

# Model Driven Interoperability for Complex Simulation Systems

Xiaobo Li, Yonglin Lei, Weiping Wang, Qun Li

Institute of Simulation Engineering, College of Information Systems and Management,  
National University of Defense Technology

Changsha, PRC

{lixiaobo, yllei, wangwp, liqun}@nudt.edu.cn

**Abstract**—As the systems under study (SUS) of Modelling and Simulation (M&S) are becoming more and more complex, different simulation models and systems need to interoperate with each other to model and simulate the SUS more efficiently. M&S community has worked on simulation interoperability for many years, e.g., HLA is an IEEE standard developed to promote interoperability between different simulation systems. However, current research mainly concentrates on technical interoperability, while substantive interoperability is not effectively promoted. In this paper, related M&S research on interoperability is reviewed and how to use Model Driven Engineering (MDE) techniques to promote simulation interoperability is discussed. Finally a MDE-based simulation system development framework combining SMP2 and ontology techniques is proposed to illustrate how to achieve the model driven simulation interoperability. The framework adopts a MDE paradigm and uses ontology metamodels to construct a common knowledge base, which promotes both technical and substantive interoperability.

**Keywords**-Model Driven Engineering, Simulation Interoperability, Ontology, Simulation Model Portability 2 (SMP2)

## I. INTRODUCTION

Interoperability is a property referring to the ability of diverse systems and organizations to work together (interoperate). Interoperability is a fundamental research issue in many engineering fields because engineers require unification of efforts, tools and other artifacts to solve complex application problems. In M&S community, models, simulators and simulation systems working together makes M&S a powerful tool to achieve results that could not be available when working separately. M&S researchers have conducted extensive research on simulation interoperability, e.g., Simulation Interoperability Standardization Organization (SISO) was found in 2003 and Simulation Interoperability Workshop (SIW) is held annually.

Simulation interoperability is the combination of enterprise interoperability and software interoperability because simulation systems, on one hand, are developed by different organizations, and on the other hand, usually take the form of software systems. So some researchers devide simulaiton interoperability into two categories: technical interoperability and substantive interoperability enterprise interoperability and software interoperability [1]. The former refers to the capability of simulation systems to physically

connect and exchange data through those connections, which concentrate on the technical standards of simulation software implementation. The latter refers to the desired outcome of exchanging meaningful data so that coherent interaction among simulation systems takes place, which is relevant to the underlying conceptual frameworks of the organizations who develop the simulations systems. Technical interoperability is the foundation of substantive interoperability. Existing M&S research on interoperability are mainly on technical interoperability. However, the community has realized substantive interoperability is the ultimate goal, e.g., Tolk introduced the Levels of Conceptual Interoperability Model (LCIM) which identified seven levels of interoperability among participating systems and the top level is conceptual interoperability [2].

In this paper, we adopt a model driven approach to formalize and standardize the development process of simulation systems, and propose a MDE-based framework combining SMP2 standard [3] (which is MDE-compliant) and ontology techniques to achieve both technical and substantive interoperability. In the following parts, we firstly analyze three fundamental concepts in simulation model engineering; then the M&S interoperability research overview is given in section 3; how to promote simulation interoperability using MDE techniques is presented in section 4 and we propose a MDE-based simulation system development framework to illustrate model driven simulation interoperability achievement in section 5. Section 6 concludes the paper and proposes future work.

## II. INTEROPERABILITY, REUSABILITY AND COMPOSABILITY

In recent M&S research, a large amount of work claim to promote the interoperability, reusability and composability. These 3 terms appear synonymously quite often but actually referring to different concepts.

### A. Reusability and Interoperability

Reusability is the ability of a simulation model to be reused when the application scenario changes. It's a measurement of applicability in new simulation applications. Interoperability is the ability of different simulation systems working together. Though there are more technical issues to be considered, stronger model reusability can efficiently support simulation interoperation if the application scenario is more or less corresponding. Depending on the application scenario, reusable model can support different level of

interoperability from syntactic level to conceptual level. One thing we should mention is that to support interoperability the reusability of models may be sacrificed for we need to add the middleware interface to the models [4]. Reusability, as the name suggests, means that component simulation models can be reused in different simulation scenarios and applications, often with off-line scenario. Interoperability implies an ability to combine component simulations on distributed computing platforms of different types, often with real-time operation.

#### B. Composability and Interoperability

Composability is the capability to select and assemble simulation components in various combinations into valid simulation systems to satisfy specific user requirements. Essentially, interoperability is the ability to exchange data or services at run-time, whereas composability is the ability to assemble components prior to run-time. Interoperation deals with how a complex system works, how the composing elements work with each other, how they are orchestrated to deliver the required functionality to the user, etc. Composition focuses on what components can be integrated into systems and what functionality can be added without creating problems with other components.

