approach that facilitates the use of Software Process Meta-Model in order to convey improvements by means of knowledge engineering approaches, they developed an SPEM ontology in order to demonstrate its technical application on the semantic web. In [23] Falbo presented an SPO aiming to establish a common vocabulary for software process, in order to support organisations working on improving software process, Falbo et al [24] they also developed an ontology model for software process, according to the authors, this ontology is sufficiently expressive to be used as a common base for mapping software process fragments of norms such as IEC 12207-ISO 9001:2000-ISO/IEC 15504, CMMI, RUP and SPEM. Wongthongtham et al. [25] presented an ontology model for software engineering, to represent software engineering knowledge, thus helping to define utilised information during exchange in semantic projects, it is also used as a communication frame. Guizzardi et al. [26] presented the last development relative to Unified Foundational Ontology (UFO), this latter is useful for improving the quality languages and models of conceptual modelling. Mendes and Abran [27] proposed an ontology prototype based on the SWEBOK [28] guide that is able to represent Software Engineering data, an extraction from the guide resulted in almost six thousand (6000) Software Engineering concepts, and about four hundred (400) types of relations between the concepts have been identified. Sicilia et al [29] have also proposed an ontology based on the SWEBOK guide, it contains; a descriptive part that identifies artefacts and activities and a prescriptive part containing approaches for commonly accepted concrete activities.

Hilera et al [30] proposed OntoGLOSE, another ontology that conceptualises the domain of software engineering, based on the Software Engineering Terminology Glossary and published by IEEE, It includes more than 1500 concepts corresponding to 1300 glossary term, each one with its different meanings. Henderson-Sellers et al [31][32] proposed initiatives aiming to standardize the different ISO norms, such as the project intended by the SC7 ISO study group. Initiatives from both parts often resort to other ontologies in order to deal with the problem of semantic interoperability. As a result, we need high quality ontologies as bases for integration. Standardization is consolidated as a source of knowledge

that reflects a common conceptualisation. However, most of the standards have not been designed to be ontologies. Ruy et al [33] presented an ontological analysis of a part of the meta-model ISO/ IEC 24744, this analysis is accomplished in the light of UFO[34], the authors have identified some problems and underlined some solutions. Suggestions and general recommendations were presented in order to transform SEMDM into an ontology of quality in accordance with the needs of ISO standardization.

[35]Souleymane Koussoube ET AL proposed an ontology-based contribution to Agent Oriented Software Engineering called OBAMAS (Ontology-Based Approach for Multi-Agents Systems engineering). It starts with an analysis phase that consists of the construction of three formal ontologies (a domain ontology, an ontology of functionalities, and an ontology of multi-agents systems) and their alignment to merge in a single one. Secondly, a design phase consists of the operationalization of the single ontology in order to infer the agents of the system in a more formal way.

In this section we have mentioned a range of works aiming to use ontologies in the domain of Software Engineering, nevertheless, Liska’s work[22] is the closest to the present approach, and we will compare both approaches in the seventh section.

### III. THE NORM ISO/IEC 24744:2010

The norm ISO / IEC 24744, commonly called SEMDM (Software Engineering-. Meta-model for Development Methodologies) defines a meta-model for development methodologies. In this context, a meta-model indicates a semi-formal language able to describe methods. These methods are themselves models, which are similar to other meta-models such as SPEM of OMG. The norm is strongly inspired from Henderson- Sellers’ works on the OPF meta-model [10]. The standard ISO/IEC 24744 has emerged after proposing a solution to duality problems caused by OMG’s four layers, which occur when an element defined in a layer is at the same time; object (the

element is the result of the instantiation of the elements of the higher layer) and class (the element s subject to an instantiation ). The standard ISO/IEC 4744 permits the possibility to bypass this problem through the use of two concepts, namely Power type and Clabject [36]. The elements of the norm are organized by the communities involved in their creation and use (Fig 1). Methodologies are created by method engineers; they will be used by software developers to create software products. Both communities intervene in three different domains i.e. metamodeling, methodology and Endeavour. Every domain is a representation of the domain “below”, which means that, methods represent Endeavour and meta-models represent methodologies.

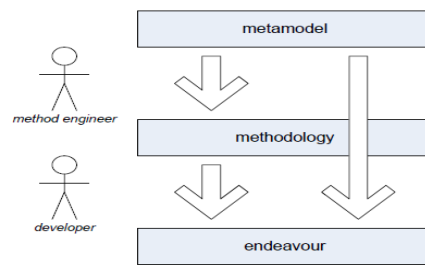


Fig.1. The Three Utilization Domains of the Norm ISO/IEC 24744

SEMDM sorts the elements of a methodology in three categories; Endeavour elements, that represent the elements to be created by the developers in a project, Template elements, representing all the elements created by method engineers during the construction of an ontology, Resource elements, representing the elements of a methodology directly used at the level of project without instantiation(Fig 2).

The elements Endeavour, Template and Resource provide all the necessary classes to proceed with the three aspects of methodologies; Process, Product and Producers. Fig3 gives a partial view of the meta-model Endeavour Element with the different links that exist between the three elements (Process, Product and producers)

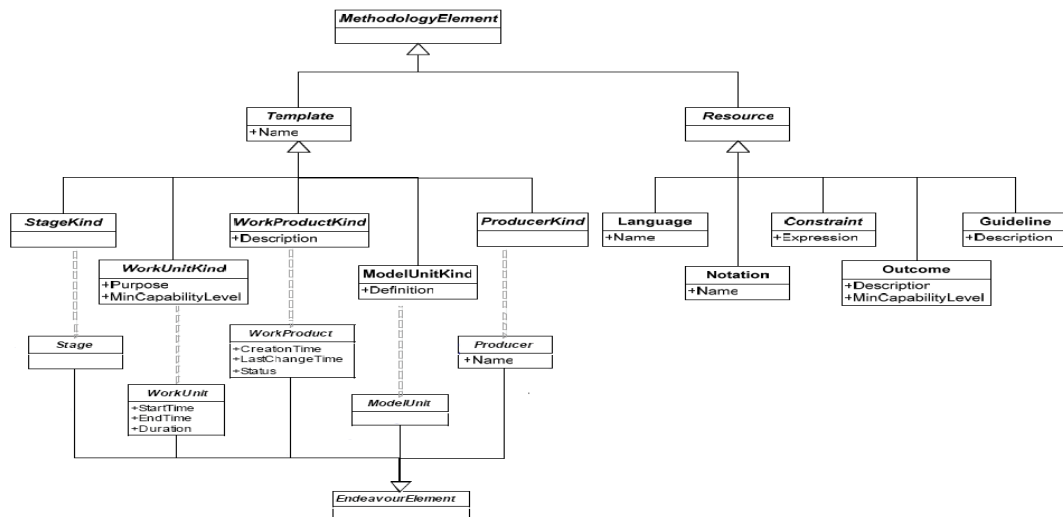


Fig.2. Partial Class Diagram of ISO/IEC 24744 Meta-Model.

