# Adapting Simulation Modeling to Model-Driven Architecture for Model Requirement Verification

Fuqi Song, Gregory Zacharewicz and David Chen
*Univ. Bordeaux, IMS, UMR 5218, F-33400 Talence, France*

Keywords: Model-Driven Architecture, Model Verification, Ontology, DEVS, Simulation Interoperability.

Abstract: In order to automate the software development process and enhance interoperability and reusability of it, Model-Driven Architecture (MDA) was proposed aiming to achieving these goals. MDA mainly involves creating models and performing model transformation, it is a high model-intensive process. However, how to verify the models by respecting to the requirements becomes a concern in the automation process. In this paper, we describe the work of proposing a method using ontology as information exchange media to adapt simulation modeling to MDA towards this goal. Simulation is integrated loosely to MDA for verifying the models with the pre-defined requirements to check if expected goals are fulfilled. An illustrative case study is given to explain visually the method.

## 1 INTRODUCTION

Model-Driven Architecture (MDA) was proposed to solve some issues in traditional software development, such as, interoperability and reusability on different platforms, efficiency of software development. In MDA, the models are created in different levels concerning different aspects of software development. Between different levels of models, model transformation is applied to convert model from one to another. It aims to automate the whole process and generate the final codes and applications. The whole process of creating models and transforming models is high model-intensive. In the process, there is no verification and validation of the models by respecting the requirements. We consider that simulation could aid to verify the models by executing models in practice.

How to verify the models before the software is developed? An intuitive idea is to simulate the models before they are used for further steps. If the issues are identified in early stage of development, the models could be improved and save costs by avoiding the application modifications in future. However, not all the models created in MDA can be simulated. Generally, the models are classified as dynamic and static models. The major difference is that dynamic models describe the abstraction of systems at run-time, whereas static models capture the unchanged features regardless at run-time or not. The scope of the verification is among dynamic models, such as, activity diagrams and state chart. The static models, such as, class diagram and object diagram, are beyond of the scope.

In order to carry out simulation, model creation is the prerequisite. The proposal of this paper is to adapt models in MDA to simulation modeling process. It is different from model transformation from one model to simulation model. The adaptation is a loose connection rather than tight mapping relation. The media to exchange information between MDA models and simulation is ontology. Ontology is a formal representation to conceptualize concepts and the relations among them. The data of MDA models is described by a specific ontology, and the ontology is transformed to the other side to aid the creations of simulation models. Also, expected results or goals are described in this ontology for verifying the model. Another specific ontology with verification criterion and results will be generated and sent back to MDA models for potential improvements.

Ontology is brought to simulation in recent research. As stated by Turnitsa *et al.* (2010), two of the roles of ontology in simulation are: ontology as specification and common access to information. In this work, ontology has the two uses in facilitating interoperability between MDA model and simulation.

Song F., Zacharewicz G. and Chen D..
Adapting Simulation Modeling to Model-Driven Architecture for Model Requirement Verification.
DOI: 10.5220/0004585603020309
In *Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (SIMULTECH-2013), pages 302-309
ISBN: 978-989-8565-69-3

The rest of the paper is organized as follows. Section 2 recalls MDA and simulation modeling, also analyzes the links between MDA and simulation. Section 3 describes the proposed method with ontology to adapt simulation to MDA for aiding the requirements verification. Section 4 demonstrates a case study to illustrate the proposed method. Section 5 draws some major conclusions and extends the future work.

## 2 MDA AND SIMULATION MODELING

### 2.1 Model-Driven Architecture (MDA)

The model for developing computer systems and applications could be represented concisely in Figure 1. In real world, there are many issues and work need to be solved and done, and the goal is to construct computer systems to do them, namely, to find computational solutions to the problems in real world.

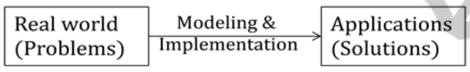

Figure 1: Relation of real world and computer system.

The process of developing applications has been transformed from procedural towards component-based years ago. However, some major issues still exist in software development, such as platform portability, interoperability issues and productivity problems. MDA was proposed by Object Management Group (OMG) on 2001 to improve the process and overcome the weakness in the other approaches. The main goal of MDA is to facilitate portability, interoperability and reusability of systems (OMG, 2003). Model-Driven Engineering (MDE) and model transformation are the bases and relevant research areas to MDA. MDE technologies provides promising methods to solve the inability of third-generation languages to alleviate the platforms complexity and express domain concepts effectively (Schmidt, 2006). Model transformation enables the conversion between models from high abstract level to low abstract level, ultimately to implementation. Kuster (2006) formalizes the model transformation with mathematics and define a set of criteria to validate the transformation. Czarnecki and Helsen (2006) give a feature-based survey of the approaches.

