A Framework to Combine MDA and Ontology Learning

Regina C. Cantele[1], Maria Alice G. V. Ferreira[1], Diana F. Adamatti[2] and Moyses Araujo[3]

[1] InterLab - Laboratório de Tecnologias Interativas
Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil

Faculdade de Tecnologia - FTEC, Caxias do Sul, Brazil

[3] IBTA, Brazil

Abstract. This paper proposes to join two different approaches: Software Engineering and Semantic Web. The first one cames from Model Driven Architecture (MDA) and the second one, from the Ontology Engineering area, more specifically Ontology Learning. The main idea of this work is to accelerate the initial construction of ontologies from knowledge already represented in several different models such as CASE repository or database dictionaries and text, based on standards of MDA for interoperability. In this way, a framework was developed to join these concepts, where the steps sequence to apply it were defined until an initial representations of ontology was obtained.

1 Introduction

Nowadays, the Web is an important knowledge source and people use it to find relevant information for their needs. However, Web information is unstructured and placed in such a way that it is inconvenient for searching. Web pages are simple collections of characters, and the agents - the knowledge collectors - may be unaware that the information they need is available. In the mid 1990s, Berners-Lee and colleagues defined the Semantic Web as an alternative approach to Web, where information can be structured and used by the agents. Semantic Web use metadata and ontologies for this task [4].

In Computer Science, ontologies started to be used by the Artificial Intelligence (AI) area, mainly in Natural Language Processing (NLP) and Machine Learning (ML). However, many works in these areas fail to provide interoperability between works.

In Software Engineering, Model-Driven Engineering (MDE) is an approach to the development of software that separates the specification of the system functionality from its implementation on a particular platform. In this way, it can share models and concepts between different tools and methods [29]. One of the major initiatives through which the MDE is accomplishing this goal is by the MDA, defined by Object Management Group (OMG), as a comprehensive interoperability framework for defining future interconnected systems. Another important specification is Architecture-Driven Modernization (ADM) - the process of understanding and evolving existing software

C. Cantele R., Alice G. V. Ferreira M., F. Adamatti D. and Araujo M. (2008).
A Framework to Combine MDA and Ontology Learning.
In *Joint Proceedings of the 5th International Workshop on Ubiquitous Computing (IWUC 2008) 4th International Workshop on Model-Driven Enterprise Information Systems (MDEIS 2008) 3rd International Workshop on Technologies for Context-Aware Business Process Management (TCoB 2008)*, pages 43-51
DOI: 10.5220/0001730400430051
Copyright © SciTePress

artifacts and which details the use of the tools of progressive engineering with the tools of reverse engineering. Among the this specifications, it is important to point out: Meta Object Facility (MOF); XML Metadata Interchange (XMI); and ODM [1].

MOF is an extensible model-driven integration framework for defining, manipulating and integrating metadata and data in a platform independent manner. MOF-based standards are used for integrating tools, applications and data [26]. In MOF terminology, metadata that describes metadata is called meta-metadata, and a model that consists of meta-metadata is called a metamodel. A MOF metamodel defines the abstract syntax of the metadata in the MOF representation of a model. The MOF integrates these metamodels by defining a common abstract syntax for defining metamodels. This abstract syntax is allied to the MOF Model and is a model for metamodels; i.e. a meta-metamodel. The classical framework for metamodeling is based on an architecture with four meta-layers: M3 - meta-metamodel, M2 - metamodel, M1 - model e M0 - instances of model.

XMI is a model-driven XML Integration framework for defining, interchanging, manipulating and integrating XML data and objects. XMI-based standards are used for integrating tools, repositories, applications and data warehouses. XMI provides rules which can generate a schema for any valid XMI-transmissible MOF-based metamodel. XMI provides a mapping from MOF to XML [26].

The specification Ontology Definition Metamodel (ODM) joins the Semantic Web with MDA. ODM is formal grounding for representation of business semantics and defines independent metamodels, related profiles, and mappings among the metamodels corresponding to several international standards for ontology and Topic Maps definition, as well as capabilities supporting conventional modeling paradigms for capturing conceptual knowledge, such as Unified Modeling Language (UML) and Entity-Relationship (E/R) modeling.

However, ontologies need to be constructed and populated, which is a hard task. Automatic machines acquisition is an open problem. Therefore, semi-automatic processes with human intervention are widely used. A lot of tools as well as standards allowing systems interoperability have been developed throughout the last years for the accomplishment of such task. In this approach, Ontology Learning appears as a set of algorithms and tools to automatically derive knowledge from domain-specific text collections or unstructured textual resources [22] [17].