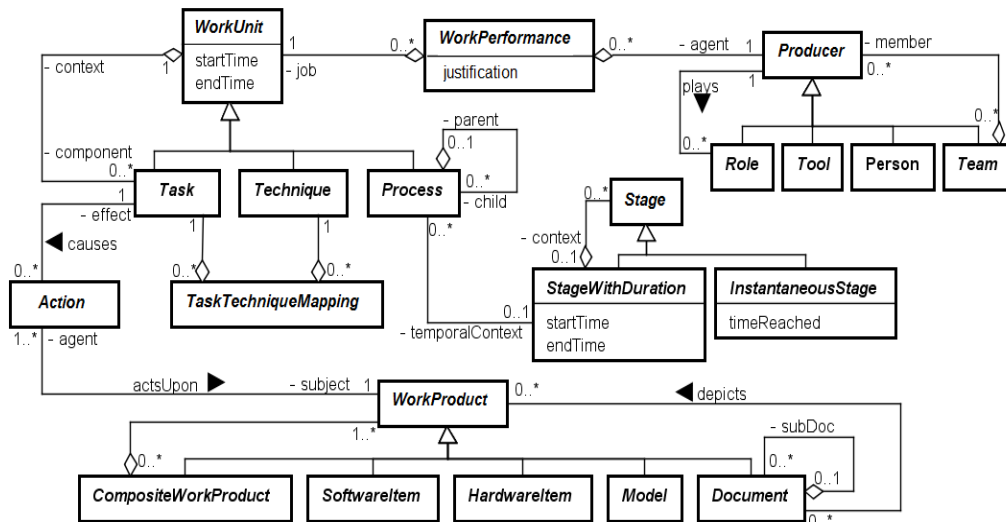


Fig.3. Partial Class Diagram of the Endeavour Elements of ISO/IEC 24744 Meta-Model

The Endeavour element contains five main classes, which are, Work Unit class; that processes works that are realised or intended to be realised, Stages class; that concerns the management of time limits, Producers class; it concerns agents in charge of executing Work Units, Work Producers class; that refers to utilised artefacts and Model Units class that treat utilised models.

#### IV. CONSTRUCTION ARCHITECTURE OF THE ISO/IEC 24744 ONTOLOGY

Though the advantages presented by the meta-model SEMDM are undeniable [37], its use is sometimes intimidating. The divergence of conventional approaches and the adoption of new concepts constitute a paradigm change, which implies a new manner of thinking and perceiving meta-modelling process, and so the learning curve is likely to be important. In addition to that, the fact that SEMDM is specified in terms of purely abstract concepts, without offering a specialised notation that permits a simplified manner of modelling methods, makes the conception and the validation processes of the method more difficult. Using ontologies as a semantic reference can be a good solution that helps users to overcome such problems.

Although the norm ISO/IEC 24744 has various advantages comparing to OMG approaches such as SPEM in the domain of Software Engineering, it is not to deny that MDA has savoir-faire and experience in model-based ontological engineering domain, furthermore, the duality problem of the MDA approach is irrelevant in this situation, since our objective is the creation of a domain ontology that allows the definition of the concepts of the norm and their relations. In addition, the norm uses UML[38] class diagrams to define its meta-models, which facilitating the use of both the MDA approach and its widely available tools.

##### A. Ontologies modelling architecture in an MDA process

For the creation of our MDA-based ontology, our

approach is inspired from the work of Djurić et al [39] in its transformation stages. Before beginning the transformation, we first extract, three separated meta-models, each one respectively corresponding to the concepts Endeavour, Template and Resource. The extraction is based on the specification of the meta-model. Secondly, we create a UML profile for the norm, it gathers three packages, each one corresponding to each of the three extracted meta-models. This profile will be used as starting point for our MDA-based transformation process. As for the validation of the domain OWL ontology, there exist numerous validation tools in the related literature. In our work we use the Protégé ontological tool.

In this work, we propose an MDA-based architecture for ontology engineering that contains (see Fig 4):

- A UML profile for the norm ISO/IEC 24744 (UPIISO).
- An ontology UML profile for ISO (OUIISO)
- An Ontology Definition Meta-model (ODM) [40].
- Semantic web languages (OWL, RDFs, etc.).

Ontology Definition Meta-model stands as a link between MDA on the one hand and both semantic web and Software Engineering on the other hand, which makes of ODM the core of this architecture.

Therefore, our architecture is an MDA-based ontological architecture that includes a UML profile for the norm ISO/IEC 24744 (UPIISO). To build this UML profile, we first choose the package containing all the stereotypes that allow us to export the profile towards the target meta-model PSM which can be OWL. Afterwards, we exploit ontologies to model the semantic aspects of the norm in order to define an ontology UML profile that will be translated in an ODM. This meta-model gathers all the ontological concepts, it is defined according to Meta-Object Facility (MOF) [41] and its construction is based on OWL.

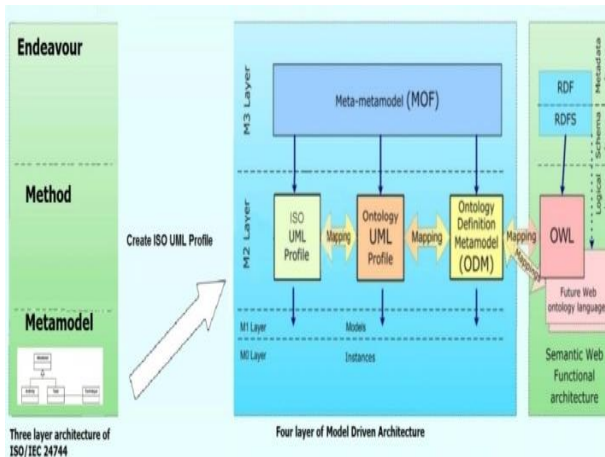


Fig.4. MDA Ontologies Architecture.

Once ODM built, it is translated towards OWL meta-model, for example to generate an OWL description. Consequently, the gateway between MDA and semantic web is created via the use of ODM. This offers numerous advantages such as flexibility and opening. Focusing only on the OWL meta-model, we will propose our approach to construct a domain ontology in the next section.

#### B. MDA based approach for the construction of a domain ontology

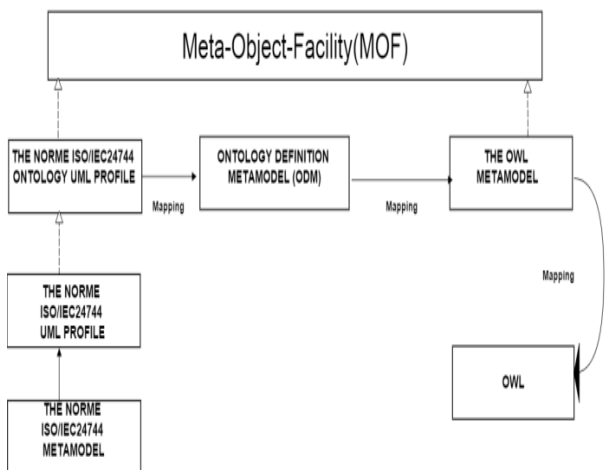


Fig.5. Construction Steps of ISO /IEC24744 Domain Ontology

Fig 5 depicts a global view of the steps to be followed during the construction of our ontology. The base of our architecture is the construction of UML profiles of ISO/IEC 24744, namely a UML profile for Endeavour, Template and Resource. These UML profiles are presented as follows:

#### C. UML profile of the norm (UPIISO)

The profile is the cornerstone of our work, because it permits passage to MDA space. It is a central part of our approach, because all the following steps of the work rely on it. This UML profile is used to model the semantic aspects of the norm. A group of UML extensions constitutes a UML profile. UML offers three extension mechanisms to extend the elements of UML's base meta-

model; Stereotypes which are classes of the profile define how an existing meta-class can be extended within the frame of a profile, while respecting base elements, tagged values that introduce properties and finally constraints that act as restrictions upon the new element.

