VERIFICATION METHODOLOGY FOR SIMULATION MODEL BASED ON SYSTEM MORPHISM

Changbeom Choi[(a)], Jangwon Bae[(b)], Minwook Yoo[(c)], Tag Gon Kim[(d)] and Soohan KIM

[(a) (c) (d)]Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology
[(b)]Dept. of Industrial and System Engineering, Korea Advanced Institute of Science and Technology
[(e)] Video Display Biz. Technical Planning Group, SAMSUNG Electronics HQ

[(a)]cbchoi@sim.kaist.ac.kr, [(b)]jwbae@sim.kaist.ac.kr, [(c)]mwyu@sim.kaist.ac.kr , [(d)]tkim@ee.kaist.ac.kr,
[(e)] ksoohan@samsung.com

ABSTRACT
As the development of modern combat system grows rapidly, the importance of a development and the verification of the simulation model are also growing rapidly in simulation based acquisition. The simulation model of the combat systems can be defined by integration of simulation models, and simulation of the models is interaction among simulation models. Therefore, the verification for each simulation model and the verification of interaction among simulation models are important. In this paper we propose system morphic verification method to support the verification of simulation model with respect to the requirement specification, and we propose the incremental system morphic verification method to verify the interaction among the simulation models. The verification method based on the system morphism was used in the development of warship simulator to support the researcher of the national defense research institute.

Keywords: Simulation based Acquisition, Combat System Verification, DEVS Formalism, and System Morphism

1. INTRODUCTION

As the development of science and technology grows rapidly, the development of modern combat system is also growing rapidly. The combat system is a composition of subsystems, such as detection, weapon, and command and control systems. Such subsystems can be the composition of various kinds of equipment, such as, the composition of various search and track radars. Therefore, a combination of a combat subsystem and its components may affect the results of the battle. Consequently, the decision maker wants to try various combat systems with various scenarios. However, developing the combat systems and experiments in reality may consume lots of time, effort, and resources. For this reason, modeling and simulations are used to tackle this problem.

Simulations have become a useful part of the mathematical modeling of various natural systems, such as computational physics, chemistry and biology, human systems in economics, psychology, and social science, in order to gain insight into the operation of those systems, or to observe their behavior (Frigg and Hartmann, 2006). In general, the natural systems are composition of subsystems, which can be defined as the composition of various subsystems, recursively. Therefore, the simulation model of the natural systems can be defined by integration of simulation models, and simulation of the models is interaction among simulation models. Accordingly, checking the effectiveness of the new combat subsystem using its simulation model is possible. As a result, domain and M&S experts may develop a simulation model for the combat system and verify the simulation model that the subsystem is working properly while interoperating with other subsystems under the military doctrine.

In this paper, we propose the verification method based on the Modeling & Simulation (M&S) theory. The proposed verification method utilizes the system morphism to show that the implementation of simulation model satisfies the requirement specification. Moreover, as mentioned above, the combat system is divide into several subsystems that interoperates with other subsystems. Therefore, it can be modeled as either a single simulation model or an interoperation of simulation models which are the subsystem of the combat system. Therefore, we propose the incremental system morphic verification method to verify from the standalone simulator to the interoperation of simulators. In addition we introduce the case study of development of warship simulator. During the development of the warship simulator, the system morphic verification was used to verify the standalone simulator while the incremental system morphic verification method was used to verify the interoperation of simulators. In our case study we adapted the discrete event system specification (DEVS) formalism (Zeigler, Kim, and Praehofer, 2000) as a modeling methodology and

ISBN 978-88-903724-3-8

HLA/RTI as an interoperation environment (IEEE Std 1516, 2000).

The rest of this paper is organized as follows: Section 2 represents the background which related to modeling formalism and the theory of simulation model verification. In Section 3, we introduce simulation model verification using system morphism, and incremental system morphic verification methodology. In Section 4, we introduce the initiative case study for the verification of combat systems using the proposed methodology. Finally, we conclude the paper.

2. RELATED WORK

This section briefly explains the background knowledge which is applied to the combat system verification framework.

2.1. DEVS Formalism

The DEVS Formalism is formalism for the discrete event system modeling based on the set theory, and it is one of the M&S theories which are applied in various military simulations (Kim, Sung, Hong, Hong, Choi, Kim, Seo, and Bae, 2011). The DEVS Formalism supports to specify the discrete event models in hierarchical and modular manner. Therefore, the user may model the target system by decomposing large system into smaller components by applying coupling scheme among them. There are two kinds of models in the formalism: Atomic model and Coupled model.