Interoperability deals with the software implementation details of interoperation, including exchange of data elements based on a common data interpretation, which can be mapped to the levels of syntactic and semantic interoperability. Composability addresses the alignment of issues at the modeling level. The underlying models are meaningful abstractions of reality used for the conceptualization implemented by the resulting simulation systems.

In summary, as the terms used in the LCIM, we usually refer to the interoperability of simulations and to the composability of models.

#### C. Reusability and Composability

Composability needs to consider the validity of new application simulation system based on the model/component composition, but the reusability only needs to ensure the equivalence of behavioral logic between the reusable model and the proposed conceptual model to be replaced. Reusability can support composability, and it is the implementing foundation of composability. Composability means reusability, but it's more demanding.

#### D. Summary

Though they are different to some extent and possess their own highlights, these terms reflect an emerging requirement of employing existing resources (models/components/systems etc.) to rapidly develop more powerful systems and efficiently implement more useful functions. Generally speaking, they are in favor of one another and usually the interoperability of simulation systems can be promoted by stronger reusability and composability of simulation models. Interoperability is a general concept which refers to the relationship of simulation systems at run time while reusability and composability are a off-line

characteristics of simulation models and components. Bear their symbiotic relationship and difference in mind, we can better understand how to promote interoperability more efficiently and learn from the fruits of other domains.

### III. M&S INTEROPERABILITY RESEARCH OVERVIEW

#### A. Technical Interoperability

At the technical level, currently the data exchange format (standard) and interface specification of different simulation systems attract most of the researchers' attention. The research on technical interoperability can be concluded as the following categories:

- 1) Propose interoperable and reusable model specification(e.g. SMP2, BOM [5]) to specify the data type and simulation service interface of the simulation model to promote interoperability at the modeling level.

- 2) Use middleware techniques or adapters to link different simulation systems and models from different platforms at the simulation system level. The M&S community has used HLA to interconnect different simulation systems together to achieve results which are not available used separately since 1995 [6].

- 3) Use common standard (e.g. XML/XMI) to specify the data exchange format (standard) and interface specification at the simulation system level [7].

- 4) Use web techniques(e.g. SOA) to wrap simulation communication as a service to enable the interoperability of distributed simulation systems [8].

These four categories are not complete and exhaustive. Some research works are combinations of two or more categories (e.g. HLA/SOA [9]). Up to now interoperability research in the simulation community mainly concentrate on technical interoperability, especially 2-4.

#### B. Substantive Interoperability

Technical interoperability only means that simulation systems can talk to each other and exchange data; but to understand each other correctly and co-simulate effectively requires substantive interoperability. Currently the M&S community is aware of this challenge and initiative works are done to promote substantive interoperability. Hofmann pointed out the necessity of a broadened view on interoperability that includes all three semiotic (syntax, semantics, pragmatics) and conceptual aspects of model development [10]. Some researchers argue that formal techniques should be used to explicitly represent the assumption, constraints and objectives of the M&S systems, e.g., ontology techniques can be used to construct a formal knowledge foundation for simulation models [11][12].

### IV. TOWARDS MODEL DRIVEN SIMULATION INTEROPERABILITY

Model Driven Engineering [13] advocates formal abstractions of both problem space and solution space separately and enables model-based design, analysis and development of software systems. Compared to document centered development, MDE accelerate the development

process and promote the efficiency via automatic model transformation and code generation.

Models also play the vital role in the simulation-based system analysis and design. On one hand, models should be built based on the correct understanding of the SUS and the requirements of the problem to appropriately abstract the SUS; on the other hand, models should be built based on the platform information and implementation considerations to specify computer implementations of the simulation software systems. In other words, in the simulation software system development process, models act as both abstractions of the SUS (descriptive model in the problem domain) and specification of the computer implementation (perspective model in the solution domain). As far as simulation interoperability is concerned, it is essential and feasible to adopt a model driven approach to promote technical and substantive interoperability among models, simulators and simulation systems.

#### A. Model Driven Technical Interoperability

Simulation interoperability at the technical level is closely related to the specification model of the simulation system process (i.e., software implementation of simulation models). So, MDE techniques can be widely used in this domain. We can use model driven approach to develop the software artifacts (like the adaptors) and borrow the tool interoperability ideas from MDE into simulation domain to enable interoperability between different M&S environments. A typical example is proposed in [14], which proposes a model-based integration approach that allows for rapid synthesis of different command and control models in complex HLA-based simulation environments based on GME meta-modeling tool-suite to. Another example is to combine MDA and SOA to promote interoperability [15]. However, According to the LCIM theory, it's very difficult for the technical interoperability reach higher than syntactic level [2].