The goal of this paper is *to develop a framework* to combine the Software Engineering concepts (based on MDA approach) and Ontology Learning to build new ontologies for Semantic Web, using knowledge already represented (e.g., legacy systems). This framework is composed by blocks, where requirements and steps are defined.

The paper is structured in 5 sections. In sections 2 and 3, a brief overview of Ontology Definition Metamodel (ODM) and Ontology Learning are presented, respectively. Section 4 presents the developed framework to combine Ontology Learning and MDA. Finally, the conclusions are presented in Section 5.

2 Ontology Definition Metamodel

According to [16], there are five components in ontology: concepts, relations, function, axioms, and instances. Concept can be anything said about something, and therefore, it could also be the description of a task, function, action, strategy, reasoning process, properties, etc. Relations represent the type of interaction between domain concepts. Functions are special case of relations in which the n-th element of the relationship is unique for the n-1 preceding elements. Axioms are used for modeling sentences are taken for granted as true. Instances are used for representing elements.

ODM is a standard to support ontology development and conceptual modeling in several standard representation languages. It provides coherent framework for ontology creation based on MOF and UML. Knowledge engineering requirements may include some ontology development for traditional domain, process, or service ontologies, but they may also include: generation of standard ontology descriptions (e.g., OWL) from UML models; generation of UML models from standard ontology descriptions (e.g., OWL); integration of standard ontology descriptions (e.g., Ontology Web Language (OWL)) with UML models [26]. Moreover, the metamodel increases with constraints. These constraints are expressed in the Object Constraint Language (OCL), a declarative language which provides constraint and object query expressions on object models that cannot be otherwise expressed by diagrammatic notation.

All MDA specifications contribute to the interoperability of the constructed diagrams already with the necessary new representations. Some researchers joined the existing ontology representation formalism to the MDA of OMG [2, 15, 32, 20].

3 Ontology Learning

Ontology construction is a complex task and needs some criteria and methodologies to be developed. Jimenez-Ruiz and Berlanga (2006)[19] studied several methodologies and classified them into two groups: centralized or collaborative approach in the development of ontologies. The first group has the examples Methontology [12], Unified Process for Ontology Building (UPON) [10] and CommonKADS [28]. The second group has the examples KA2 [3], Human-Centered Ontology Engineering Methodology (HCOME)[21] and Diligent [7]. The choice of approach is in accordance with the type of ontology to be developed. For example, for an ontology of application, the centralized approach is recommended and for an ontology of domain, the collaborative development is.

Ontology development can be based on the layer cake built by Cimiano [8], as shown in Figure 1, where the acquisition of knowledge is executed in subtasks. It is primarily concerned with axiomatizing the definition of concepts as well as the relationships between them. After that, it is important to connect concepts and relations to the symbols referring to them. This implies the acquisition of linguistic knowledge of terms used for referring to a specific concept and potential synonyms of these terms. Moreover, ontology may consist of some concept hierarchy or any other non-hierarchical relation. In order to constrain interpretations of concepts and relations, axiom schemata, such as disjointness between concepts as well as symmetry, reflexivity, transitivity, etc., can be instantiated for relations. Finally, one is also interested in using an ontology to derive facts that are not explicitly modeled in the knowledge base but can be derived from it. For this purpose, logical axioms modeling implications between concepts and relations can be defined.

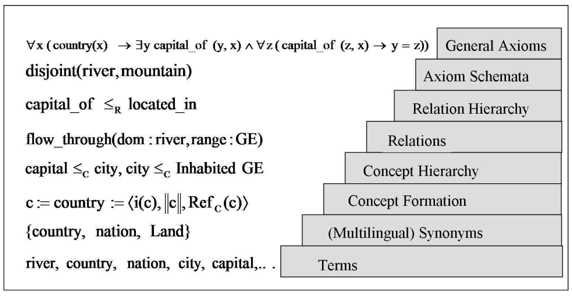
$\forall x\,(\text{country}(x) \rightarrow \exists y\ \text{capital\_of}\ (y,x) \wedge \forall z\,(\text{capital\_of}\ (z,x) \rightarrow y = z))$
General Axioms
disjoint(river, mountain)
Axiom Schemata
$\text{capital\_of} \leq_R \text{located\_in}$
Relation Hierarchy
flow_through(dom : river, range : GE)
Relations
$\text{capital} \leq_C \text{city},\ \text{city} \leq_C \text{Inhabited GE}$
Concept Hierarchy
$c := \text{country} := \langle i(c), \|c\|, \text{Ref}_C(c)\rangle$
Concept Formation
{country, nation, Land}
(Multilingual) Synonyms
river, country, nation, city, capital,.. .
Terms

Fig. 1. Ontology Learning Layer Cake[8].