In MDA, three levels of models are defined by focusing on certain aspects in software development life cycle as shown in Figure 2. The top level is Computation Independent Model (CIM), which is independent to the future systems and focuses on only business requirements and models. The next level is Platform Independent Model (PIM). PIM describes the requirements of software system regardless specific platform going to be used. Last level is Platform Specific Model (PSM), which concerns the models to be used on a specific platform, for example, oracle database schema and java beans. MDE and model transformation automate the process of conversion from abstraction to implementation.

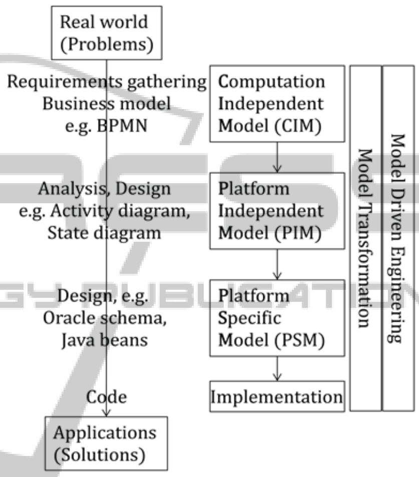
Real world (Problems)
Requirements gathering Business model e.g. BPMN
Computation Independent Model (CIM)
Analysis, Design e.g. Activity diagram, State diagram
Platform Independent Model (PIM)
Design, e.g. Oracle schema, Java beans
Platform Specific Model (PSM)
Code
Implementation
Applications (Solutions)
Model Transformation
Model Driven Engineering

Figure 2: Illustration of MDA.

### 2.2 Simulation Modeling

Simulation is the process of simulating real world in order to observe and verify the real world from certain perspectives. Before performing simulation, the models, which the simulation follows, should be created. Thus there is another step involved: modeling. In the book of theory of simulation and modeling written by Zeigler (1976), a generic relation among real world, modeling and simulation is given as Figure 3.

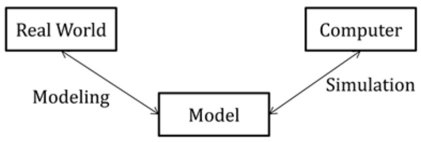

Figure 3: Basic elements and relation of modeling and simulation (Zeigler, 1976).

By focusing on certain aspects of real world, the reality is abstracted and created into models, namely, models are the abstraction of real world from a specific perspective. Simulation is performed by following the models on computer systems with simulation languages. Pollacia (1989) gives a review of discrete event simulation languages. The objective of simulation is to observe the behaviors of one system, in order to understand the reality and predict the behaviors in future. It is widely applied in many areas.

### 2.3 Connections between MDA and Simulation Modeling

Bourey and Seguer (2011) describes that a model is a consensus about an abstraction of phenomena of the real world, in other word, a representation of an aspect of the world for a specific purpose. From this definition, we can see that when talking about models, the objectives and scope should always be stated, otherwise pure models are not helpful. In general, the models in MDA and models for simulation are different, they emphasize on different aspects. Models in MDA try to understand the reality in real world or future systems by exploring certain aspects,for instance, using BPMN to analyze the business process, using E-R diagram to analyze the data structure, or using state chart to study the changes among different states. However, the models for simulation focus more on the execution, the models need to be executable. Major elements for simulation are event, time and behaviors. Thus with the aides of model transformation and extra information, the models created in MDA can be executed for simulation.

In MDA, creating models and transforming models are two major activities involved, namely, MDA is a model-intensive process. Models are staying at an abstract level. An existing issue is how to verify whether these models are good. A general standard is that *M* is a good model of *S* if *M* makes it possible to answer predefined questions about *S* in a satisfactory way (Ross). This standard is quite blur and hard to formolize. But we can see that a good model should respect the facts as one of the main objectives of simulation. Requirement verificatioin could be a strong complement to MDA. The purpose of this paper is to introduce simulation to MDA for model verification with respecting to the requirements at a high level.