#### B. Model Driven Substantive Interoperability

Substantive interoperability is more related to abstraction of the SUS. According to the conclusion of the literature review, we propose that the following aspects should be studied to promote substantive interoperability.

1) Use engineering methods to specify subjective factors (including the modeling constraints, assumptions, objectives and application scenarios. Simulation models are approximation of the SUS and no simulation models are absolutely correct but only useful according to the problem requirements. So we need to make the implicit factors underlying the modeling process explicit to let other systems know how simulation systems have been built and how to interoperate.

2) Extract the domain knowledge in which the system under study inhabit to build a reference model (or ontology) to formulate the domain conceptual architecture. Then meaningful interoperation could be achieved on a common knowledge base.

3) Align models of different formalisms and at different abstraction levels to enable the simulation systems. Complex

systems are should be modeled in appropriate formalisms and at appropriate level to reduce accidental complexity.

What's more, organizational and social factors affecting substantive interoperability should also be considered, e.g., [10] presented the efforts in military simulation to promote organizational and social interoperability.

For all the three aspects above, MDE techniques can provide technical supports. Metamodeling and model transformation techniques are very helpful to build knowledge foundations and connect them by transformation. For 1, MDE advocate “modelling everything” principle, which can also provide technical support for modeling subjective factors. For 2, domain specific modelling provides an efficient method to build domain conceptual architecture, which can also be used in simulation conceptual modelling. For 3, it's an effective method to metamodel different formalisms and transforms models of different models according to the mapping of the formalism metamodels.

### V. A MDE-BASED SIMULATION INETROPERABILITY FRAMEWORK COMBINING SMP2 AND ONTOLOGY

To illustrate how model driven simulation interoperability is achieved, we construct MDE-based simulation system development framework in this section combining SMP2 and ontology techniques.

SMP2 is proposed by Europe Space Agency to promote model portability, reuse and interoperability. SMP2 standard comprises a metamodel for simulation models, Simulation Model Definition Language (SMDL), and complies to Model Driven Architecture (MDA, a division of MDE) principles by the usage of Platform Independent Model (PIM) and Platform Specific Model (PSM) in the simulation model development process [3]. SMP2 standard has been successfully used in various application domains as the simulation model specification to develop large complex simulation systems, and it has proven to promote technical interoperability. However, SMP2 lacks of behavioral modeling capability and mainly specify the software implementation of simulation models. So it is difficult to describe the characteristics of multi-domain and multi-subsystem for complex simulation systems and achieve substantive interoperability using SMP2 alone.

Model Integrated Computing (MIC) [16], another division of MDE, provides domain specific modeling languages (DSML) and environments for each application domain, generates executable application systems based on domain specific models and combines domain models and application systems together to support the interoperation between subsystems of various domains. So the combination of DSML and SMP2 can fully exploit the potential of MDE paradigm and lays a sound foundation for complex simulation interoperability.

However, substantive interoperability demands a common foundation for the semantics of simulation models and systems. To achieve substantive interoperability, ontology techniques should also be incorporated into the framework to support semantic description and inference. In Figure 1, we propose a SMP2-based simulation system

development framework to promote both technical and substantive interoperability combining ontology. The foundation of the framework is the combination of SMDL and OWL (Ontology Web Language)[17] metamodels, based on which we build a M&S ontology metamodel. Then domain ontology metamodels specifying the DSMLs for the application domains can be instantiated from the M&S ontology metamodel, which comprises two parts: the first is a linguistic metamodel modeling the abstract syntax of the DSML which is based on the SMDL, and the second is a ontological annotation specifying the static semantics which is based on OWL. Finally the domain models are built using the domain ontology metamodels.

In this framework, all models are technically SMP2-based, which means the interfaces linking models and simulators are based on the same technical implementation, so the technical interoperability is guaranteed; and also they are on a common ontology, which lays the foundation for substantive interoperability.