Another way of ontology construction is Ontology Learning. This work considers the two approaches in Ontology Learning: from texts and from schematas.

The first approach shows to be a set of algorithms and tools for automatically deriving knowledge from domain-specific text collections or unstructured textual resources [8]. Pattern-based techniques are used to identify words that are useful inputs for learning methods because they are likely to represent concepts or relations [20]. Special syntactic patterns have been developed for populating the instances identified in the ontological relations.

The second approach, Ontology Learning from schematas contemplates the data in the form of schemata with a formal semantics and are typically represented in models of data such as E/R, UML and XML. Generally, the techniques of Reverse Engineering with mappings for representation of ontology languages are used.

A comparison between environments of Ontology Learning can be seen in Shamsfard and Barfoush [30].

Some tools based on these techniques have been developed, such as, ASIUM [11] of the University of Paris; OntoLT [5] of the German Research Center for Artificial Intelligence; Text2Onto and AEON [9] of the University of Karlsruhe; OntoLearn [24] of the University La Sapienza of Rome; and Ontogen [14] of the Joef Stefan Institute. One essential requirement for all methods and tools is that a representative input data for the domain must be built in the ontology.

4 The Proposed Framework

The main goal of the proposed framework is to bring together the concepts of Software Engineering with MDA and Ontology Engineering to accelerate the construction of ontologies from knowledge already represented. This process comprises four blocks and associated resources, as shown in Figure 2: Selection of Resources, Ontology Learning, Environment Ontologies and Ontology Engineering.

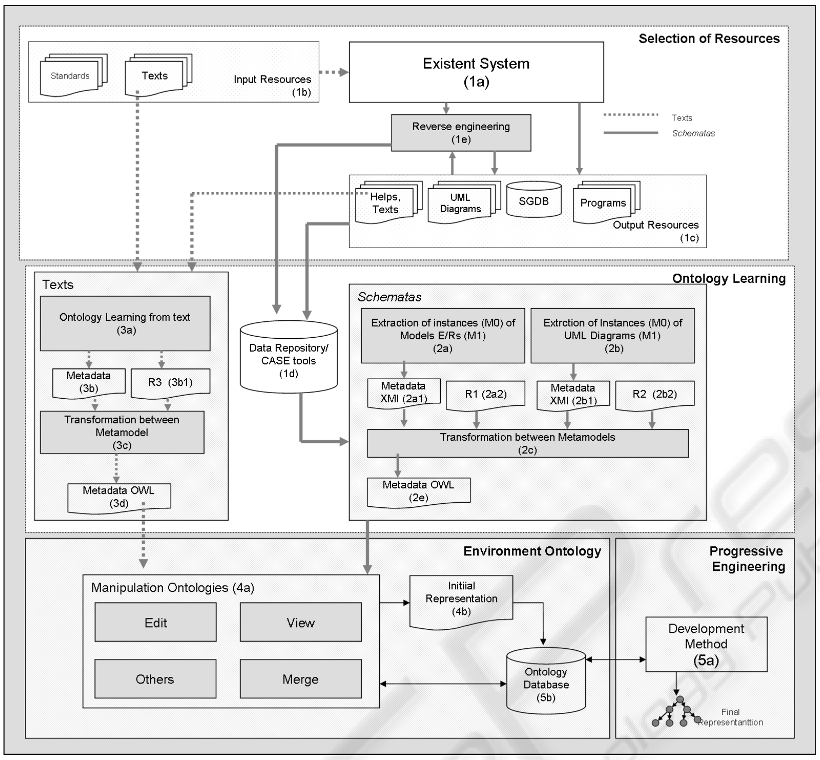

Fig. 2. Framework to combine Ontology Learning and MDA.

