# ONTOLOGY AND ITS ROLE IN DOMAIN-SPECIFIC MODELING

**ZHU Ning[(a)], LEI Yonglin[(b)], LI Xiaobo [(c)], ZHU Yifan[(d)]**

College of Information System and Management, National University of Defense Technology, Changsha, Hunan, P.R. China, 410073

[(a)]zhun87025@126.com, [(b)]samuelyonglin@gmail.com, [(c)]lixiaobo@nudt.edu.cn, [(d)] nudtzyf@163.com

**ABSTRACT**
Domain-Specific Modeling (DSM) aims to support domain experts building models with familiar concepts by improving the abstract level of modeling languages. And it is a trend in the Model Driven Engineering (MDE) domain. However, there are still a series of problems left to be solved well for DSM, such as how to design a domain-specific language adhering to the domain nicely and how to verify the domain model during building models. An ontology is a conceptual model of a domain with strict and explicit semantics consisting of domain concepts and their relationships. It seems to be a promising way to utilizing ontology to boost DSM up. In this paper, what's an ontology and its metamodel is discussed firstly. Then the DSM method is introduced and its problems are proposed. Finally the different ways of applying ontologies in DSM to solve its problems are summarized.

Keywords: domain-specific modeling, domain concept, ontology

## 1. INTRODUCTION

Domain-Specific Modeling (DSM) supports domain experts to build models rapidly with familiar domain concepts by improving the abstract level of modeling languages. And the models are more comprehensible and easier to communicate with others. What's more, DSM can enhance software production with tools to translate the domain models to target productions (such as code, executable programs) (Tolvanen and Rossi 2003).

DSM is a trend in Model Driven Engineering domain. It aggregates advantages of Model-Integrated Computing (MIC) and Model Driven Architecture (MDA), two important branches in MDE. MIC was proposed by Janos Sztipanovits and Garbor Karsai from Vanderbilt University in 1997 (Sztipanovits and Karsai 1997). It utilizes the metamodeling environment to create metamodels of a specific domain and configures a domain-specific modeling environment for the metamodel to support domain modelers to create domain models. Object Management Group put forward MDA in 2001 to solve the integration problem of software architecture and various software tools (Kleppe, Warmer, and Bast 2006). MDA decomposes the whole development process into three models with different abstract levels, respectively Platform Independent Model (PIM), Platform Specific Model (PSM) and executable codes with transformation standards among different models. DSM absorbs ideas of MIC and utilizes technique and standards provided by MDA to support modeling domain-friendly.

However, there are still a series of problems left to be solved for DSM such as how to design a domain-specific language adhering to the domain nicely, how to reduce the learning curve for domain languages learners, how to verify and debug a domain model and so on. An ontology is a conceptual model of a domain with strict and explicit semantics consisting of domain concepts and their relationships. It seems to be a promising way to utilizing ontologies to solve the problems in DSM.

The paper is organized as the following. In section 2 what's an ontology and its metamodel is discussed and the pervasive used ontology language Web Ontology Language (OWL) and the corresponding tool Protégé OWL is introduced simply. Then the method of DSM is concluded and its problems are proposed in section 3. In section 4, the different ways of applying ontologies in DSM to solve its problems are summarized. Finally, we conclude the paper and talk about how we will use ontologies and DSM in our next research.

## 2. ONTOLOGY

### 2.1. What's an ontology

Ontology is initially a philosophical concept. It was proposed in the $17^{th}$ century and accepted widely in the philosophy circle in the $18^{th}$ century. And ontology became a branch of metaphysics in philosophy domain and considered the nature of existence and relationship. According to Aristotle and Peirce, it is the science of being qua being and aims to find the most general features of reality and real objects. It does not focus on specific disciplines such as physics or chemistry, but the transcendent rules over all the entities in the different domains. (Guizzardi 2005)

In 1967, ontology was introduced in Computer Science, and then it is pervasively used in the following several decades in domains of Database & Information System, Software Engineering, especially in Domain Engineering and Artificial Intelligence. When ontology in philosophy is immigrated into Computer Science, its concept is localized according to the application domains and experience by different researchers:

- An ontology defines the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary. (Neches, Fikes, and Finin 1991)
- An ontology is an explicit specification of a conceptualization. (Gruber 1993)
- An ontology is an explicit, partial account of a conceptualization/the intended model of a logical language. (Giaretta and Guarino 1995)
- An ontology is a formal specification of a shared conceptualization (Borst 1997).
- An ontology is a set of structured terms that describing some domain or topic and provides a skeletal structure for a knowledge base. (Swartout, Patil, and Knight 1997)
- An ontology is the vocabulary related to a generic domain (like medicine, or automobiles) or a generic task or activity (like diagnosing or selling). (Guarino 1998)

Though many definition of ontology exist, there's no consensus on what's an ontology. However, examining these concepts, there are some common foundations of an ontology according to Fensel (Fensel, Harmelen, and Horrocks 2001):

- Conceptualization. An ontology is a conceptual model for reality and real object.
- Explicit. Concepts and their relationships are defined using precise semantic.
- Formal. An ontology should be machine understandable.
- Shared. Knowledge represented by an ontology is recognized commonly by the users in the same domain.