We use a UML profile in order to mark the model that is independent from the platform, called PIM (Platform Independent Model), and to facilitate the profile's transformation into the target meta-model PSM (Platform Specific Model) which is in our case the OWL meta-model. During the following steps, we will need this PSM to generate our domain ontology in OWL. The creation of these profiles is not source to problems, because the meta-model of the norm ISO/IEC 24744 is defined in diagrams of UML OMG class. The results of this phase are the three profiles; Endeavour, Template and Resource. Endeavour profile is the most important; it will be composed of three main elements which are Product, Producer and Process. However to generate an OWL ontology of ISO/IEC 24744, we have adopted a methodology composed of numerous steps, these steps are described in the following section.

#### D. Generation methodology of a domain ontology

For an automatic generation of our ontology, we first extract information relative to the concepts (Endeavour, Template and Resource) from the description of the meta-model of the norm.

In the extraction process, we use the UML plug-in tools under the platform Eclipse (Eclipse, 2015) [42] in order to create the profiles Endeavour, Template and Resource that will respectively have as sources, the norm's meta-models Endeavour, Template and resource.

Once the profiles created, they will be stereotyped with the concepts of the norm. The profiles will be called UPIISO which stands for UML Profile of the norm ISO, they are in fact three UML profiles.

In order to generate an OWL domain ontology, we have used mappings between the concepts UPIISO, OUPIISO, ODM, OWL. For the validation of the generated ontology, we have used the Protégé tool (Ref). Fig 6 depicts the generation process of the ontology ISO/IEC 24744.

This process includes the following steps:

##### Step 1: Meta-model fragmentation

This step permits the extraction of the information relative to the concepts Endeavour, Template and Resource of the norm's description. This step is not quite complicated since it selects the different part of the meta-model that are relative to the three domains described in the norm which are; Endeavour, Template and Resource.

##### Step 2: UPIISO construction (UML profiles for the norm)

After fragmentation, we use the UML tool (Eclipse Modelling Tools and its Plug-in Papyrus) in order to create three UML profiles with stereotypes that include constructors of the norm. In this step, we obtain three profiles; Endeavour, Template and Resource.

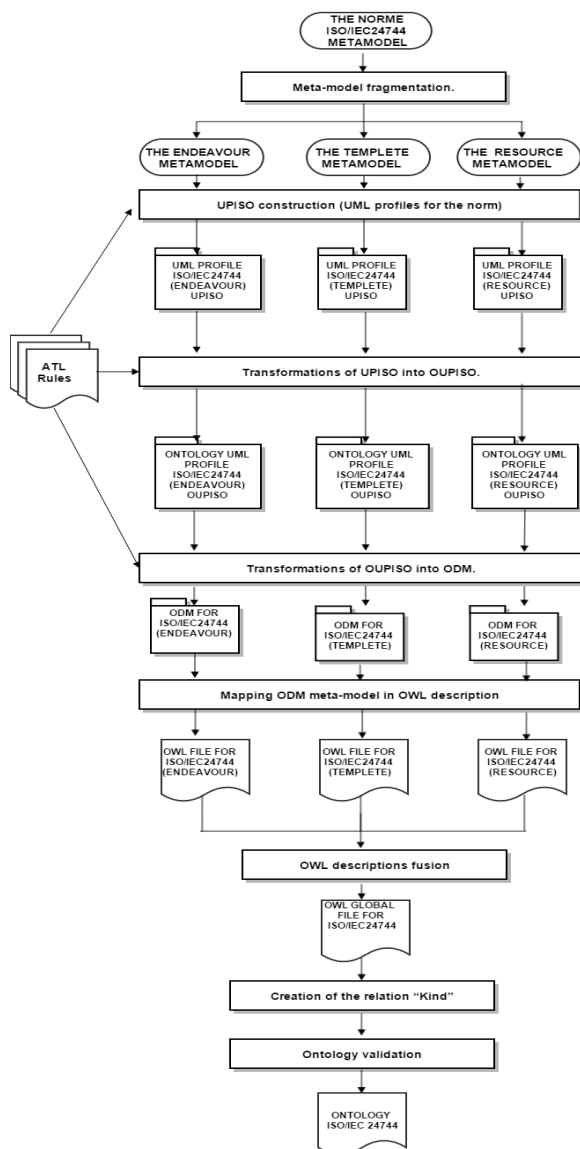


Fig.6. OWL Ontology Generation Process.

### Step 3: Transformations of UPIISO into OUPIISO

In this step, we transform these three UPIISO profiles in their corresponding OUPIISO profiles: an ontology UML profile for Endeavour profile (Endeavour Ontology UML Profile), an ontology UML profile for Templates profile (Templates Ontology UML Profile) and an ontology UML profile for Resource profile (Resource Ontology UML Profile).

### Step 4: Transformations of OUPIISO into ODM

At this level, the three OUPIISO profiles are transformed into their corresponding ODM. Since all the meta-models are in accordance with MOF (Meta-Object Facility) of MDA architecture, we first generate XMI documents (XML Metadata Interchange) for these meta-models; XMI formats for UML, OUPIISO for ODM. After that, we use ATL (Atlas Transformation Language) [43] to transform the source representations into their corresponding target representations. In the implementation section, we will mention the

transformation rules that have been applied in this part. Once these rules executed, we will obtain three XML representations of the ODM meta-models.

### Step 5: Mapping ODM meta-model in OWL description

The purpose of this step is to generate an OWL description of the norm. This stage is not quite difficult since the ODM concepts are almost identical to the OWL concepts. The utilised transformation rules are defined in Table 1. An example of transformation rule is defined in the implementation section.

### Step 6: OWL descriptions fusion

The sixth step consists of gathering the three OWL documents respectively corresponding to the profiles Endeavour, Template and Resource in order to create a complete OWL file corresponding to ISO/IEC 24744 domain ontology.

### Step 7: Creation of the relation "Kind"

This step consists of establishing a link between the concepts « element » and « element Kind » for the creation of a Kind non-taxonomic relation. The creation of this relation has been implemented using Jena, a Java API (Atlas Transformation) that permits ontologies creation and manipulation under OWL. In this stage, we browse the ontology and we create for every concept of the sub-ontology Endeavour a Kind relation with the corresponding concept of the sub-ontology Template, for example, the concept Workunit will be linked to WorkunitKind.

### Step 8: Ontology validation

The ontology validation stage allows the detection of syntax errors, or possible incoherencies in the final description of the OWL ontology. The XMI documents are exported in the Protégé tool for validation

## V. PROFILE CONSTRUCTION AND TRANSFORMATION IN OWL

### A. UML profile construction for Endeavour UPIISO

The first thing to do during this step is the construction of a package that gathers all the stereotypes of ISO/IEC 24744. These stereotypes correspond to the concepts of the norm. Fig 7 depicts a screenshot of the package of the stereotypes Endeavour elements.

Once the package built, we construct three UML profiles (UPIISO) corresponding to the modular specification of the norm concerning the three parts. In order to construct these UML profiles, we first have to select the stereotypes corresponding to each part Endeavour, Template and Resource starting from the package. Once elaborated, every UML profile is saved in its corresponding package. Since the number of transformations is important, we restrict our representation to certain construction and transformation steps of these profiles. Fig 8 depicts the profile UPIISO Endeavour.