2.2. System Morphism

System algebra is a mathematical tool used to express a real world system in a specific form with an attribute set and its binary relations. In general, an attribute set can be a system set, input/output event set, and time constraint set of a discrete event system of a real world system. Also, binary relations for any two attributes of system algebra describe the behaviors of the real world system.

The system morphism maps the relation from one system algebra to other system algebra with binary relation property preservation (Zeigler, Kim, and Praehofer, 2000). Formal representation of system algebra can be defined by three-tuples as follows:

Definition 1 System Algebra

A system S_A = <I, O, F> is system S_A, together with the following conditions:

I: system attribute set I, that indicate the input event set

O: system attribute set O, that indicate the output event set

F: I → O: binary relation F, that indicate system transfer relation

Definition 2 System Morphism

Let S_A and S_B are systems. Mapping relation ϕ: $S_A \rightarrow S_B$ relates the system structure of S_A to the system structure of S_B. Mapping relation ϕ is the system morphism if and only if satisfies the following system relation preservation condition:

$$\phi = \{\phi_I, \phi_O\}$$

$$\phi_I : S_A.I \rightarrow S_B.I$$

$$\phi_O : S_A.O \rightarrow S_B.O$$

$$\phi_O(S_B.F(\phi_I(i_A))) = S_A.F(i_A) \text{ for } \forall i_A \in S_A.I$$

Figure 1 shows the system morphism between two systems: S_A and S_B. The functions ***g*** and ***k*** are the mapping relation ϕ showing that the *System$_A$* is system morphic to *System$_B$* under the mapping relation ϕ = {***g***, ***k***}.

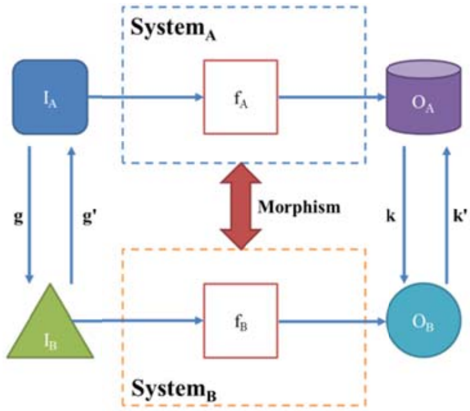

Figure 1: Morphism between two systems

In the M&S theory, the system morphism provides the foundation for simulation model verification. The system morphism assures the system structure preservation among systems. As a result, we can prove that the simulation model is valid and reflects the real system, or we can say that the simulation model is verified and meets the requirement specifications.

3. SYSTEM MORPHISM BASED VERIFICATION METHODOLOGY FOR SIMULATION MODEL VERIFICATION

In practice, implementation of the simulation model can be viewed as a system, where input is the model specification, output is the implementation of the simulation model, and the verification of a simulation model shows that implementation is flawless. This can be proved by the system morphism between the requirement and implementation of the simulation model.

In this chapter, we introduce the verification method using system morphism and incremental system morphic verification which utilizes the system morphism based verification method.

3.1. Verification Method using System Morphism

The verification of simulation model using system morphism defines two systems: requirement specification and implementation of the simulation model. Then it proves the existence of a transition rule that links two systems. As a result, the cooperation

ISBN 978-88-903724-3-8

among domain experts and M&S experts is required for writing requirement specifications of simulation model and checking implementation of simulation model against requirement specification. Figure 2 describes the system morphic relation between requirement and the implementation.

First, the domain experts make a Simulation Logic Description (SLD) document that contains military domain knowledge such as field manuals and technical manuals. The SLD documents provide sufficient information for M&S experts to design and implement the simulation model. In addition, the SLD document usually embraces the UML diagrams (Booch, Rumbaugh, and Jacobson, 1999), such as use-case diagrams, class diagrams, and sequence diagrams to indicate the characteristics and behavior of a target system. When the SLD document is completed, M&S experts design the simulation model from the SLD document, and create the simulation model specification, which can be various modeling formalism.