A. Selection of resources. It is the part responsible for representing old resources of an information system. The knowledge of the existing system is present in the input resources (Standards and Texts - (1b)) and in the output resources (database dictionary RDBMS, UML Diagrams, Programs, Helps, Texts etc. - (1c)) often represented in Data Repository - metamodels of the CASE tools used in its conception and implementation (1d). When the existing system (1a) did not use a tool for its construction (it is very common), the Data Repository (1d) does not exist, and then, Reverse Engineering (1e) is necessary to populate this repository (1d); for example, the database dictionary can be used to get the Class Diagram in UML [13]. This is represented by the dotted lines in Figure 2.

B. Ontology Learning. In this block, the process of models extraction is carried out by two tasks: extraction of models with standards MOF/XMI (2a) and Transformation between metamodels (2c). The process of extraction of models with standards MOF/XMI can be achieved by means of tools implemented with standard MOF (2a) so that the final result is presented in standard XMI in Metadata XMI (2a1) and (2b1). The (2c) realizes a core transformation by having an input in E/R or UML models represented in (2a1) and (2b1) and, producing an ontology according to the OWL metamodel, resulting in an XML document at OWL/XML syntax: Metadata OWL (2e). The transformation part is dedicated to the mapping out of the UML model for ontology, i.e. UML classes are mapped into OWL classes, attributes into datatype property, associations into object property, instances into OWL individuals, etc. This process needs to apply rules defined in the ODM to OWL model (2a2) and (2b2). It is the transformation of a representation form into another, with the same level of abstraction, preserving the system external behavior (functionality semantics). This process offers the possibility to manage E/R and UML instances and populate the ontology with corresponding knowledge. This process pipeline is represented in gray lines in Figure 2.

The other process is Ontology Learning from text (3a). This process is responsible for processing text resources (e. g. HTML pages and papers from a given domain) and creating a very simple ontology from the extracted data. It considers the use of some tools - based on techniques of natural language and machine learning - to get concepts, its associations, and hierarchy in metadata (3b). In general, these tools have implemented many algorithms for this, with incremental executions until significant concepts were reached. After (3a) locks up, this process of extraction in standard OWL is realized by obtaining the metadata OWL (3d). The extracted information (3b) is stored in internal metamodel depending on the applied tools and can be directly produced by applying specific algorithms. The requirement of tools must be in compliance with metamodel OWL for output documents or provide their proprietary metamodel for the transformation processes.

C. Environment Ontology. After the accomplishment of process 2 or 3, it is necessary to use the Manipulation Ontology (4a) to visualize the initial ontology representations - (2e) or (3d). Thus, the first conceptual representation of the ontology could be edited (4b), in a clearly understandable way for ontology engineers, reusing the first model of the ontology in format E/R and UML (1c) or in text format (1b) and represented on a Ontology Database (5b).

D. Progressive Engineering. (4b) will be used for executing tasks of Development Method (5a) in order to obtain the final ontology representation. The use of methodologies assures the Ontology Base (5b) to be consistent and coherent with the represented domain and with the current system model. The choice of methodology is linked to the use of ontology; therefore, an ontology of application development will be centralized as well as a domain ontology of the collaborative development.

One important aspect in this framework is the use of the OMG metamodels standards to Ontology Learning and ontologies environment according to Figure 3. At this point, the proposed framework introduces the use of the MOF and their metamodels already defined to represent the OWL model with ODM. This ensures that all models (M1) considered by the different tools have their metamodel shared between the tools through applied rules processing on metamodels with XMI. Also, all instances (M0) of the models are shared. To really enjoy the results of the various tools for Ontology Learning from texts, it is necessary to tailor the metamodel to the OWL metamodel. This can be done with the creation of tools for conversion between the metamodel existing non-standardized model for the OWL, or a structural change in the architecture of the tool to implement the standardized metamodel. Text2Onto [18] and Protégé [25] are examples of tools with metamodels own disclosed.

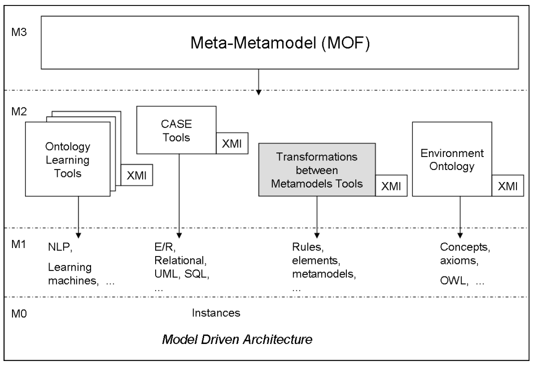

Fig. 3. Four-Layer Architecture of MOF.