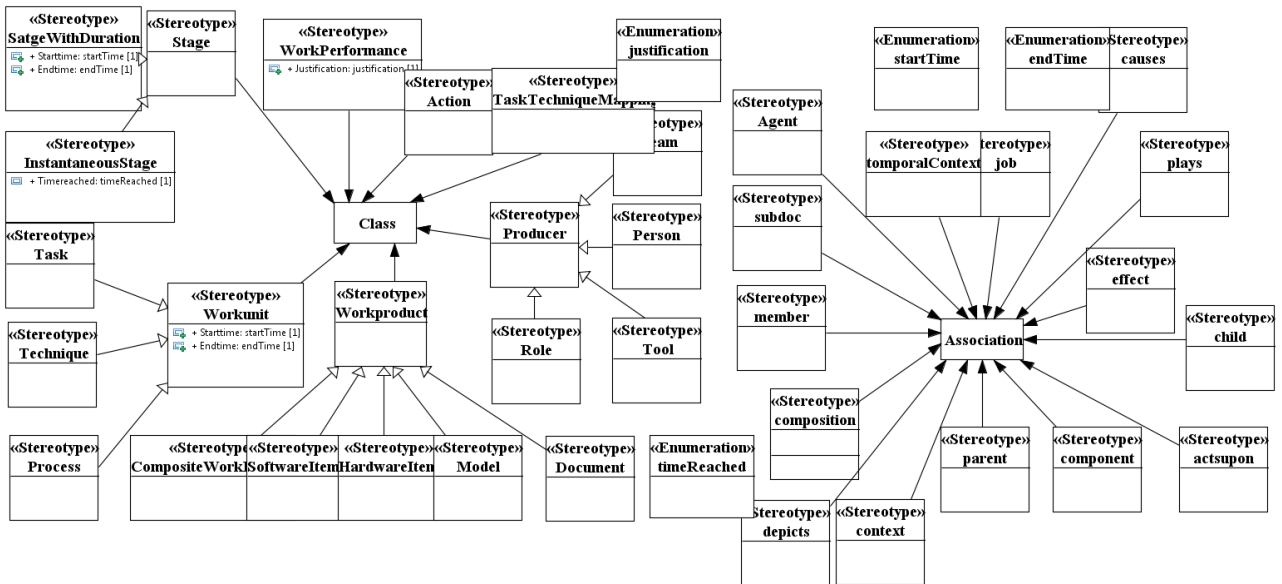


Fig.7. ISO/IEC 24744 Stereotype Package.

B. Transformation and validation

To realise models transformations, and to stay relevant to the MDA-based standardization context, we have used ATL, a language that meets the specification MOF 2.0 and QVT (Query/Views/Transformations), defined by OMG as a meta-model for models transformation (OMG, 2011b)[44].

The Meta-models UPIISO, OUPIISO, ODM and OWL

are first translated into XMI representations, after that we realise the transformations via ATL. These transformations concern the mappings from

UPIISO to OUPIISO then from OUPIISO to ODM and finally from ODM to OWL, at last, The Protégé ontological tool is used for the validation of OWL generated descriptions.

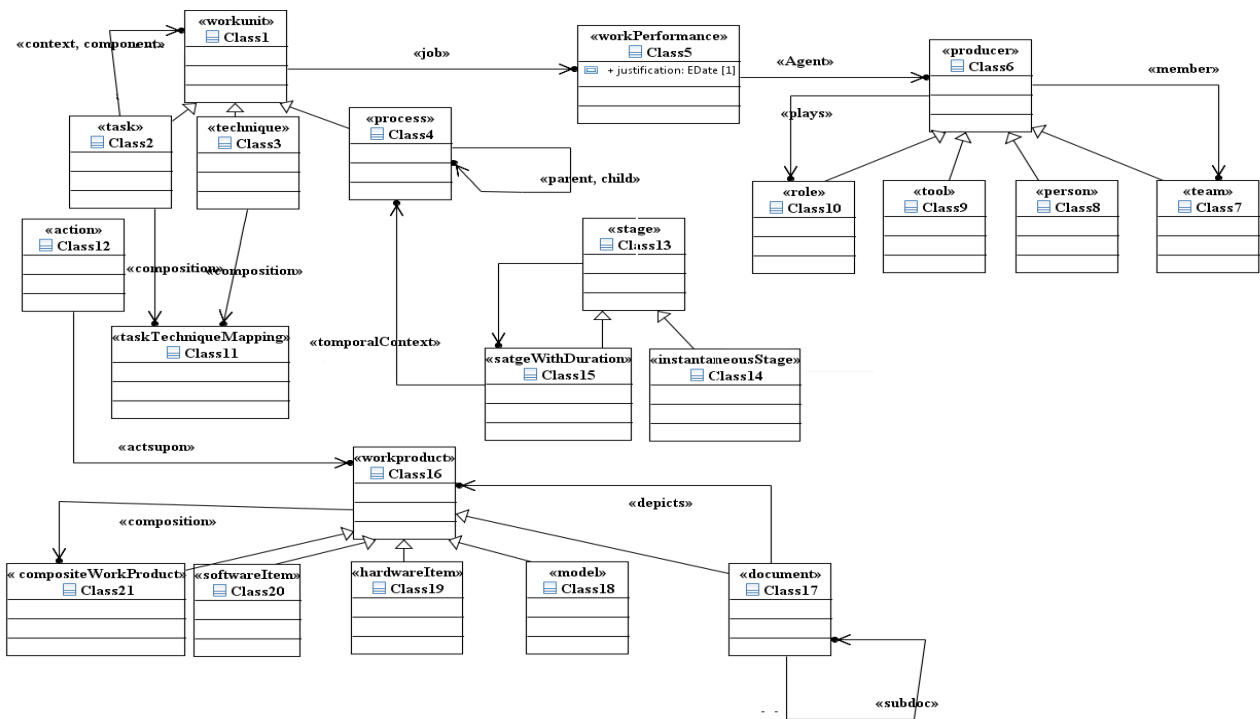


Fig.8. The Endeavour Profile.

a. Transformation of UPIISO into OUPIISO

The translation of UPIISO into OUPIISO is performed

by transformation tools that are defined in Table 1. UPIISO elements indicate ontological concepts in OUPIISO. Consequently, UPIISO extends UML with new constructors to support our specific ontological concepts

such as stereotypes classes <<Ontology>> and <<OntClass>>. We convert stereotypes classes to ontological classes with the stereotype <<OntClass>>. Tagged values corresponding to properties or attributes are transformed to ontological data with the stereotype <<DatatypeProperty>>. Finally, the stereotype <<Package>> becomes <<Ontology >> and the stereotypes association are translated to the concept <<ObjectProperty>>.

Table 1. Mapping UPISO and OUPISO

UPISO Concepts (Stereotypes or tags)	OUPISO Concepts (Ontological concepts)
<<Package >>	<<Ontology >>
<<Class >>	<<OntClass >>
<<Association >>	<<ObjectProperty >>
<<Attribute >>	<<DatatypeProperty >>

b. Transformation of OUPISO into ODM and OWL

The construction of Ontology Definition Meta-model (ODM) is based on OWL with ontological concepts.

Fig 9 presents a partial view of the ODM meta-model.

