

EMF based Domain Specific Modeling for Distributed Simulation Systems: An Outline

Zhu Zhi^{1, a}, Li Haibing^{2, b}, Lei Yonglin^{3, c}, Zhu Yifan^{4, d}

^{1, 2, 3, 4} College of Information System and Management, National University of Defense Technology, Changsha, Hunan, P. R. China, 410073

¹zhuzhi@nudt.edu.cn, ²hbli910913@163.com, ³samuelyonglin@gmail.com, ⁴yfzhu@nudt.edu.cn

Keywords: Distributed Simulation Systems; DSM; EMF

Abstract. The design and development of distributed simulation systems are a complex process, and the Domain Specific Modeling (DSM) methodology is progressively gaining popularity as it uses domain knowledge directly to specify systems in a higher level of abstraction. In this research plan an Eclipse Modeling Framework (EMF) based domain specific modeling method is proposed to realize the abstract models specification and systematically transform them into concrete implementations automatically. With the toolkits that EMF provides with, the method reduces the gap between problem domain and technological implementations mainly by means of meta-modeling, models transformation and code generation technology. Finally, a proper research schedule, displayed as a research topic-time table, is configured to achieve the idea.

Introduction

Distributed Simulation Systems and Its Modeling Requirements

Distributed simulation systems [1] are typical complex systems employed in many fields like sensor networks, transportation monitoring, emergency management and advanced command and control etc. These systems are usually distributed in a large-scale, consist of a spectrum of nodes such as detecting system, tracking system, controlling system, displaying system etc., and are expected to work in a dynamic environment to cooperatively perform a common mission and withstand unexpected errors. The exploration of supportable novel methodologies for distributed simulation systems development is therefore a special type practice of great importance.

Distributed simulation systems specification is usually required to be developed from different views, aspects and level of abstraction. For these system models are featured with scalability, multi-disciplines, flexibility and error-prone, their development and maintenance become to be more complex since the ability of cooperation has to be considered additionally. In order to deal with these features that resides in the dynamic world in which changes can occur frequently, the corresponding modeling methodologies shall provide the capabilities to support architecture centric modeling, disciplines division, user-friendly modeling, error early detection and computer supported cooperative design. However, traditional design and implementation have many limitations in face of these increasing modeling requirements, and the contradiction between supply and demand side prompt the formation of novel techniques to leverage it.

Traditional Design & Implementation Limitations

Traditional methodologies used in distributed simulation systems design and implementation can be roughly defined as domain neutral methodologies which are independent of the intensive domain knowledge. From a technical point of view, they are usually based on unified simulation protocol (like HLA [2]), model specification (like SMP2 [3]) and modeling formalism (like DEVS [4] and UML [5]). Based on a common infrastructure, the unified modeling and simulation mechanisms listed here are important to allow us to realize a large-scale communication and interoperability, but they are not enough to reach all the modeling requirements mentioned above without considering the intensive domain knowledge from a domain specific point of view.

- 1) With the united mechanisms stated above it may be difficult for developers to enter an absolutely new field to learn any mechanism. Besides, developers have to translate concepts or relationships abstracted from a domain into the expression that united mechanisms can understand.

- 2) System clients cannot participate in the development process because of strict distinction between the problem domain and the technology domain, resulting in low quality products or even inaccuracy of the resultant system.
- 3) The result system, not supported by a composable modeling framework, may be not enough robust, i.e. if the scenario changes, the whole systems may have to be partially or even completely redesigned and reconfigured.
- 4) Additionally, united mechanisms appear to be low productive or even useless when specifying some domain special properties. For example, UML can not specify the behavior that an object goes from location A to location B with a probability of 0.5.

The Alternative Methodology

In this research plan, EMF (Eclipse Modeling Framework [6]) based domain specific modeling is proposed to be address those problems stated earlier in a way that can be easily updated or even fully changed, with abstracting concepts and relationships from the research domain, additionally adopting the technological supportable toolkits of Eclipse Modeling Framework. By this, it can not only result in improving productivity, quality and complexity hiding in distributed simulation systems development, but also responding to frequent changes that occur in the outside world. The research plan is organized in the following sections. In section 2, the research background, mainly including the related methodologies and technologies, is detailed. The research contents, containing research goal and mission, research method and the technological framework are presented in section 3. Finally, the last section gives a research schedule for the research plan.