## 3 ADAPTING SIMULATION MODELING TO MDA

In this section, we describe how to apply simulation with MDA. First, a general structure to combine simulation with MDA is illustrated, and then the specific ontology for information exchange is proposed.

### 3.1 General Structure

According to the structures of MDA and simulation and the connections between them discussed in previous section, first a general structure for adapting simulation to MDA is illustrated in Figure 4. CIM, PIM and PSM belongs to modeling process, they will create different models. A simulation process will be used to verify the models by respecting the real requirements. An iterative process between modeling in MDA and simulation is constructed to involve simulation in MDA. The information concerning event and behaviors will be sent to the other side to perform simulation process. The simulation results will be compared with expected results and generate a verification report. This report is used to improve the modeling process in MDA. If necessary, several iterative processes will be carried out.

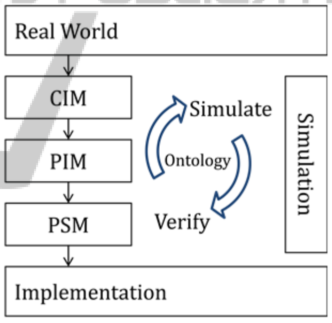

Figure 4: Combination of MDA and simulation.

Figure 5 demonstrates an elaborated structure with exchange information. The formalism of information exchange and sharing between the two sides is represented by ontology. Ontology of Information Exchange (OoIE) describes the models in MDA and is sent to the other side for aiding to create simulation models.

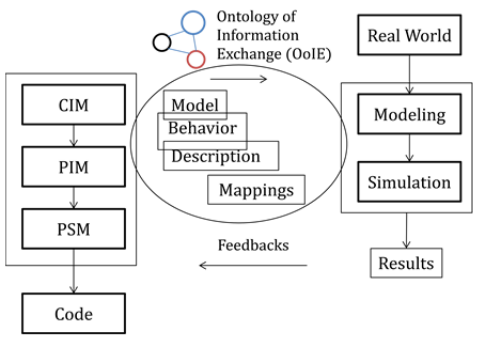

Figure 5: Structure of adapting simulation to MDA.

Besides the ontology OoIE, in order to create simulation models, additional information is required, such as time constraints and sequential limitations. The information will be added when creating the simulation models. Also, the expected results of models in MDA are described in OoIE, in order to verify and validate whether the built models correspond to the requirements.

## 3.2 Ontology of Information Exchange (OoIE)

Ontology represents a set of concepts and relations between them in a specific domain of knowledge. Ontology has several advantages in facilitating interoperability of models and exchange information, especially, the semantic representation in term of knowledge sharing and exchange. It represents the semantics explicitly, which can be processed and understood by computers. In (Song et al., 2012a), the authors classify three roles that ontology played in contributing to interoperability of enterprise modeling: concepts specification, knowledge sharing, and concepts annotation. As a type of media for knowledge sharing, ontology plays an effective role. There are various languages to represent ontology, which can be chosen depending on particular demands. An investigation of ontology languages is given in (Song et al., 2013).

At the different levels of MDA process, the models are created to represent certain aspects of requirements. Each model is built in a specific modeling language, such as using BPMN to gather and analyze the business requirements at CIM level, using activity diagram to model the main functions involved in the process at PIM level. OoIE aims to build a semantic representation at a generic level to describe the main concerns of the models. The ontology will be sent to the simulation side and aid to create simulation models. The ontology describes mainly three parts: model itself, description and mappings as shown in Figure 6.

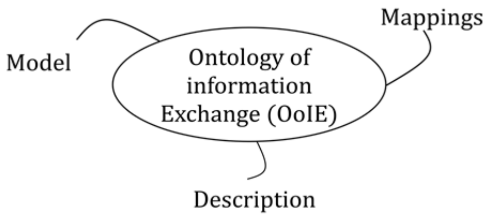

Figure 6: Ontology of Information Exchange (OoIE).

The models are the ones from MDA, different models will be formalized and represented with ontology in the form of meta model. The requirements for creating the models will be added to OoIE. The functions are twofold: 1) aiding to create simulation models, 2) verifying the consistency between requirements and built models. Mappings are for alignment between MDA models and simulation models.