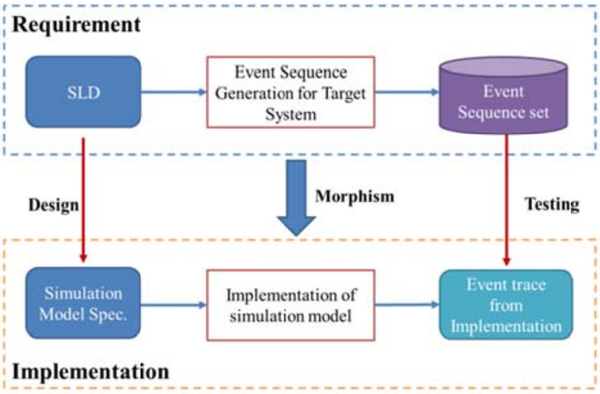

Figure 2: System Morphism between Simulation Model Requirement and Simulation Model Implementation

When the specifications of simulation model are completed, the M&S experts implement the simulation model. In order to verify the implementation of the simulation model, domain experts make the test cases, which are the desired event sequences of the simulation model. During the implementation of the simulation models, the domain experts make the Simulation Model Test Description (SMTD) document, which contains the input value for the simulation model, desired output, and the event trace among the simulation models. In other words, organizing the SMTD document generates an event sequence for the target system. Therefore, the domain and the M&S experts may verify the implementation of the simulation model using SMTD documents by checking input of the simulation model and the corresponding output result, and by monitoring the event traces among the simulation models.

3.2. Incremental System Morphic Verification

In general, adaptation of new technology may have high risk. Therefore, in order to support the decision makers, the simulation models for the target system are necessary.

The systems in the real world are complicated and consist of several sub-systems. For example, a combat system contains several sub-systems such as detection, combat fire control, and weapon systems. Therefore, in order to develop a simulation model for a combat system, the level of detail may vary. In other words, the simulation model of a combat system may contain various simulation models; some may be abstract, and others may be detailed enough to substitute for the real equipment.

As a result, during the simulation based acquisition, the developer may make various simulation models with various levels of detail and utilize the simulation models to simulate the target system. In general, there are two ways to simulate target system, building standalone simulator and organizing the interoperation of simulators. In general, the simulation models of the standalone simulator are usually hard-coded and embedded in the simulator so that the simulation model developer cannot easily modify or extend the simulation models. Moreover, the time required to develop standalone simulation is relatively less than the interoperation of simulators. On the other hand, in the interoperation of simulators, a simulation model can be mapped into a model that participates in the interoperation of simulators. Therefore, the user may extend and modify the simulation model easily, and may substitute an abstract simulation model into a detailed simulation model. Moreover, the simulation model can interoperate with real equipment so that the hardware-in-loop simulation is possible. However, the performance of the interoperation of simulators may be relatively lower and the required development time is longer than the standalone simulator. Figure 3 describes the phase of the simulation based acquisition using the interoperation of simulators. In this simulation based acquisition process, we develop and verify each simulation model in the standalone simulator. Then we extend the standalone simulator into an interoperation of simulators so that domain experts and M&S experts can develop more accurate simulators.

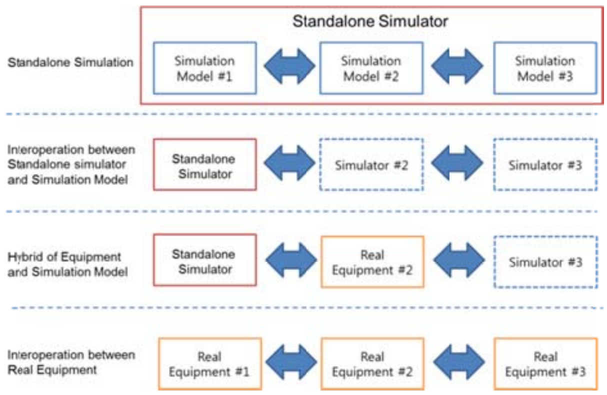

Figure 3: Phase of the simulation based acquisition using Interoperation of simulators

First above all, the M&S experts and the domain expert develop the simulation models and integrate them into standalone simulator. The standalone

ISBN 978-88-903724-3-8

simulator can alter the parameters of the simulation models, and build a simulator by compositing various simulation models. In this phase the M&S experts and the domain experts can use the system morphism based verification method to verify each simulation model in the standalone simulator.