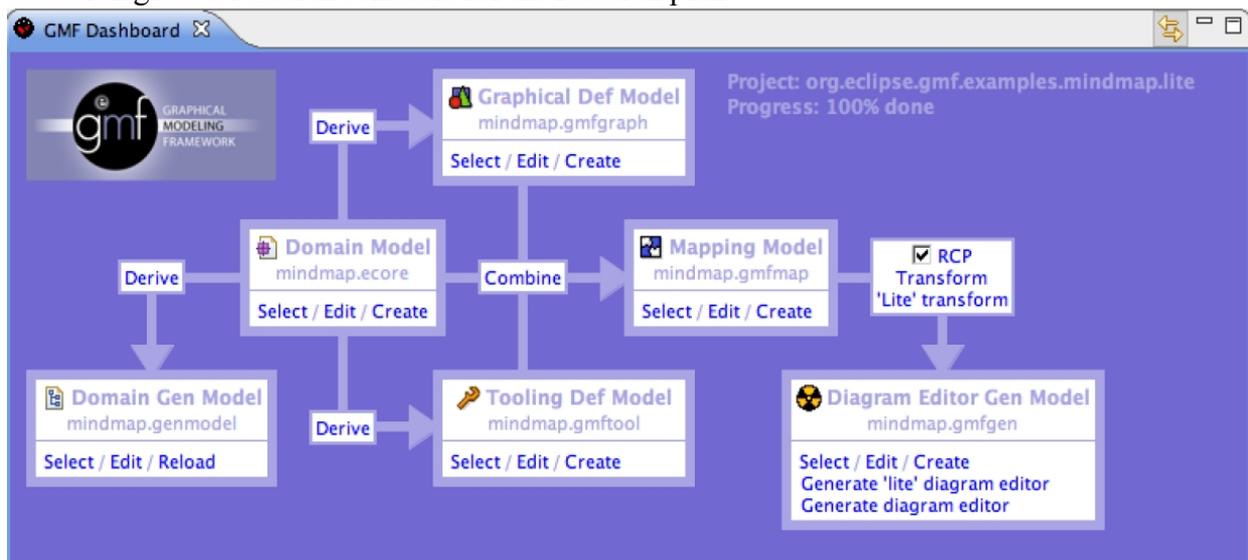


Figure 1 GMF-based workflow

Research Background

Meta-modeling

The goal of meta-modeling is to create a meta-model. From the definition proposed by Object Management Group (OMG), a meta-model is a model that defines the language for expressing a model [7]. Just like the specification of a language, it has two kinds of modeling specification, the modeling syntax and the modeling semantics, to describe concepts, relationships between these concepts and composable rules etc. Here the word “meta” is used to explain that meta-modeling locates in a level of abstraction higher than that of modeling. This hierarchical philosophy can also be extended to the activities of meta-metamodeling. A four-layer meta-modeling architecture, taken from [8], is created to be the following table 1.

TABLE 1 FOUR-LAYER META-MODELING ARCHITECTURE

| layer | activity | target | description |
|-------|-------------------|----------------|--|
| M3 | meta-metamodeling | meta-metamodel | Defines the language for describing meta-models |
| M2 | meta-modeling | meta-model | An instance of a meta-metamodel. Defines the language for specifying a model |
| M1 | modeling | model | An instance of a meta-model. Defines a language to describe an information domain. |
| M0 | Programming | user objects | An instance of a model. Defines a specific information domain |

Eclipse Modeling Framework

Eclipse Modeling Framework (EMF) is a modeling framework and code generation facility for building tools and other application based on a structured data model. From a model specification described in XMI (XML-based Metadata Interchange), EMF provides tools and runtime support to produce a set of Java classes for the model, along with a set of adapter classes that enable viewing and command-based editing of the model, and a basic editor. Graphical Modeling Framework (GMF) is an EMF-based framework to generate visual editors with individual customization. Taken from [9], the GMF-based workflow is viewed as the above screenshot of dashboard of figure 1.

Domain Specific Modeling

Domain Specific Modeling (DSM) is a problem-oriented methodology which allows one to exactly specify the specific domain with the concepts and relationships that map closely to the given domain. To provide the computing and execution of these domain models, technologies of model transformations and model anchoring need support the automated generation of pre-implemented models mapped from those domain models. A three-level architecture of DSM, taken from [10], is illustrated in the following figure 2.