The purpose to build mappings between MDA model and simulation model is to facilitate the creation and transformation of simulation models. The mapping is carried out based on the ontology of models, thus a hypothesis is that the description ontology of MDA models and simulation models are existing. The description ontology is a semantic representation of models and reality. To find the equivalent concepts between the source ontology and target ontology, several levels from different aspects are investigated. The mappings are set up by measuring the similarity between different elements in ontology, and the measurement is based on different algorithms and rules. Song et al. (2012b) present a multiple strategies-based ontology matching approach. In this approach, several matching algorithms are proposed from different levels of ontology, and the different matching results are combined by an analytic process AHP by balancing the weights of each matcher. After having obtained the mappings between equivalent concepts, the creation of simulation models would be facilitated by understanding the connections between the models.

## 3.3 Models Adaptation

With the help of OoIE, a general structure of model adaptation is illustrated in Figure 7. The whole process is started from the models in MDA, and only part of the models is taken into account. Because of the natures of models, it is argued that only dynamic models in MDA could be adapted to simulation models. The major difference between dynamic and static models is whether the model captures the behaviors in system run times or not. Normally, static models do not include the elements of

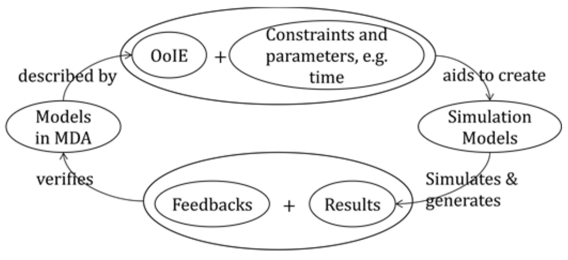

Figure 7: Model adaptation between MDA and simulation.

behaviors and time constraints, so we consider that this kind of models is beyond of the scope.

OoIE describes the models in MDA, in order to aid to create simulation models, some additional information is needed. OoIE mainly describes the data and behaviors occurred in MDA, however, to adapt to specific simulation models, constraints and parameters should be complemented, such as time limitation and sequential requirements. This process is different from model transformation. Model transformation is a strict mapping process to transform elements from source models to target models. The whole process is supposed to be automatic. However, in this process, the model adaptation requires a loose mapping between MDA models and simulation models, and the process is manual work and adjustment involved.

Simulation models will generate simulation results after execution. In order to verify if the models are executed as expected, the generated results are compared with expected results, which are described in OoIE. The results are in two forms, first form is the general execution results of models, such as the specific behavior generated or state arrived. For some models and situations, they cannot generate general results as all the models do; the results vary depending on different scenario. In this case, a specific case or scenario and expected results are described in OoIE.

The results of verification are sent back to the MDA side. The differences where the goals are not reached will also be noted, so that to improve the models. If the validation results do not reach to a satisfactory level, more iterative processes will be done.

## 4 DEMONSTRATION

A study case is made to illustrate the proposed idea and method. In this study case, first the scenario in the domain of manufacture and background of case are described. At PIM level in MDA, a state chart is created according to the requirements. And then in order to verify the modeled state chart, we follow the methodology described in above sections to verify and improve the built model.

### 4.1 Scenario

In product manufacturing, order processing is an ordinary requirement. In MDA, after requirements gathering at CIM level, the client want to focus on the state change of order processing at PIM level using state chart. The requirements are described as follows:

*The state of order changes from "initial" to "Ordered" after selection products. And then the system check the stock, if the product is out of stock, then the order will be cancelled and the order process is completed. If the product is available, then the order awaits the payment from user. After user pays the order, the manager will verify (manually or automatically) the payment. If the payment is received correctly, then the order is confirmed and the stock prepares to deliver the product. If the payment is not received correctly, then the order changes to state "awaiting payment". The user could cancel the order after ordering the products and before the products are delivered.*

An existing data model of orders is given in form of ontology in Figure 8. According to the description of scenario, the model is created in OWL, which is a standard ontology language and is used mostly to describe models in MDA. This model illustrates the knowledge of order state changing.

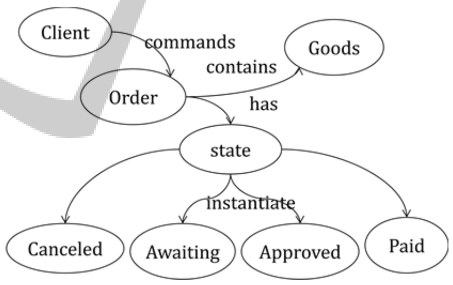

Figure 8: Data model of order states in OWL.

A version of state chart is created according to above description as shown in Figure 9. Now the demand to check whether the model corresponds to the requirements is required.