So considering about all the definitions of ontology and the research problem we are facing (Applying DSM in military modeling and simulation), we see an ontology as a conceptual model of reality which including concepts, their relationships and common entities of these concepts in some domain. And the conceptual model is described in exact and formal semantics.

## 2.2. Ontology metamodel

Colin Atkinson and Thomas Kuhne proposed an ontology metamodel using the biological taxonomy as an example (Atkinson and Kuhne 2003). They separated the biological ontology language into four levels, respectively O3, O2, O1 and O0. In the O2 level are the metaconcepts such as Kingdom, Phylum, Class and so on. In the O1 level are the concepts (Animal, Chordate, Mammal, et al.) instantiated from the metaconcepts while the instances are in the O0 level. However, what's in O3 level was not explained clearly and how these four levels correspond to the four-level metamodel hierarchy was not discussed, too.

OMG provided Ontology Definition Metamodel in the four-level metamodel hierarchy (OMG, 2014). In the M3 level is the MOF, and the structure of the ontology language is defined in the M2 level. The ontology is an instance of the M2 level and dwells in the M1 level. The M2 level consists of the instances of the ontology. But The O2 level which composes of the metaconcepts in Atkinson and Kuhne's ontology language is not considered about.

Referring to Ontology Definition Metamodel provided by OMG and ontology metamode proposed by Colin Atkinson and Thomas Kuhne, the ontology metamodel hierarchy combining them is shown in Figure 1.

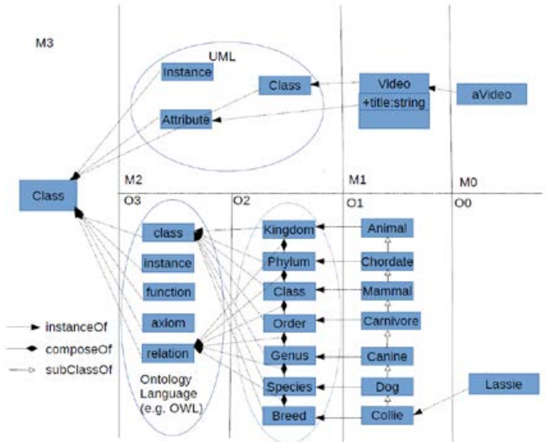


Figure 1: The ontology metamodel hierarchy

In Figure 1, the M2 level of the four-level metamodel hierarchy is divided in two levels respectively O3 and O2 levels. The O3 level is the instance of the M1 level and corresponds to the M2 level of Ontology Definition Metamodel. The O2 level is optional for the domain consists of metaconcepts. The other levels are the same as the Ontology Definition Metamodel.

## 2.3. OWL and Protégé

Currently, OWL is a pervasive ontology language and is adapted by World Wide Web Consortium (W3C) as a standard for ontology modeling. The foundation elements of OWL are Class, Individual and Property. OWL consists of three types of languages, respectively OWL-Lite, OWL-DL and OWL-Full. OWL-Lite is the least subset of OWL, which can be used to describe the simple class hierarchy and constraints. OWL-DL involves Description Logics and can support automated reasoning to verify consistence of the concepts and the class hierarchy. OWL-Full is the most expressive subset of OWL, but doesn't enable automated reasoning. The OWL ontology is stored based on RDF/XML. (Horridge 2011, Li 2013)

Protégé is an open-source ontology modeling tool developed by Stanford University in Java for ontology modeling and reasoning. It coheres to OWL and is composed of four main views. In the class view, the concepts and their hierarchical relationships can be modeled. Properties of a concept should be described in the Data Property view, while relationships of concepts in the Object Property view. The Individual view composes of all the instances of the concepts. Protégé supports to be extended by plugins like Eclipse. So quite a lot of excellent customized views are developed by researchers to describe, visualizes or analyze the ontology model. Reasoners such as Racer, FaCT++ and HermiT are integrated into Protégé as plugins to support checking and reasoning of ontology model. (Li 2013)

# 3. DOMAIN-SPECIFIC MODELING

## 3.1. The DSM Method

The process in the DSM method is shown in Figure 2. It consists of five different steps, respectively domain analysis, design of domain-specific modeling language (DSML) and domain-specific modeling environment (DSME), domain model analysis and target product generation.