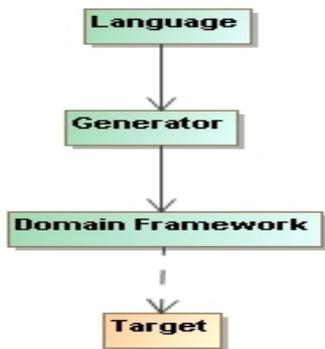


Figure 2 Architecture of DSM

- Ÿ A Domain Specific Language (DSL) provides an abstraction mechanism by providing concepts and rules within a language that represent things in the application domain, rather than concepts of a given programming language, to deal with complexity.
- Ÿ A code generator provides a generational automation by specifying how to access domain models, how to extract information from them, and how to transform information extracted into output.
- Ÿ A domain framework provides the interface between the generated code and the underlying platform. It can be very thin or even nonexistent if the generated code can directly be the input for some engine or call the interface of the target environment.
- Ÿ The underlying target provides an environment that consists of different infrastructure on which our generated applications run.

Architecture centric effectiveness simulation modeling

From the definition of architecture for system and software engineering, architecture centric modeling methodology is proposed by Lei [11] for effectiveness simulation to achieve the requirements of composable modeling, domain specific modeling and model evolvement in typical complex systems. This proposal is made up of structural architecture, physical behavior architecture and cognitive architecture in the horizontal direction from three viewpoints, and domain modeling and application modeling in the vertical direction from two layers.

On the one hand, firstly the structural architectural modeling defines fundamental model elements and structural relationships between these elements in an abstract manner. Secondly determined by the natural laws and constraints, the physical behavior architecture describes the behaviors within physical domain for elements defined in the structural architecture and the dynamic relationships among those elements. Thirdly depending on the free will of human beings, cognitive behavior architecture specifies the behaviors within cognitive domain where changes occur frequently across different applications.

On the other, domain modeling is the process of abstracting common domain invariable knowledge (DIK) from a given domain, while the application modeling is the activity of extracting individual application variable knowledge (AVK) framed in the composable modeling framework of DIK [12].

Research Contents



- Goal of a mission in a particular situation is a preferred outcome that can be obtained with a specified time period [14].
- Mission is what an object should work on. For example, the mission of an institute is doing research.
- Ways are methodologies that an object takes to conduct a mission.
- Means are technologies that supporting the given methodologies to complete a mission.

Research Goal

As stated in the earlier section, current simulation protocol, model specification and modeling formalism cannot achieve the increasing requirements of distributed simulation systems. Within the technological framework of EMF, the mission of this research in the visiting period is to explore system architecture framework that covers the DIK and AVK for distributed simulation systems, so as to create a user-friendly domain specific language that represents the concepts and rules in distributed simulation systems, and to invent a novel model transformations mechanism that can efficiently transform the domain models into the pre-implemented models or the running application, resulting the research goal of improving productivity, quality and complexity hiding.

Research Method

The research method, consists of the following methodological procedure, domain specific language and model transformation mechanism, is the ways to be commonly used to instruct systemic transformation that abstract models of distributed simulation systems into the concrete application.

Methodological procedure

The methodological procedure, as is illustrated in the following figure 4, is made up of a series of steps and several step-outputs. These steps in the methodological procedure include: domain specification whose output is system architecture framework, meta-modeling whose output is meta-models (e.g. domain specific language), model transformations whose output is domain specific modeling environment, and user-friendly modeling [15-16]. Certainly there are several activities not illustrated in this figure after the end step of user-friendly modeling, these activities are some model mapping mechanisms to generate the preferred running application.

Domain specific language

As stated in the previous section, domain specific language, commonly represented by meta-models, consists of syntax and semantics as same as the definition of language. The syntax has abstract syntax and concrete syntax: the former is used to denote modeling concepts, along with allowable relationships existed between these modeling concepts, while the latter is the representational form with notational symbols like rectangle, line, compartment etc. Similarly, the semantics has static semantics and dynamic semantics: the former is generally modeled as invariant conditions that can be defined in meta-models, while the latter is specified based on the concrete conditions that associate with particular instances.

Within the tools of EMF, domain specific language can be achieved by means of several formats. Here Annotated Java, XML Schema, Ecore-based tree and Ecore-based UML are illustrated sequentially in the following figure 5.