After building standalone simulator, the M&S experts and the domain experts should implement the interoperation features to support simulation between standalone simulators and other simulators. Note that in the interoperation of simulators, the simulator may be the detailed simulation model which reflects the real equipment better than the simulation models of the standalone simulator. Since the simulation models in the standalone simulators are abstract models of the real equipment, by implementing simulators for each simulation models can increase the correctness and reliabilities of the simulation results in distributed simulation environment. At the final phase, the domain expert and the M&S expert can test the real equipment with the simulation models.

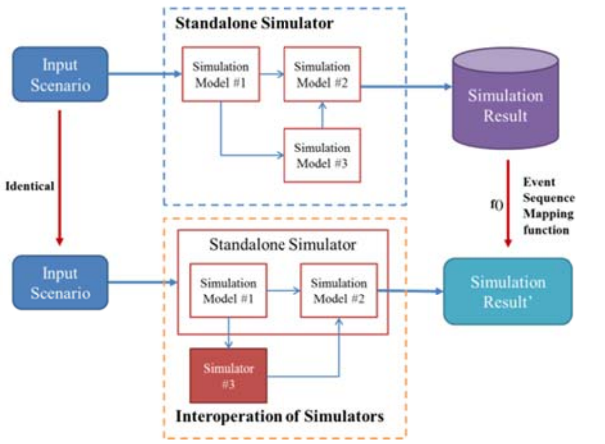

Figure 4: System Morphism between standalone simulator and the interoperation of Simulators

The figure 4 denotes the system morphism between standalone simulator and the interoperation of simulators. In order to give the basis of the verification, the domain experts and the M&S experts utilize the system morphic verification method to show that the standalone simulator and the requirements are in the system morphic relation. Then, the domain experts and the M&S experts utilize the input scenario of standalone simulator and its simulation results to verify that the interoperation of simulator satisfies the requirement specification. In other words, during the simulation based acquisition phases, the domain experts and the M&S experts find system morphism among each phase incrementally.

4. CASE STUDY: VERIFICATION OF SIMULATAION MODEL FOR WARSHIP SIMULATOR

In this section, we introduce the background and verification results of the simulation model of a warship simulator. However, this case study is related to the national defense of South Korea. Ergo, the name of the institute, the modeling results, and the simulation results are classified at the CONFIDENTIAL level. Therefore, we give the initiative information of the simulation models in this case study.

4.1. Background

The main objective of this case study is to develop a framework for the combat system of a warship, which allows engagement among the combat systems. Based on a survey of the Ship Air Defense Model of BAE Systems, the domain experts and the M&S experts have designed the M&S framework for the combat system. The characteristics of this framework are that it supports the simulation model to have various levels of details by implementing Plug and Play feature, supporting interoperation of the simulators, and providing human-in-the-loop simulation and hardware-in-the-loop simulation.

The simulation framework has two modes standalone mode and interoperable mode. The standalone mode contains a simple model of the combat system. Therefore, the user can easily check the trend of the engagement simulation of several combat systems. However, if a user wants to know the results of simulation without using an abstract simulation model, he or she can utilize the detailed simulator, such as MATLAB/Simulink models, by using HLA/RTI. Moreover, the simulation framework supports both human-in-the-loop simulation and hardware-in-the-loop simulation by using HLA/RTI.

In order to implement the warship simulator, we adopted the DEVS graph notation. Using a DEVS graph, Atomic model and Coupled model, the DEVS formalism can be expressed using various symbols. Figure 5 and Figure 6 denotes the DEVS Graph for the Atomic model and the Coupled Model.

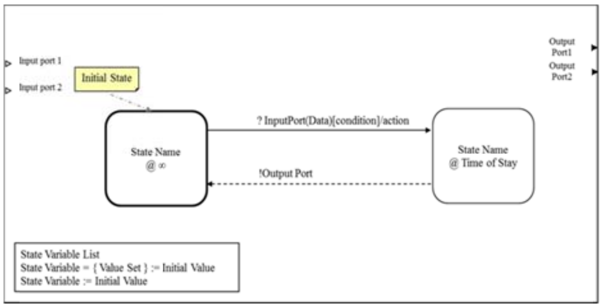

Figure 5: DEVS Graph for Atomic Model

ISBN 978-88-903724-3-8

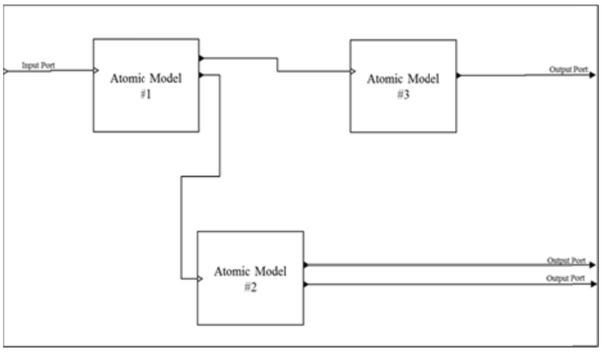

Figure 6: DEVS Graph for Coupled Model

4.2. Implementation

In order to develop the simulation model, we utilize two libraries to support the military experts and the M&S experts: the DEVSim++ and the KHLAAdaptor Library.