Domain analysis is a branch of Domain Engineering. It analyzes the domain to collect domain information and constructs domain conceptual models which describe various entities, their properties, roles, relationships and constraints in the domain (Reihhartz-Berger and Sturm 2004). So domain analysis brings out the domain concepts, relationships and constraints and is the foundation of DSML design.

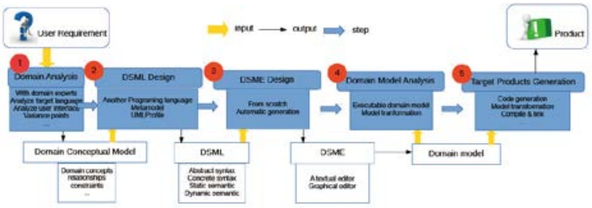


Figure 2: The process in the method of DSM and its products

DSML is the core of the DSM method. Generally, a DSML composes of abstract syntax, concrete syntax, static semantic and dynamic semantic (Cho 2011). Abstract syntax describes concepts, relationships and constraints of a language in a formal way. Concrete syntax corresponds to the representation of the models, such as a word in textual format or an icon in graphical format. Static semantic represents the static constraint relationships between concepts, while dynamic semantic shows the meaning of the components when the model executes. There is a type of DSML which is based on the common programming languages (such as Yacc, SIMULA67) and can be designed as another programming language. However, the metamodeling and UML profile are the mainstream methods to construct a visual domain-specific modeling language.

DSME is related to the concrete syntaxes and static constraints of the DSML. Concrete syntaxes are correspondent to the visual icons or graphics representing the models or the highlighted text. Static constraints correspond to the limits in using the domain-specific modeling environment such as the confinement on linking one modeling element to another. DSME can be developed from scratch according to the concrete syntaxes and static semantics. But the pervasive used method to construct DSME is to generate DSME from them automatically (Rath and Varro, Zbib, Jain and Bassu 2006). Some mainstream DSM tools such as MetaEdit+, EMP, and GME all support this method.

Domain experts build models using DSME. These models can be used to communicate with others, and also can be analyzed. Currently, there are two methods to analyze the domain model. Firstly, the corresponding executable semantics can be embedded in the DSML to make the domain model executable. Secondly, the domain models can be transformed to another modeling language or formalism to be analyzed according to the analysis ability of the target language or formalism.

The most valuable of DSM is to support domain models to generate target products, automatically transforming the design into realization.

## 3.2. The problems of DSM

DSM is a good way to support rapid model design, development and product implementation. But it is not perfect. There are still a series of problem left waiting to be solved well (Walter, Parreiras and Staab 2014). The problems of DSM include:

- How to design a DSML adhering to a domain well. Currently the emphasis of most DSM researches is on the middle part of the whole DSM method, while few of them consider on how to design the components of the DSML to match the domain well and if there is a better DSML for the same domain.
- How to capture formal constraints for DSML Design. The constraints of DSML in a domain are embedded in the domain knowledge, the current DSM method lacks of a mechanism to help DSML designer to extract the constraints in the domain.
- Lacking a tool for connecting domain analysis with DSML design. Though current pervasive DSM tools provide environments to build metamodels of a domain, there is not a tool to help or guide users to capture the language constructs.
- How to help domain experts to learn and use the DSML and DSME quickly. For domain modelers who are new to the DSML, suggestions should be provided to guide the use of the domain-specific modeling languages.
- How to verify and debug a domain model. Most of current DSM tools are devoid of the ability to check the domain models' correctness and completeness. What's more, few of them support domain modelers to debug the models to find the bugs.

# 4. ONTOLOGY'S ROLE IN DSM

Ontology is a conceptual model of reality and real objects. It consists of concepts and their relationships, constraints, some common entities in the domain, which can be used to solve the problems in the DSM proposed in the previous section.

The roles of an ontology in the DSM method are sorted according to the different steps of the DSM method, which is shown in Table 1.

In the domain analysis step, an ontology can be used as a domain conceptual model to represent the domain as the result of domain analysis. And it has been used widely in domain engineering (Tairas, Mernik, and Gray 2009).

In the DSML design phase, an ontology can guide the design of DSML. If the ontology is modeled in a formal language such as OWL, the transformation from ontology model to the metamodel in the M2 level of the four-level metamodel hierarchy can be realized to generate DSML automatically. What's more, different DSMLs for the same domain can be compared for their appropriateness based on the ontology (Guizzardi 2005).

In the DSME design step, the ontology information can be embedded to support suggestions and guidance for new modelers to build domain models. And an ontology can be utilized to verify and debug the domain model in the Domain Model Analysis (Walter, Parreiras and Staab 2014). The last step Targets Products Generation is about code generation and model transformation. In our opinion, currently ontology information is not needed there.