The practical application of this framework was started in a previous work [6], where the authors tested the schematas approach to generate new ontologies (phases 1, 2 and 4 in the Figure 2). The tests were conducted with a large number of tools, such as Protégé [27], an ontology environment and Magic Draw [23], a CASE tool. Regardless of the semantics of each of the models being different (E/R, UML and, Ontology), the represented knowledge is not lost by the transformations. The use of representative mnemonics from the existent systems helps to generate knowledge-represented. This process has shown to be effective when there is a large number of entities or classes from the legacy system. Another test was conducted with UML diagrams provided by the IMS Learning Design [31], and 192 classes and 390 properties were obtained from these UML diagrams.

5 Conclusions and Further Works

This paper presented the conceptual viability of different standards integration and techniques associated to the metadata to help Ontology Engineering. Many standards of OMG (MDA, ADM, UML, MOF, XMI, ODM), as well as many standards of the W3C (XML, and OWL) were recognized, accepted and used by the Information Technology community.

The success of the Semantic Web depends on the deployment of ontologies. Using ontology studies from NLP and ML can make the Engineering Ontologies process easier and faster, because they just share a metamodel of the common representation from an ontology as proposed by ODM. In the case of Ontology Learning from schematas, the process is very simple and it presents good results. Therefore, in the case of Ontology Learning from texts, it adopts standards of metamodel and it requires efforts from the researchers' community. Ontology Engineering can use the initial representations so that it can be favored by the use of some already represented knowledge. It must be taken into account that the semantic terms may vary from one context to another, from a place to another, and from a person to another. Considering these semantic heterogeneities, Ontology Engineering is not a trivial activity and requires time, availability and consensus by specialists. Therefore, the idea of exploring things that have already been represented in the past has been taken into consideration in [11, 5, 15, 2] and in this article and this idea can help in the construction of an initial model of ontology for a specific domain.

The proposed framework starts to be the ontology engineers' convergence point, allowing the sharing of represented knowledge and the description of the most common concepts obtained from the representantions initial ontology. As further works, some tests will be conducted with Ontology Learning from texts tools (phases 1 and 3 in the Figure 2), transformation tools and repository for XMI files with the application of Knowledge Discovery Meta-Model (KDM) and Abstract Syntax Tree Meta-Model (ASTM) specifications. The chosen tools must provide metamodels or generate meta-models to ADM, ODM and OWL standards and control of the files intermediaries.

References

1. Omg: Object management group - architecture driven modernization. http://adm.omg.org, 2008.
2. K. Baclawski, M. Kokar, P. Kogut, L. Hart, J. Smith, W. Holmes, J. Letkowski, and M. Aroston. UOL: Unified ontology language. Assorted papers discussed at the DC Ontology SIG Meeting, 2002. http://www.omg.org/cgi-bin/doc?ontology/2002-11-02.
3. V. R. Benjamins, D. Fensel, S. Decker, and A. Goméz-Pérez. (ka)2: building ontologies for the internet: a mid-term report. *International Journal of Human-Computer Studies*, 51(3):687–712, 1999.
4. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):28–37, 2001.
5. P. Buitelaar, D. Olejnik, and M. Sintek. A protégé plug-in for ontology extraction from text based on linguistic analysis. In *Proceedings of the 1st European Semantic Web Symposium (ESWS)*, Heraklion, Greece, May 2004.
6. R. C. Cantele, D. F. Adamatti, M. Ferreira, and J. S. Sichman. Aplicação da reengenharia de software na construcão acelerada de ontologias. In L. F. Friedrich, editor, *Simpósio Brasileiro de Sistemas de Informacão (SBSI 2005)*, Florianópolis, Brasil, Outubro 2005. SBC.
7. P. Casanovas, N. Casellas, C. Tempich, D. Vrandecic, and V. Benjamins. OPJK and diligent: ontology modeling in a distributed environment. *Artificial Intelligence and Law*, 15(2):171–186, 2007.
8. P. Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer, Germany, 2006.
9. P. Cimiano and J. Volker. Text2onto - a framework for ontology learning and data-driven change discovery. In *roceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB'2005)*, Alicanet, Spain, June 2005.
10. A. De Nicola, M. Missikoff, and R. Navigli. A proposal for a unified process for pntology building: Upon. In *Proceedings of the 16 International Conference Database and Expert Systems Applications*, pages 655–664, Dinamarca, 2005.
11. D. Faure and T. Poibeau. First experiments of using semantic knowledge learned by asium for information extraction task using intex. In *Proceedings of Ontology Learning ECAI-2000 Workshop*, pages 7–12, 2000.
12. M. Fernández, Gómez-Pérez, and N. Jurino. Methontology: From ontological art towards ontological engineering. In *Spring Symposium Series*, pages 33–40, 1997.
13. M. Ferreira, M. Araujo, and R. C. Cantele. Educacão baseada na Web e Web Semântica: construindo uma nova forma de educacão a distância. In *Proceedings of the WCCSETE'2006*