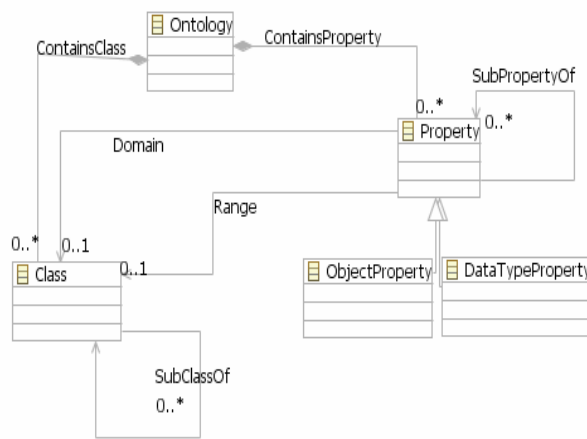


Fig.9. Partial view of ODM

The transformation rules from OUPISO to ODM and from ODM to OWL meta-model are quite simple and direct since ODM concepts are used as stereotypes within OUPISO on the one hand, and ODM is defined according to OWL concepts on the other hand.

The use of stereotyped classes <<Ontology>> and <<OntClass>> in OUPISO serves to identify respectively a domain and a concept. Therefore, they will be represented as classes in ODM. The stereotype <<ObjectProperty>> is a representation of the classes <<ObjectProperty>> in ODM and only with individuals from its domain and range. Lastly, the stereotype <<DatatypeProperty>> is considered as a class 'DatatypeProperty' in ODM. Table 2 depicts the mappings between the concepts OUPISO, ODM and OWL.

Table 2. Mappings between the concepts OUPISO, ODM and OWL.

OUPISO Concepts (Ontological concepts such as stereotypes or tag values)	ODM Concepts (ontological concepts)	OWL Concepts
<<Ontology >>	Class Ontology	OWL : Ontology
<<OntClass >>	Class Class	OWL : Class
<<ObjectProperty >>	Class ObjectProperty	OWL : objectProperty
<<DatatypeProperty >>	Class DatatypeProperty	OWL : datatypeProperty

VI. IMPLEMENTATION

The objective of our experimentation is to verify the feasibility of the process on which relies our approach. Our implementation and transformations are based on the Framework Eclipse and the plugin Papyrus. Our choice has been motivated by the fact that EMF[45] is at present the reference environment for Model Driven Engineering, because it proposes the implementation of the Meta layers for most of the tools derived from MDE. For model transformation, we have chosen a project among the most adopted by the Eclipse community, which is the transformation engine ATL.

This engine provides an implementation similar to OMG's standard Query View Transform (QVT) that supports both declarative and imperative enunciation of transformation rules, in our case, we have developed all UML profiles with the UML tool, which relies itself on the platform Eclipse. It will be noticed that models manipulations have been realised through MDE technologies and a part of their code has been generated by means of models.

We have implemented our profiles as Ecore format EMF meta-model, in order to use them at the beginning of the serialisation phase and in model transformation, their edition was completely realised in UML with the tool Papyrus. However, since Papyrus is a UML editor the meta-classes that it instantiates and serialises in these models are Meta-classes of the UML meta-model but not Meta-classes of the MOF meta-model. We had to convert UML models in into Ecore models; this format is the implementation of the MOF provided by EMF. To illustrate our MDA process of generation, which we have described previously, we will only present screenshots related to the elaboration of the UML profiles for OWL, their transformations and then the validation of the domain Ontology.

A. UPISO transformations to OUPISO

These transformations concern mappings of UPISO concepts to OUPISO concepts. OUPISO is a UML ontology profile for ISO/IEC 24744 and its concepts are ontological. Before undertaking any transformations, we first construct a package containing all the stereotypes that can support ontological concepts such as "Ontology", "Ontoclass", "ObjectProperty" and "DatatypeProperty". After that, the stereotypes that we create should support



ODM in order to facilitate mapping between OUPISO and ODM. Once the stereotypes created, the user is able to perform his transformations. Fig 10 depicts a screenshot of the stereotypes OUPISO package.

Once the stereotypes OUPISO created, we translate the UML profile (UPIISO) constructed before, to its corresponding OUPISO. In order to do it, we execute transformation rules defined in Table 2. We mention Rule 1 as an example that permits transformation of a class into an Ontoclass.

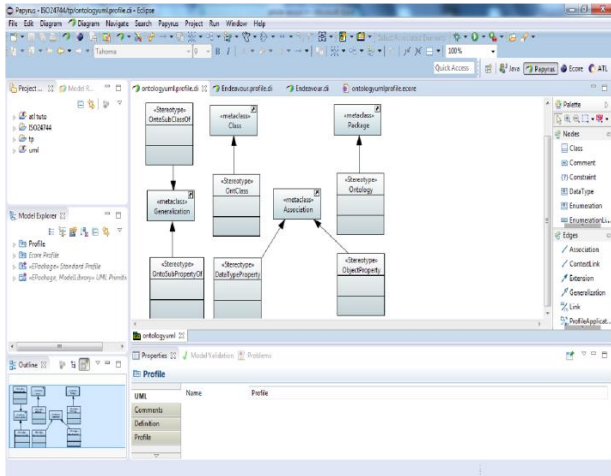


Fig.10. Screen Shot of the Stereotypes OUPISO Package.

**Rule 1:** transformation of a UPIISO class into an OUPISO class

```

1: rule Class2Class
2: {
3:   from
4:     a: UPIISO!Class
5:   to
6:     p: OUPISO!Class(
7:       name<-a.getDefNameSet(),
8:       generalization<-a.generalization
9:     on: OUPISO!OntClass(base_Class<-p)
10:  }

```

Fig 11 depicts a screenshot after the complete execution of the transformation rules that permitted the generation of the Ontology UML Profile OUPISO.