Table 1: Ontology's role in DSM

| Ontology's role in DSM | |
|---|---|
| Domain Analysis | As a domain conceptual model<br>As a basis for model comparison |
| DSML Design | Guide the DSML design<br>Generate DSML |
| DSME Design | As suggestions or guidance in the modeling process |
| Domain Model Analysis | Verify and debug the domain model |
| Target Products Generation | Not yet |

## 5. CONCLUSION

In this paper we discuss about what's an ontology, the ontology metamodel and the pervasive used ontology language OWL and its correspondent tool Protégé. And then we introduced the DSM method and separated it into five steps including domain analysis, DSML design, DSME design, domain model analysis and target products generation. The problems of DSM are proposed including how to design a DSML, how to extract the constraints from the domain and so on. Finally, we summarize the roles of ontology in the five different steps of the DSM method.

Currently, how to transform the ontology model to the M2 level of the four-level metamodel hierarchy of UML has not been studied so well for generating a DSML automatically. And it will enhance the DSM method significantly if the problem is solved. So in the future, we will stick to the problem and research on how to transform DoDAF Meta Model (DM2, an ontology model for military architecture framework) to a DSML for domain-specific architecture modeling.


## ACKNOWLEDGMENTS

The work is partly supported by the national Natural Science Foundation of China (No. 61273198)


## REFERENCES


Tolvanen J. and Rossi M., 2003. Metaedit++:defining and using domain-specific modeling languages and code generator. OOPSLA'03. October 26-32, Anaheim , California, USA.

Sztipanovits J. and Karsai G., 1997. Model-integrated computing. IEEE Computer, 30(4), 110-111.

Kleppe A., Warmer J. and Bast W., 2006. Translated by Bao Zhiyun. MDA Explained. Beijing: Posts & Telecom Press. (in Chinese)

Guizzardi G., 2005. Ontological Foundations for Structural Conceptual Models. Thesis (PhD). Telematica Insituut Fundamental Research Series.

Neches R., Fikes R.E., Finin T., et al., 1991. Enabling technology for knowledge sharing. AI Magazine, 12(3), 36-56.

Gruber T.R., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition, 2(5), 199-220.

Giaretta P. and Guarino N., 1995. Ontologies and knowledge bases: towards a terminological calrification. Towards very Large Knowledge bases: Knowledge Building and Knowledge sharing, 25-32.

Borst W., 1997. Construction of Engineering Ontologies. Thesis (PhD), Universiteit Twente.

Swartout B., Patil R., Knight K., et al, 1997. Toward Distributed Use of Large-Scale Ontologies. Ontological Enginering Aaai Spring Symposium.

Guarino N., 1998. Formal Ontology and Information Systems. In N. Guarino(ed.), Formal Ontlogy in Information Systems. Proceeding of FOIS'98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, 3-15.

Fensel D., Harmelen F.V., Horrocks I., et al., 2001. OIL: an ontology infrastructure for the Semantic Web. Intelligence Systems IEEE, 16(2),38 – 45.

Colin Atkinson and Thomas Kuhne, 2003. Model-driven development: A Metamodeling Foundation. IEEE Software 20(5) , 36-41.

OMG, 2014. Ontology Definition Metamodel. Object Manangement Group. Available from ： www.omg.org/spec/ODM/. [accessed 14 April 2015]

Horridge M., 2011. A Practical Guide To Building OWL ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3.

Li X.B., 2013. A DSM-based Multi-Paradigm Composable Modeling Method for Effectiveness

Simulation. Thesis (PhD). National University of Defense Technology, 44-45. (in chinese)
Reihhartz-Berger I. and Sturm A., 2004. Behavioral domain analysis – the application-based domain modeling approach. Lecture Notes in Computer Science, 410-424.
Cho H., 2011. A Demonstration-Based Approach for Designing Domain-Specific Modeling Language. SPLASH'11 Companion, 51-53. October 22-27, Portland, Oregon, USA.
Rath I. and Varro D. Challenges for advanced domain-specific modeling frameworks.
Zbib R., Jain A., Bassu D., et al., 2006. Generating domain specific graphical modeling editors from meta models. Proceddings of the 30th Annual International Computer Software and Application Conference (COMPSAC'06).
Walter T., Parreiras F.S., Staab S., 2014. An ontology-based framework for domain-specific modeling. Softw Syst Model, 83-108.
Guizzardi G., 2005. An ontology-based approach for evaluating the domain appropriateness and comprehensibility appropriateness of modeling languages. MODELS 2005, 691-705.
Guizzardi G. Ferreira Piress L., Sinderen V.M.J., 2002. On the role of domain ontologies in the design of domain-specific visual modeling languages. University of Alabama at Birmingham.
Tarias R., Mernik M., Gray J., 2009. Using ontologies in the domain analysis of domain-specific languages. Lecture Notes in Computer Science, 332-342.