- *World Congress on Computer Science, Engineering and technology Education*, pages 692–696, Itanham/Santos, 2006.

14. B. Fortuna, M. Grobelnik, and D. Mladenic. Background knowledge for ontology construction. In *Proceedings of the 15th International conference on World Wide Web 2006*, pages 949–950, Edinburgh, Scotland, 2006.
15. D. Gasevic, D. Djuric, and D. Devedzic. *Model Driven Architecture and Ontology Development*. Springer, Berlin Heidelberg, Germany, 2006.
16. A. Gómez-Pérez and R. Benjamins. Overview of knowledge sharing and reuse components: Ontologies and problem solving methods. In V. Benjamins, editor, *IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lesson learned and Future Trends*, volume 18, pages 1–15, Estocolmo, 1999. CEUR Publications.
17. A. Gómez-Pérez and D. Manzano-Macho. A survey of ontology learning methods and techniques. http://ontoweb.aifb.uni-karlsruhe.de/Members/ruben/Deliverable, 2003.
18. P. Haase and J. Volker. Ontology learning and reasoning - dealing with uncertainty and inconsistency. In *Proceedings of the 4 International Semantic Web Conference*, Ireland, 2005.
19. E. Jimenez-Ruiz and R. Berlanga. A view-based methodology for collaborative ontology engineering: an approach for complex applications (vimethcoe). In *Proceedings of the 15 IEEE International Workshops on Enabling Technologies: Infrastructure for collaborative Enterprises*, pages 376–381, Myanmar, 2006.
20. F. Jouault and I. Kurtev. On the architectural alignment of ATL and QVT. In *In Proceedings of ACM Symposium on Applied Computing (SAC 06)*, pages 1188–1195, Dijon - France, 2006. ACM Press. Chapter Model Transformation.
21. K. Kotis and G. Vouros. Human-centered ontology engineering: the HCOME methodology. *International Journal of Knowledge and Information Systems*, 10(1):109–131, 2006.
22. A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.
23. Magicdraw. Magicdraw architecture made simple. http://www.magicdraw.com/, 2007.
24. R. Navigli and P. Velardi. Ontology enrichment through automatic semantic annotation of on-line glossaries. In *Proceedings of the 15th International Conference on Knowledge Engineering (EKAW 2006)*, pages 126–140, Podebrady, Czech Republic, 2006. LNAI 4248.
25. N. F. Noy, R. W. Fergerson, and M. A. Musen. The knowledge model of protégé-2000: combining interoperability and flexibility. In *Proceedings of the 2 International Conference on Knowledge Engineering and Knowledge Management*, pages 69–82, France, 2000.
26. OMG. Object Management Group - ontology definition metamodel specification. http://www.omg.org/, 2008.
27. Protégé. The protégé ontology editor and knowledge acquisition system. http://protege.stanford.edu/, 2007.
28. G. Schereiber, M. Crubezy, and M. Musen. A case study in using protégé-2000 as a case tool for commonkads. In *Proceedings of the 12 International Conference Knowledge Engineering and Knowledge Management*, volume 1937, pages 33–48, Berlin, 2000.
29. D. Schimt. Model driven-engineering. In *IEEE Computer Society*, pages 25–31, 2006.
30. M. Shamsfard and A. Barforoush. Learning ontologies from natural language texts. *International Journal of Human-Computer Studies*, 60(1):17–63, 2004.
31. Van der Vegt, Wim. The poseidon file containing the uml profile of ims-ld level c. http://dspace.ou.nl/handle/1820/659, 2006.
32. T. Win, H. M. Aung, and N. L. Thein. An mda based approach for facilitating representation of semantic web service technology. In *Proceedings of 6th Asia-Pacific Symposium on Information and Telecommunication Technologies*, pages 260–265, Myanmar, 2005.