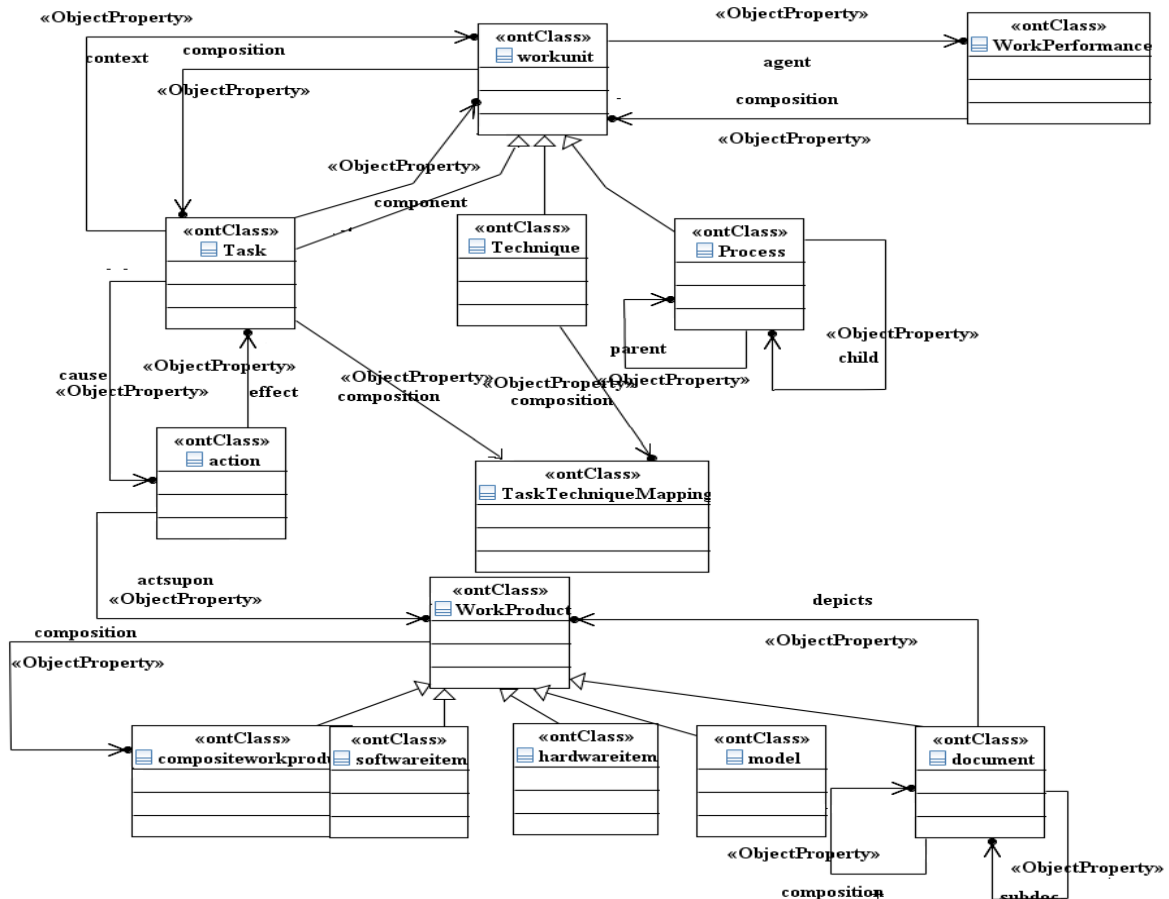


Fig.11. OUPISO Part Generated for Endeavour.

**B. Transformations from OUPISO to ODM**

The transformations of the UML Ontology profile (OUPISO) to ODM are simple because ODM concepts are used as stereotypes in OUPISO. After the execution of the rules corresponding to this part, we obtain the ODM generated Endeavour part (Fig 12). We mention as an example rule 2 that permits the transformation of an OUPISO class to an ODM class.

**Rule 2: transformation of an OUPISO class in an ODM class.**

```

1: rule Class2Class
2: {
3: from
4: a : OUPISO!Class
5: to
6: p : ODM!Class(name<-'Class',
    ownedAttribute<-pr,generalization<-
    a.generalization),
8: pr:ODM!Property(name<-a.name)
9: }
    
```

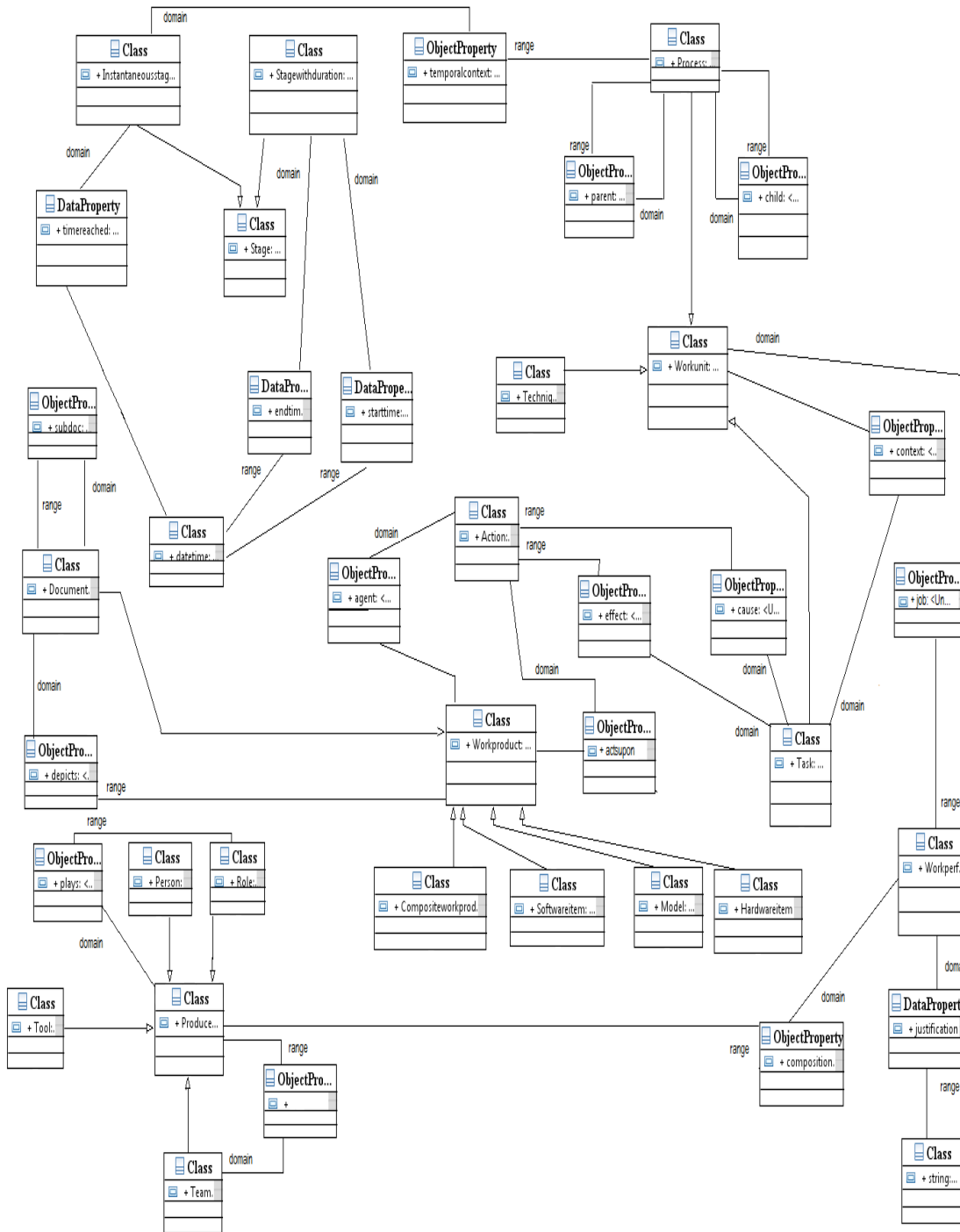


Fig.12. The ODM Part Generated for Endeavour.

### C. Transformations from ODM to OWL

Once the three ODM parts generated, we execute the transformation rules corresponding to this phase, in order to generate corresponding OWL parts. These transformation rules are already defined in table2. After executing ATL rules, we obtain an OWL description that we present in Fig 13. We mention as an example rule 3 that permits the transformation of all ODM classes in OWL classes.