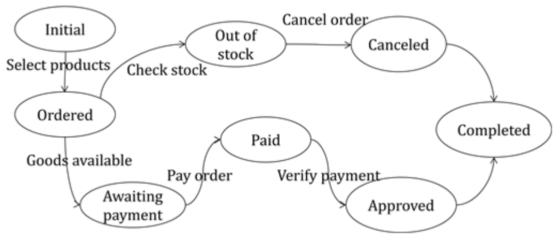

Figure 9: Modelled state chart of order processing.

### 4.2 Model Adaptation

In this section, first a meta model of state chart is described, and then the mappings between two models in different representation languages are represented. The discovered concepts are used in simulation models.

### 4.2.1 Model Data

The description of state chart meta data can be in different ways, Favre (2010) describes a metadata model of state chart. A simplified ontology is used to describe the state chart as shown in Figure 10. The model starts with an initial state and ends with an end state, between them, there are iterative states with transitions.

The possible sequences of state transition derived from model in Figure 9 are:

1) Initial- Ordered-out of stock- canceled-completed;
2) Initial- Ordered- Awaiting payment- paid-confirmed- delivered- completed.

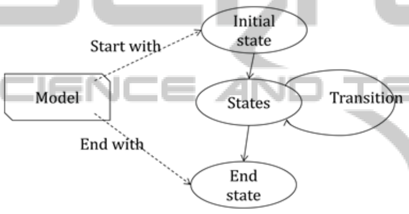

Figure 10: Description of simplified state chart.

### 4.2.2 Mappings

Figure 11 shows a data model represented in System Entity Structure (SES), it is an existing model used in simulation side. SES is presented by Zeigler and Hammonds (2007), which plays roles as ontology framework and information exchange media, to describe static and dynamic work state change in simulation modeling with DEVS formalism.

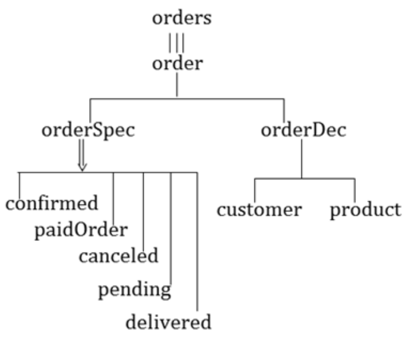

Figure 11: Data model of order state described in SES.

The second step is to find mappings between the model in SES (see Figure 11) and the model in OWL (see Figure 8). This step enables to find correspondent concepts between two data models. The matching method used is from Song *et al.* (2012b). The left part is in format of SES and right part is in OWL, and discovered alignments are labeled in dotted line with a value of similarity. The similarity measures the sameness of two concepts in term of semantic. For example, the similarity between "*pending*" and "*Awaiting*" is 0.6, and "*paidOrder*" and "*paid*" is 0.9. The discovered correspondences are also listed in Table 1. The threshold of similarity is set as *th* = 0.5 manually, for these correspondences, whose similarity value is greater than 0.5, they will be used. The equivalent concepts will be applied in simulation models.

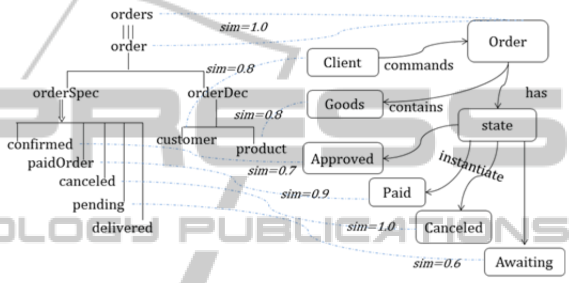

Figure 12: Alignment of two data models.

Table 1: Discovered correspondences.

| Model in OWL | Model in SES | Sim. |
|---|---|---|
| order | order/orders | 1.0 |
| client | customer | 0.8 |
| approved | confirmed | 0.7 |
| paid | paidOrder | 0.9 |
| awaiting | pending | 0.6 |
| canceled | canceled | 1.0 |
| - | delivered/ orderSpec/orderDec | - |
| state/commands/ contains has/instantiate | - | - |

## 4.3 Verification and Improvements

Simulation tools and languages can help to facilitate this step, such as, Wood *et al.* (2008) uses VHDL to represent UML state diagram. At current stage, we simulate the process using LSIS_DME (Baati et al., 2007) that proposes to create G-DEVS graphical models. The snapshot is shown as Figure13. The generated DEVS model needs to be defined by temporal time life, for instance, the state time life is represented in the DEVS simulation model.