The DEVSim++ is a DEVS simulation environment based on C++ (Kim, 2007), and it provides several object-oriented features such as encapsulation, inheritance, and reusability. Moreover, it supports the distributed simulation environment by using HLA/RTI (Sung, Hong, and Kim, T.G., 2005; Kim, Hong, and Kim, 2006). The M&S experts have several advantages in simulation model verification when they use DEVSim++. First, if the simulation model requirements are described in DEVS Formalism, the M&S experts can implement the simulation model, which is atomic and coupled model directly. In other words, the simulation algorithm of DEVS formalism is implemented in DEVSim++, so that if the user implements the simulation model, the DEVSim++ manages the time scheduling and data handling. Therefore, the verification of simulation model using DEVSim++ can be viewed as monitoring the event sequence of the simulation model. Second, the DEVSim++ can cooperate with HLA/RTI, and helps the user to focus on the design and implementation of the simulation model rather than on the interoperation among simulators. Finally, the M&S experts can reuse the simulation model in an object-oriented fashion.

The KHLAAdaptor library enables developers to build an interoperation of simulators using HLA/RTI (Kim, Sung, Hong, Hong, Choi, Kim, Seo, and Bae, 2011). The KHLAAdaptor supports the developers to select the HLA services which will be used in the interoperation of simulators. The advantages using KHLAAdaptor library are twofold. First, the KHLAAdaptor library handles the time management and the data management, so that the developer may consider the encoding and decoding of the simulation message. As a result, the developers can focus on the design and implementation of the simulation model. Second, the KHLAAdaptor library handles the entire simulation messages of the simulation application during interoperation simulation. Therefore, the tester may collect the event sequences from the KHLAAdaptor to verify the interoperation of simulators.

4.3. Verification Result

The verification process for the simulation models of a warship combat system comprises of four phases as shown in the Figure 7. During Phase I, the military and M&S experts verify that the behavior of the simulation model based on the system morphism. In this phase, the domain experts make SLD document in natural language. Afterward, the domain experts extract the SMTD documents from the SLD; on the other hand, the M&S experts make Simulation Model Specification from SLD. In this case study, we choose the DEVS formalism as a simulation model specification to model the combat system and have drawn every DEVS graph in the warship simulator.

In order to verify the combat systems in the warship simulator, we have collaborated with the researchers of the national defense research institute. Figure 7 depicts the DEVS graph of the missile launcher. The M&S experts have developed the DEVS models based on the SLD document; both military and M&S experts inspect the DEVS graph against the SLD document. Figure 8 illustrates the DEVS graph for the missile launcher.