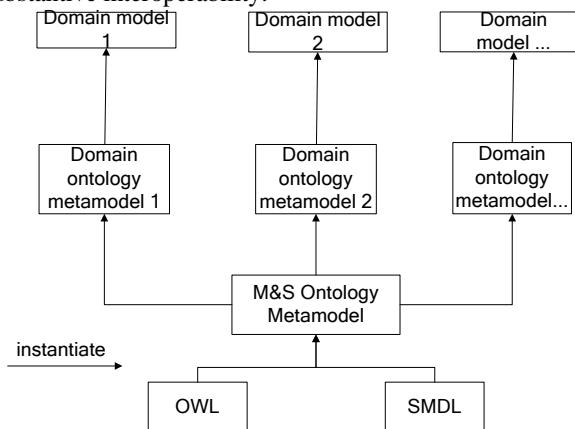


Figure 1. Model driven simulation system development framework

## VI. CONCLUSION

This paper compares the interoperability, reusability and composability and find that usually the interoperability of simulation systems at run-time can be promoted by stronger off-line reusability and composability of simulation models . The overview of current M&S research on simulation interoperability shows that substantive interoperability is of critical importance to meaningful co-simulation but not enough research has been done on substantive interoperability yet. To promote simulation interoperability, MDE techniques are introduced to formalize and standardize the development process of simulation systems. A MDE-based simulation interoperability framework is proposed to show how to use MDE as well as ontology techniques to promote both technical and substantive interoperability.

However, the work is at a primary phase and lots of work needs to be done. The future work will concentrates on the following aspects: the first is to develop a prototype to implement the framework based on the integration of SMP2 modeling tool and OWL modeling environment; the second is to do a more comprehensive overview of current M&S

research on simulation interoperability. The third is perform a series of case studies to verify the framework using the prototype.

## ACKNOWLEDGMENT

The work presented in this paper is partially supported by National Natural Science Foundation of China (No.61273198, No.60974073, No.60974074 and No.71031007).

## REFERENCES

- [1] B. N. Tomlinson, "Simulation interoperability-where are the challenges?," *THE AERONAUTICAL JOURNAL*, no. 3255, pp. 153–160, 2008.
- [2] A. Tolk and J. A. Muguria, "The Levels of Conceptual Interoperability Model," in *Simulation Interoperability Workshop 2003*, 2003.
- [3] E. S. Agency, "SMP 2.0 standard Handbook," 2005.
- [4] C. Chiang, "The use of adapters to support interoperability of components for reusability," *Information and Software Technology*, vol. 45, no. 3, pp. 149–156, Mar. 2003.
- [5] G. Hosseini and A. Shahmirzadei, "Syntactic and Semantic Model Composability based on BOMs," 2007.
- [6] K. L. Morse, M. Lightner, R. Little, B. Lutz, and R. Scrudder, "Enabling simulation interoperability," *IEEE Computer*, vol. 39, no. 1, pp. 115–117, Jan. 2006.
- [7] S. Mittal, B. P. Zeigler, and J. L. Risco-martin, "Implementation of Formal Standard for Interoperability in M&S/Systems of Systems Integration with DEVS/SOA," *International Command and Control, C2 Journal*, vol. 3, pp. 1–27, 2009.
- [8] A. Moreno and J. M. Cruz, "Interoperability between DEVS and non-DEVS models using DEVS/SOA," in *Proceedings of the 2009 Spring Simulation Multiconference*, 2009, pp. 1–9.
- [9] W. WANG, W. YU, Q. LI, W. WANG, and X. LIU, "Service-Oriented High Level Architecture," in *European Simulation Interoperability Workshop*, 2008.
- [10] M. a. Hofmann, "Challenges of Model Interoperation in Military Simulations," *Simulation*, vol. 80, no. 12, pp. 659–667, Dec. 2004.
- [11] G. A. Silver, K. R. Bellipady, J. A. Miller, K. J. Kochut, and W. York, "Supporting Interoperability using the discrete-event modeling ontology (DeMO)," in *Proceedings of the 2009 Winter Simulation Conference*, 2009, pp. 1399–1410.
- [12] L. Yilmaz, "Using meta-level ontology relations to measure conceptual alignment and interoperability of simulation models," in *2007 Winter Simulation Conference*, 2007, pp. 1090–1099.
- [13] D. C. Schmidt, "Model Driven Engineering," *IEEE Computer*, vol. 39, no. 2, pp. 25–31, 2006.
- [14] H. Neema, H. Nine, G. Hemingway, J. Sztipanovits, and G. Karsai, "Rapid Synthesis of Multi-Model Simulations for Computational Experiments in C2," in *AFCEA-GMU Symposium*, 2009.
- [15] R. Jardimgoncalves, A. Grilo, and A. Steigergercao, "Challenging the interoperability between computers in industry with MDA and SOA," *Computers in Industry*, vol. 57, pp. 679–689, Dec. 2006.
- [16] James R. Davis, "Model Integrated Computing: A Framework for Creating Domain Specific Design Environments," in *The 6th World Multiconference on Systems, Cybernetics, and Informatics (SCI)*, 2002.
- [17] D. L. McGuinness and F. van Harmelen, "OWL Web Ontology Language Overview," *World Wide Web Consortium (W3C) recommendation*, 2004. [Online]. Available: [www.w3.org/TR/owl-features](http://www.w3.org/TR/owl-features).