According to the metadata of state chart in Figure 10 and requirement description, the sequence of real state changes are as follows, where [xxx]* refers to a loop:

1) Initial-Ordered -canceled-completed
2) Initial- Ordered- [pending- paidOrder]*- canceled- completed
3) Initial- Ordered-[pending- paidOrder]*- confirmed- canceled- completed
4) Initial-Ordered- [pending- paidOrder]*- confirmed- delivered- completed

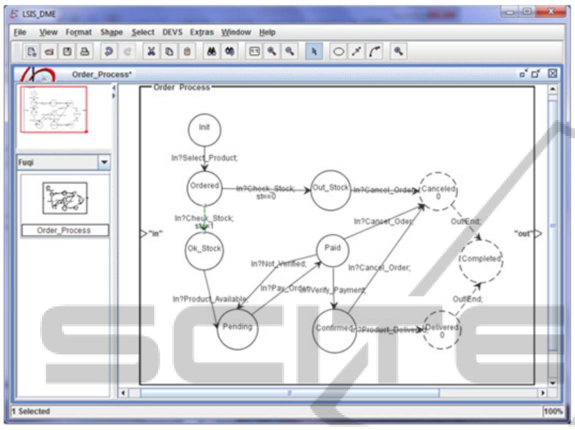

Figure 13: Snapshot of simulation model with LSIS_DME.

Compared with the sequences, which are described in the first version of modeled state chart (see Figure 9), an improved model is given in Figure 14. The dotted line refers to the added entities. The improvements are as follows:

1) Remove the state “out of stock”;
2) Add state “delivered”;
3) Add transition between “paidOrder” and “Pending”;
4) Add transition between “paidOrder” and “canceled”;
5) Add transition between “confirmed” and “canceled”.

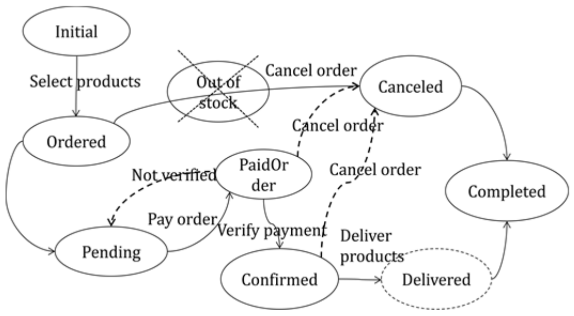

Figure 14: Improved state chart of order processing.

### 4.4 Discussion

There is little work about combining MDA and simulation modeling. In (El Haouzi, 2006), the author proposes a method combining MDA and HLA (High Level Architecture) to improve the deficiencies of current simulation methods at the level of interoperability and reusability. Cortellessa et al. (2007) extends MDA to non-functional-MDA framework to allow generating non functional models. The extended framework allows verifying the requirements of models. The work described in this paper differs from the above two methods. The proposal advocates running the models and then getting the real feedback by comparing the results with expected requirements. The work proposes a generic methodology of combination MDA and simulation modeling process. In the process of adaption, ontology plays a key role as the media of information exchange.

Benjamin et al. (2007, 2009) applies ontology as information exchange media to enable integration and interoperability of simulation models. In the approach, three kinds of ontology: domain ontology, Community Of Interest (COI) ontology and simulation tool ontology are extracted and created. COI ontology builds a bridge of common concepts and knowledge between simulation models and domain knowledge, ultimately to enable the communication and interoperability of models. Ontology is also adopted as media of information exchange in the approach of this paper. However, its purpose is to enable the knowledge sharing between MDA and simulation modeling. Also, ontology is applied as simulation model to enhance semantic in simulation, for instance, Silver et al. (2011) proposes Discrete-event Modeling Ontology (DeMO) to improve semantic search and integration of heterogeneous information sources.

## 5 CONCLUSIONS AND FUTURE WORK

In this paper, we described a generic methodology to adapt simulation modeling to MDA for verifying the models by respecting to the contextual requirements. It aims to complement and enrich the MDA process. Combination of MDA and simulation can contribute to the model verification at requirement level. The application of ontology as information exchange and sharing media provides a semantic and loose relation between models and simulation.