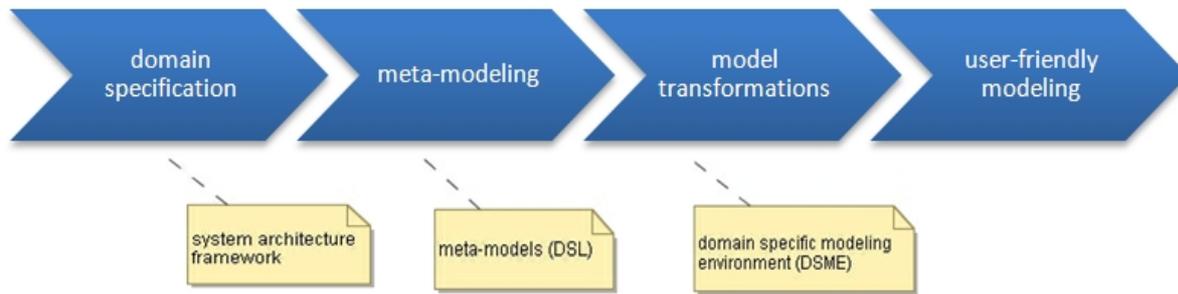


Figure 4 Methodological procedure

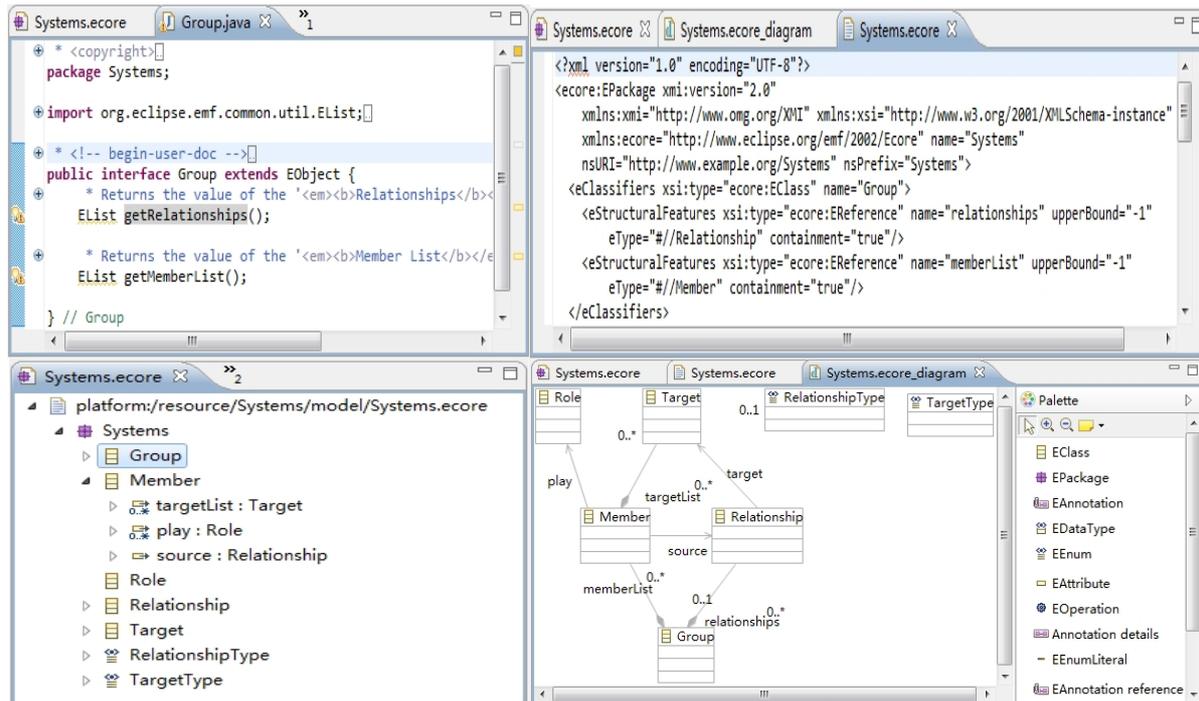


Figure 5 EMF based domain specific language achieved by (a) Annotated Java; (b) XML schema; (c) Ecore-based tree; (d) Ecore-based UML