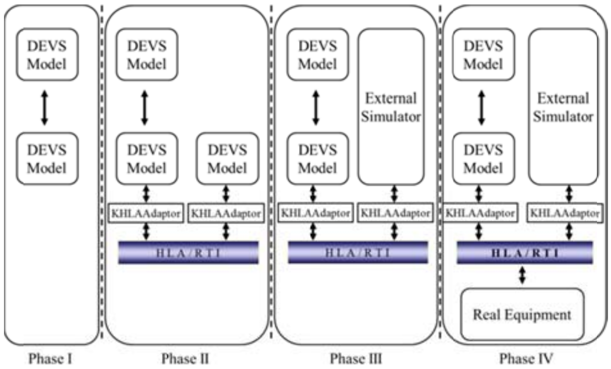

Figure 7: Verification process of simulation models of warship combat systems

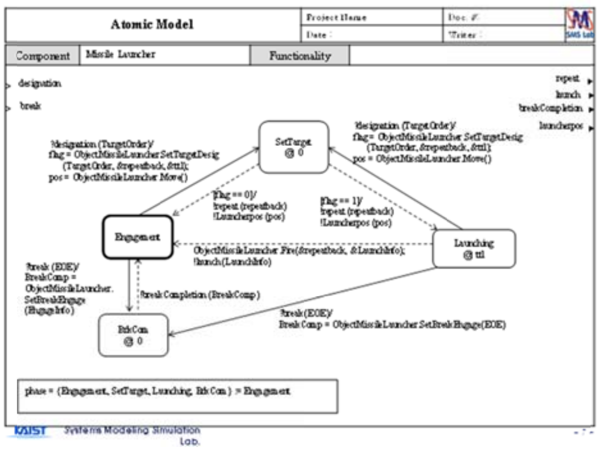

Figure 8: DEVS Graph for the Missile Launcher

Table 1: Air-to-Surface Missile Verification

Step	Action	Desired Output	Passed/ Failed
1	Model Initialized (set target combat	Check the initial parameters	Passed

ISBN 978-88-903724-3-8

	system)	though GUI	
2	Model Generation	Missile Launcher model launches the ASM	Passed
3	ASM follows the target combat system	Check the trajectory of the ASM	Passed
4	Model Destroy	Check through GUI	Passed

After drawing each DEVS graphs in the warship simulator, the M&S experts begin implementing the simulation models; the domain experts must extract test cases from the SLD documents and make the SMTD documents. Table 1 shows this part of the SMTD document.

In Phase II, we verify the design and the implementation of HLA modules, i.e. KHLAAdaptor. In order to verify the HLA modules, we use the DEVS models to simulate the detailed simulator. During this phase, we can verify the time synchronization, i.e. verify the modules that use HLA services, and data exchange, such as data encoding and decoding. After this phase, we can guarantee that the HLA modules are stable enough to test the interoperation between standalone simulator and detailed simulators.

In Phase III, we simply exchange the DEVS model into the External Simulator, i.e. the detailed simulation model for the real equipment. After we verify the external simulator by reviewing the source code and checking the log files, we can finally interoperate the real equipment.

5. CONCLUSION

This paper introduces a verification method based on the system morphism. System morphism is the relation between systems that shows the existence of a structure preservation function between two systems. The verification of the simulation model against the requirement specification is verifying the structure of the system. In other words, if we assume that designing and testing are the ideal structure preservation functions, the verification of the simulation model is that checking the test cases which are generated based on the requirement specification. We propose a system morphic verification method to support the verification of a simulation model with respect to the requirement specification, and we propose the incremental system morphic verification method to verify the interaction among the simulation models. The verification method based on the system morphism was used in the development of the warship simulator in order to support the researcher of the national defense research institute. Regarding future research, we will extend the incremental system morphic verification method to verify the various simulation models, such as discrete the time model and continuous model.

ACKNOWLEDGMENTS

This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2009

REFERENCES

Booch, G., Rumbaugh, J., and Jacobson, I., 1999. *The Unified Modeling Language User Guide*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.

Frigg, R. and Hartmann S., 2006. *Models in Science.* Stanford Encyclopedia of Philosophy, Available from: http://plato.stanford.edu/entries/models-science/

IEEE Std 1516, 2000. *IEEE standard for modeling and simulation (M&S) High level architecture (HLA) - framework and rules,* doi: 10.1109/IEEESTD.2000.92296,

Sung, C.H., Hong S.Y. and Kim, T.G., 2005. Layered Structure to Development of OO War Game Models Using DEVS Framework. *Proceedings of Summer Computer Simulation Conference,* pp. 65-70. July 24-28, Philadelphia (Pennsylvania, USA).

Kim, J.H., Hong, S.Y., and Kim. T.G., 2006. Design and Implementation of Simulators Interoperation Layer for DEVS Simulator., *Proceedings of Summer Computer Simulation Conference,* pp. 195 – 199, July, Calgary, Canada.

Kim, T.G., 2007. *DEVSimHLA User's Manual.* Available from: http://smslab.kaist.ac.kr

Kim, T.G., Sung, C.H., Hong, S.Y., Hong, J.H., Choi, C.B., Kim, J.H., Seo, K.M., and Bae, J.W., 2011. DEVSim++ Toolset for Defense Modeling and Simulation and Interoperation, *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 8(3), 129–142.

Zeigler, B.P., Kim, T.G., and Praehofer, H., 2000. *Theory of Modeling and Simulation*. Orlando, FL, USA: Academic Press, Inc.

ISBN 978-88-903724-3-8