At current stage, the work is ongoing, and the work that has been done focusing on defining a general methodology and framework, also giving a typical illustrative demonstration. Remaining work mainly concerns elaborating the method and giving operational application steps, also enabling the automatic process. First, the rules and formalism of OoIE will be defined. This specific ontology can vary enormously depending on different models, so different kinds of ontology concerning different models will be defined. Second, the way of exchanging information between the two sides needs to be elaborated by defining the interface and format in details. Third, the way and criterion of verification will be elaborated in order to enable automatic verification.

## REFERENCES

Turnitsa, C., Padilla, J. J. & Tolk, A. 2010. Ontology for Modeling and Simulation. *In: Winter Simulation Conference (WSC) 2010*. 643-651.

OMG. 2003. MDA guide version 1.0.1. Available: http://www.omg.org/mda/.

Schmidt, D. C. 2006. Guest Editor's Introduction: Model-Driven Engineering. *Computer,* 39**,** 25-31.

Kuster, J. M. 2006. Definition and validation of model transformations. *Software & Systems Modeling,* 5**,** 233-259.

Czarnecki, K. & Helsen, S. 2006. Feature-based survey of model transformation approaches. *IBM System Journal,* 45**,** 621-645.

Zeigler, B. P. 1976. *Theory of modelling and simulation*, Wiley.

Pollacia, L. F. 1989. A survey of discrete event simulation and state-of-the-art discrete event languages. *SIGSIM Simul. Dig.,* 20**,** 8-25.

Bourey, J.-P. & Seguer, R. G. 2011. Model, Modeling, Metamodel, Metamodeling... Available: http://www.easy-dim.org/reunions/supports-du-module-recherche-en-interoperabilite-des-systemes-de-lecole-des-jd-macs/4-Meta-Modelling%20JD%20MACS%2020110606.pdf (Accessed 18 Dec. 2012).

Song, F., Zacharewicz, G. & Chen, D. 2012. Ontology for Contributing to Enterprise Interoperability. *In: UNITE 2nd Doctoral Symposium: R&D in Future Internet and Enterprise Interoperability*, 2012a, Bulgaria. Avangard Prima, 127-134.

Song, F., Zacharewicz, G. & Chen, D. 2013. An Ontology-Driven Framework towards Building Enterprise Semantic Information Layer. *Advanced Engineering Informatics*, 27, 38-50

Song, F., Zacharewicz, G. & Chen, D. Year. Multi-strategies Ontology Alignment Aggregated by AHP. *In: 16th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, 2012b San Sebastian. IOS Press, 1583-1592.

Favre, L. 2010. *Model Driven Architecture for Reverse Engineering Technologies: Strategic Directions and System Evolution*, IGI Global.

Zeigler, B. P. & Hammonds, P. E. 2007. *Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange*, Academic Press.

Wood, S. K., Akehurst, D. H., Uzenkov, O., Howells, W. G. J. & Mcdonald-Maier, K. D. 2008. A Model-Driven Development Approach to Mapping UML State Diagrams to Synthesizable VHDL. *Computers, IEEE Transactions on,* 57**,** 1357-1371.

Baati, L., Frydman, C. & Giambiasi, N. 2007. LSIS_DME M&S environment extended by dynamic hierarchical structure DEVS modeling approach. *Proceedings of the 2007 spring simulation multiconference - Volume 2.* Norfolk, Virginia: Society for Computer Simulation International.

El Haouzi, H. 2006. Models simulation and interoperability using MDA and HLA. *In: Interoperability for Enterprise Software and Applications: Proceedings of the Workshops and the Doctorial Symposium of the Second IFAC/IFIP I-ESA* 2006. ISTE, 277-284.

Cortellessa, V., Di Marco, A. & Inverardi, P. 2007. Non-Functional Modeling and Validation in Model-Driven Architecture. *In: Software Architecture, 2007. WICSA '07. The Working IEEE/IFIP Conference on*, 6-9 Jan. 2007.

Benjamin, P., Akella, K. & Verma, A. 2007. Using ontologies for simulation integration. *In: Proceedings of Winter Simulation Conference* 2007. 1081-1089.

Benjamin, P. & Akella, K. 2009. Towards ontology-driven interoperability for simulation-based applications. *In: Proceedings of Winter Simulation Conference 2009*. 1375-1386.

Silver, G. A., Miller, J. A., Hybinette, M., Baramidze, G. & York, W. S. 2011. DeMO: an ontology for discrete-event modeling and simulation. *Simulation*, 87, 747-773.