---

**Rule 3:** ODM classes transformation in OWL classes.

---

#### helper context

Class ! package def:

#### GetCodeOwl

```
(| :String = Class!Class.allInstances()-
>iterate|C;acc :String="|acc +'<owl:' +C.name+' rdf:ID="")
```

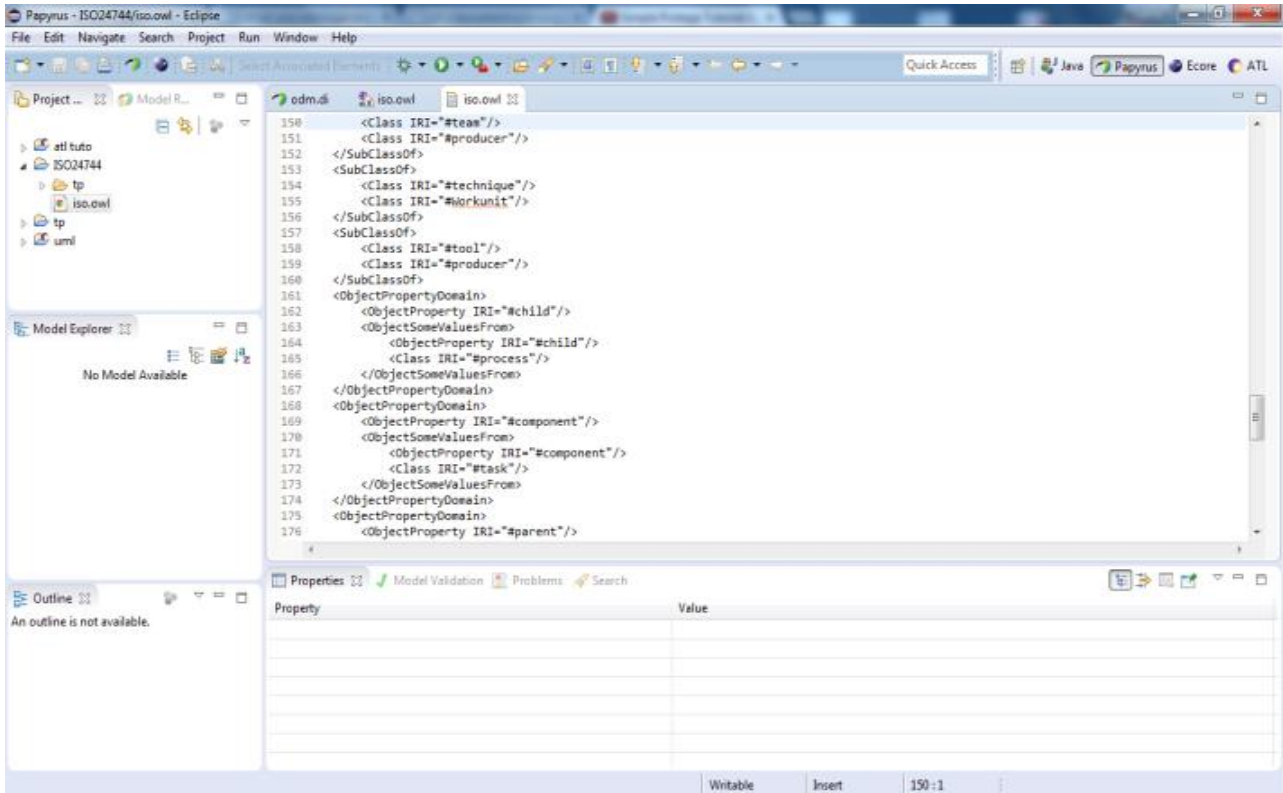


Fig.13. The OWL Part Generated for Endeavour.

### D. Fusion of the generated OWL descriptions

In this final step, we fuse the three OWL descriptions generated in the preceding steps in a single OWL file. We use API JENA to generate links between the elements Endeavour and Template with the relation Kind. The OWL file obtained will be exported to the tool Protégé for validation.

## VII. COMPARISON

We have compared our work with Liska's work [22], this latter aims at creating an ontology for the method SPEM 2.0. We did not intend to compete with his works for the simple reason that we treat different methods, but we intended to reuse his contributions in order to improve our approach. From an architectural perception, our work is quite similar to Liska's work, since both approaches have used Djuric's architecture for ontologies creation.

Both articles propose the transformation of an MDA norm into Semantic Web with a mapping between an ontology UML profile and an arbitrary UML profile. The existence of similarities between the models «SPEM and ISO/IEC 24744 » motivates us for this choice.

Our approach permitted the creation of a UML profile for the norm, while Liska's work had the advantage of having SPEM UML profile with formal OMG specification. Though Liska's work had this advantage since SPEM originates from MDA with an XMI specification, which permits him to use directly modelling tools such as Eclipse, ATL and Ecore, the author preferred writing mapping tools in Adhoc. However, in our approach we use passage rules with ATL in order to generate mappings automatically. Furthermore, we have preferred modularity by using the profiles Endeavour, Template and Resource, in order to facilitate maintenance and evaluation, which has not been done in the other works. Liska's work has been a reference for us and a great source of inspiration though the objectives are

not specifically the same, but the main purpose was to create a domain ontology for Software Engineering.

### VIII. CONCLUSION

The main purpose of this work consists of developing a domain ontology of the norm ISO/IEC 24744, this ontology has the objective of defining the concepts relative to the norm. The use of the MDA approach for the construction of ISO/IEC 24744 ontology proved to be audacious since ISO/IEC 24744 with its three layers has been created to fill the lacks of the MDA approach. An intermediary objective can be considered as another contribution is the creation of a UML profile for ISO/IEC 24744, this profile will serve as reference to other works relative to the domain. Other experimentations will be necessary in order to confirm that the proposed ontology is able to meet the user's needs. Unfortunately, it was difficult to find industrial partners that are familiar with the norm ISO/IEC 24744 and ready to experiment our Ontology. We must recognise that our research has to continue in order to check and validate the approach in real use situations.

As a perspective we will work on OCL constraints that contain semantic[46] (OMG, 2014b), which allows us to reason on the obtained OWL description.