Model transformation mechanism

Using the transformation capabilities with which EMF toolkits provides, model transformation defines the process of transformation from a source model into a target model. For the target model mentioned here can be specified in a diagrammatic form or a textual form, the corresponding Model to Model (M2M) and Model to Text (M2T) process take place respectively [11]. The former mechanism is developed in order to generate pre-implemented models required by EMF toolkits (like tool-model, gen-model etc.) or other formalisms (like DEVS, SMP2 etc.), which is supported by Atlas Transformation Language (ATL), Query/View/Transformation (QVT) and other artificial transformation rules. However, the latter is used to generate the running applications that we typically want to acquire, which is supported by Xpand template engine, Java Emitter Templates (JET), and other novel code generation mechanisms.

Technological framework

Actually technological framework is an organized collection of different technologies to help designers get a good master of technological infrastructure, so as to apply them effectively and to extend them if necessary. Working with the method mentioned above in the form of methodological procedure, domain specific language and model transformation mechanism is great, but it will not be achieved for designers to get a working application if without the technological infrastructure. Consequently, how to integrate plug-ins and tools from Eclipse Modeling Project and extend them is important for designers to obtain the final application.

Research Schedule

Given above statements, it is clear to what I shall proceed in the visiting meantime. Within the research background depicted in the section 2, my research is mainly decomposed into 6 topics, and my visiting time is separated into 8 sections based on the research contents.

Topic A: mastering the theory of distributed simulation systems

Topic B: abstracting the domain knowledge from distributed simulation systems

Topic C: constructing the domain specific language with a series of meta-models

Topic D: developing a domain specific modeling environment for distributed simulation systems

Topic E: modeling domain models with the language in the modeling environment

Topic F: generating distributed simulation systems application

References

- [1] Rajan, Hridayesh. Building scalable software systems in the multicore era. FSE/SDP Workshop on the Future of Software Engineering Research, FoSER 2010: 293-297
- [2] IEEE 1516.2-2000, IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT) [S]. 2010.
- [3] ESA . SMP 2.0 Handbook , EGOS-SIM-GEN-TN-0099 , Issue 1.2 , pp22-25 , 2005.
- [4] Bernard P. Zeigler, Steve B. Hall and Hessam S. Sarjoughian, Exploiting HLA and DEVS To Promote Interoperability and Reuse in Lockheed's Corporate Environment, SIMULATION 73:5, 288-295,1999
- [5] Mohammad S. Al-Aqrabawi, Combat System Modeling: Modeling Large-Scale Software and Hardware Application Using UML, Master Thesis, Virginia Polytechnic Institute and State University, 2001
- [6] EMF tutorials: <http://www.eclipse.org/modeling/emf/docs/>
- [7] Richard C. Gronback. Eclipse Modeling Project: A Domain-Specific Language Toolkit. Addison-Wesley Professional, 2009
- [8] Greg Nordstorm, Janos Sztipanovits etc. Metamodeling – Rapid Design and Evolution of Domain-Specific Modeling Environments. 6th Symposium on Engineering of Computer-Based Systems-ECBS. pp68-74, 1999
- [9] GMFTutorials: http://wiki.eclipse.org/Graphical_Modeling_Framework/Tutorial#Get_started
- [10] Steven Kelly, Juha-Pekka Tolvanen. Domain-specific Modeling Enabling Full Code Generation. Johan Wiley & Sons, Inc., 2008
- [11] Yonglin Lei, Ning Zhu etc. Architecture-oriented Combat System Effectiveness Simulation Modeling. In reviewing
- [12] LEI Y L, LI Q, et al. A Composable Modeling Framework for Weapon Systems Effectiveness Simulation. Systems Engineering-theory & Practice, 2013.11:2954-2966.
- [13] Weigping Wang. Network Design and Simulation Report for SoS. Group of SoS, 2014
- [14] Russell L. Ackoff. Towards A System of Systems Concepts. Management Science, 1971.
- [15] Gallardo J, Bravo C, Redondo MA. A model-driven development method for collaborative modeling tools. Journal of Network and Computer Applications, 2012. 35: 1086-105
- [16] Molina AI, Gallardo J, Redondo MA, Ortega M, Giraldo WJ. Metamodel-driven definition of a visual modeling language for specifying interactive group applications: an empirical study. Journal of Systems and Software, 2013. 86: 1772-1789