### REFERENCES

- [1] Rida Noor, Muhammad Fahad Khan, "Defect Management in Agile Software Development", IJMECS, vol.6, no.3, pp.55-60, 2014. DOI: 10.5815/ijmeecs.2014.03.07.
- [2] Jin, D. and J. R. Cordy. "Ontology-based software analysis and reengineering tool integration: The OASIS service-sharing methodology". Software Maintenance. ICSM'05. Proceedings of the 21st IEEE International Conference on, IEEE, 2005.
- [3] Uschold, M. and M. Gruninger. "Ontologies: Principles, methods and applications." The knowledge engineering review 11(02): 93-136, 1996.
- [4] Smith, B. "The Blackwell Guide to the Philosophy of Information and Computing." Oxford, 2003.
- [5] Studer, R. B., V Richard Fensel, Dieter. "Knowledge engineering: principles and methods." Data & knowledge engineering 25(1): 161-197, 1998.
- [6] Chandrasekaran, B., J. R. Josephson and V. R. Benjamins. "What are ontologies, and why do we need them?" IEEE Intelligent systems 14(1): 20-26, 1999.
- [7] Isotani, S., I. Ibert Bittencourt, E. Francine Barbosa, D. Dermeval and R. Oscar Araujo Paiva. "Ontology Driven Software Engineering: A Review of Challenges and Opportunities." Latin America Transactions, IEEE (Revista IEEE America Latina) 13(3): 863-869, 2015.
- [8] ISO/IEC."ISO/IEC 24744:2007/Amd 1:2010 notation. Software Engineering – Metamodel for Development Methodologies." ISO, Geneva, 2010.
- [9] ISO/IEC. "ISO/IEC 24744. Software Engineering – Metamodel for Development Methodologies." ISO, Geneva, 2007.
- [10] Henderson-Sellers, B. and C. Gonzalez-Perez. "A comparison of four process metamodels and the creation of a new generic standard." Information and software technology 47(1): 49-65, 2005.
- [11] OMG. "SPEM 2.0." <http://www.omg.org/spec/SPEM/2.0/>, 2005.
- [12] Atkinson, C. et T. Kühne. The essence of multilevel metamodeling. UML 2001- The Unified Modeling Language. Modeling Languages, Concepts, and Tools, Springer: 19-33, 2001.
- [13] Atkinson, D. C., D. C. Weeks and J. Noll. The design of evolutionary process modeling languages. Software Engineering Conference, 2004. 11th Asia-Pacific, IEEE, 2004.
- [14] OMG."MDA Guide V1.0.1, document." <http://www.omg.org/mda/specs.htm>, 2003.
- [15] Becker, P., F. Papa and L. Olsina. "Process Ontology Specification for Enhancing the Process Compliance of a Measurement and Evaluation Strategy." CLEI Electronic Journal 18(1): 3-3, 2015.
- [16] Gazel, S., E. AKCAPINAR SEZER and A. Tarhan. "An Ontology Based Infrastructure to Support CMMI-Based Software Process Assessment." Gazi University Journal of Science 25(1): 155-164, 2012.
- [17] SEI."CMMI for Development, V1.2." <http://www.sei.cmu.edu/library/assets/cmmi-dev-v12-fr.pdf>, 2006.
- [18] Rungratri, S. and S. Usanavasin. Project assets ontology (PAO) to support gap analysis for organization process improvement based on CMMI v. 1.2. Making Globally Distributed Software Development a Success Story, Springer: 76-87, 2008.
- [19] Soydan, G. H. and M. Kokar. An OWL ontology for representing the CMMI-SW model. Workshop on Semantic Web Enabled Software Engineering (SWESE), Citeseer, 2006.
- [20] Lee, C.-S., M.-H. Wang, J.-J. Chen and C.-Y. Hsu. Ontology-based intelligent decision support agent for CMMI project monitoring and control. Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American, IEEE, 2006.
- [21] Liao, L., Y. Qu and H. Leung. A software process ontology and its application. ISWC2005 Workshop on Semantic Web Enabled Software Engineering, 2005.
- [22] Líška, M. and P. Návrát. An ontology driven approach to software project enactment with a supplier. Advances in Databases and Information Systems, Springer, 2010.
- [23] Falbo, R. A. a. B., G. " A Software Process Ontology as a Common Vocabulary about Software Processes." International Journal of Business Process Integration and Management 4: 239-250, 2010.
- [24] Falbo, R., M. P. Barcellos, J. C. Nardi and G. Guizzardi. "Organizing ontology design patterns as ontology pattern languages". The Semantic Web: Semantics and Big Data, Springer: 61-75, 2013.
- [25] Wongthongtham, P., E. Chang, T. Dillon and I. Sommerville. "Development of a software engineering ontology for multisite software development." Knowledge and Data Engineering, IEEE Transactions on 21(8): 1205-1217, 2009.
- [26] Guizzardi, G., R. de Almeida Falbo and R. S. Guizzardi. "Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. CibSE", 2008.
- [27] Mendes, O. and A. Abran. "Issues in the Development of an Ontology for a Emerging Engineering Discipline. SEKE", 2005.
- [28] P. Bourque and R.E. Fairley, e. "Guide to the Software Engineering Body of Knowledge, Version 3.0." IEEE Computer Society, 2014; [www.swebok.org](http://www.swebok.org).
- [29] Sicilia, M., J.-J. Cuadrado, E. Garc á, D. Rodríguez and J. R. Hilera . "The evaluation of ontological representation

of the SWEBOK as a revision tool". 29th Annual International Computer Software and Application Conference (COMPSAC), Edinburgh, UK, 2005.

- [30] Hilera, J., S. Sánchez-Alonso, E. García and C. Del Molino. "OntoGLOSE: A Light-weight Software Engineering Ontology. 1st Workshop on Ontology, Conceptualizations and Epistemology for Software and Systems Engineering (ONTOSE)", Alcalá de Henares, Spain, 2005.
- [31] Henderson-Sellers, B., T. McBride, G. Low and C. Gonzalez-Perez. "Ontologies for international standards for software engineering". *Conceptual Modeling*, Springer: 479-486, 2013.
- [32] Henderson-Sellers, B., C. Gonzalez-Perez, T. McBride and G. Low. "An ontology for ISO software engineering standards: 1) Creating the infrastructure." *Computer Standards & Interfaces* 36(3): 563-576, 2014.
- [33] Ruy, F. B., R. A. Falbo, M. P. Barcellos and G. Guizzardi. "An Ontological Analysis of the ISO/IEC 24744 Metamodel". *Proc. 8th International Conference on Formal Ontology in Information Systems (FOIS'14)*, 2014.
- [34] Guizzardi, G. "Ontological foundations for structural conceptual models", CTIT, Centre for Telematics and Information Technology, 2005.
- [35] Souleymane Koussoube, Armel Ayimdj, Laure Pauline Fotso, "An Ontology-Based Approach for Multi-Agent Systems Engineering", *IJMECS*, vol.5, no.1, pp.42-55, 2013. DOI: 10.5815/ijmeecs.2013.01.06.
- [36] Odell, J. J. "Power types." *Journal of Object-Oriented Programming* 7(2): 8-&, 1994.
- [37] Henderson-Sellers, B. and C. Gonzalez-Perez. "On the ease of extending a powertype-based methodology metamodel." *Meta-Modelling and Ontologies. WoMM 2006*: 11-25, 2006.
- [38] OMG. "UML2.4.1." <http://www.omg.org/spec/UML/2.4.1/>, 2011.
- [39] Djurić, D., D. Gašević and V. Devedžić (2005). "Ontology modeling and MDA." *Journal of Object technology* 4(1): 109-128, 2005.
- [40] OMG. "ODM1.1." <http://www.omg.org/spec/ODM/1.1/>, 2014.
- [41] OMG. "MOF2.4.1." <http://www.omg.org/spec/MOF/2.4.1/>, 2013.
- [42] Eclipse. 2015." <http://eclipse.org/>, 2015.
- [43] ATL. <https://eclipse.org/atl/>.
- [44] OMG. "QVT1.1." <http://www.omg.org/spec/QVT/1.1/>, 2011.
- [45] Eclipse. "EMF." <http://eclipse.org/modeling/emf/>.
- [46] OMG. "OCL2.4." <http://www.omg.org/spec/OCL/2.4/>, 2014.



**Sidi Mohamed Benslimane:** is a Full professor at Ecole Supérieure en informatique de Sidi Bel Abbes. He received his PhD degree in computer science from Sidi Bel Abbes University in 2007. He also received a M.S. and a technical engineer degree in computer science in 2001 and 1994 respectively from the Computer Science Department of Sidi Bel Abbes University, Algeria. He is currently Head of Research Team 'Service Oriented Computing' at the Evolutionary Engineering and Distributed Information Systems Laboratory, EEDIS. His research interests include, semantic web, service oriented computing, ontology engineering, information and knowledge management, distributed and heterogeneous information systems and context-aware computing.

## Authors' Profiles



**Mohamed Mehdi Hamri:** is a PhD student at Djillali Liabes University. He received his M.S. degree in computer science from Sidi Bel Abbes University in 2009. Since 2009, he is an Assistant Professor at the Computer Science Department of Sidi bel abbes University. Currently, he is member of the EEDIS Laboratory. His research interests include semantic web, ontology engineering, information and knowledge management, and software process modeling.