TEST CASE BASE FOR DYNAMIC VERIFICATION OF DEFENSE SIMULATION MODEL BASED ON SYSTEM ENTITY STRUCTURE AND MODEL BASE

Jaewoong Choi[(a)], Jiyong Yang[(b)], Inho Jung[(c)], and Changbeom Choi[(d)]

[(a),(b),(c)]School of Information and Communication Engineering, Handong Global University,
Handong-ro 558, Pohand-si, Kyungbuk, Rep.of Korea
[(d)]School of Global Entrepreneurship and ICT, Handong Global University,
Handong-ro 558, Pohand-si, Kyungbuk, Rep.of Korea

[(a)]21631005@handong.edu, [(b)]21631004@handong.edu, [(c)]21731002@handong.edu
[(d)]cbchoi@handong.edu

ABSTRACT
The development process of the defense simulation consists of developing simulators and models for purpose of simulation. To increase the efficiency of the existing defense simulation development process, it is possible to apply SES / MB to defense simulation. When SES / MB is applied to the development process of defense simulation, defense simulation can be developed by importing and synthesizing models corresponding to nodes of SES from MB. In this paper, we propose Test case Base for dynamic verification of defense simulation model based on SES / MB. The proposed TB can be used for dynamic verification of defense simulation based on SES / MB. As a case study, we verify dynamic verification of simulation models modeled in the AddSIM environment being used for modeling in the National Defense Science Institute of Korea.

Keywords: Dynamic Verification, Defense Simulation, SES / MB, Test case Base

1. INTRODUCTION

Defense simulation, also known as a war game, is a simulation for testing and improving the war theory without actual war. Defense simulation is widely used for training, analysis, and acquisition. In addition to the defense simulation that is used for various purposes, the general simulation development process consists of two processes: determining the purpose of the simulation and developing a model that fits the purpose.

The two steps of the simulation development process, establishing the simulation purpose and modeling process, have significant meaning in the M&S. Establishment of the simulation purpose involves setting the target system of the simulation and determining the simulation purpose for the target system. First, since modeling does not represent the entire system but abstracts the system according to the purpose of simulation, it is critical to establish the purpose of simulation. Second, since the resolution of the models that are configured according to the purpose of the simulation is determined, and the model has changed accordingly, it is important to establish the purpose accurately. Third, because the target system should be modeled exactly to express the characteristics used in M&S, modeling process is important.

In conventional defense simulation, developers had to re-develop and verify simulators and models that depend on the target system and simulation purpose. Although there are efforts to improve reusability such as using reusable parts of the developed model in the process of development, it does not have a significant influence on the increase of the reusability of the model.

The System Entity Structure (SES) formalism and the Model Base (MB) concept can be applied to improve the reusability of the model. SES formalism is a formalism that expresses all the alternatives of one system using tree structure, and it includes three kinds of information such as system configuration, classification, and connection relation (Kim, 1990). The MB is a database for managing models constituting the system represented by the SES formalism. By using SES / MB appropriately, a defense simulation, which is developing the model by designing a model that is suitable for the simulation purpose, pruning the system represented by the SES formalism, and combining the sub-models taken from MB, can be constructed. Details of SES / MB are described in Chapter 2.

As described briefly above, applying the SES / MB to the defense simulation makes it possible to build a model management system that improves the reusability of the model. In this paper, we propose Test case Base for dynamic verification of defense simulation model. TB manages the test cases saved in the MB.

This paper is composed as follows. Section 2 explains SES / MB, which is a background for understanding this paper. Section 3 describes the TB. Section 4 describes the case study of dynamic verification of defense simulation using the proposed TB and SES / MB and concludes in Section 5.

2. SYSTEM ENTITY STRUCTURE FORMALISM AND MODEL BASE

The SES formalism is that expresses the target system in the form of a total solution including all the alternatives of the target system using the tree structure. The SES formalism includes three pieces of information: configuration relation, classification relation, and connection relation of the target system. Configuration relation indicates that the system is composed of any model, classification relation indicates that the system is a model may have any Alternatively, the connection relation is expressed with respect to the connection relation of the model.

The SES formalism has three nodes to represent this information. First, entity node refers to models that make up the system and, aspect indicates whether the node is drawn is one of the model consists of any sub-model line was solid in the tree. A specialization node is a node that indicates which alternatives a model can have, drawn in solid double-dashed lines in the tree. The multiple aspect node, which is a type of aspect node, draws three solid lines in the tree, representing the subordinate models constituting a single model, all of which are the same kind of special aspect nodes.

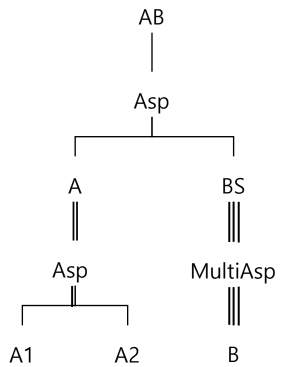


Figure 1 SES example

Above Figure 1 is a simple example to illustrate SES formalism. In the figure above, AB, A, BS, A1, and B correspond to entity nodes, and Asp and Spec correspond to aspect nodes and specialization nodes, respectively. MultiAsp is a multiple aspect node. In Figure 1, the system AB is represented by AES and BS. A has an alternative of A1 or A2. The BS consists of several B models.

The target system expressed through the SES formalism represents all the alternatives that the target system can have. The pruning process is required to re-express the model in a suitable model structure after selecting the appropriate nodes according to the simulation purpose. The result is a Pruned Entity Structure (PES).

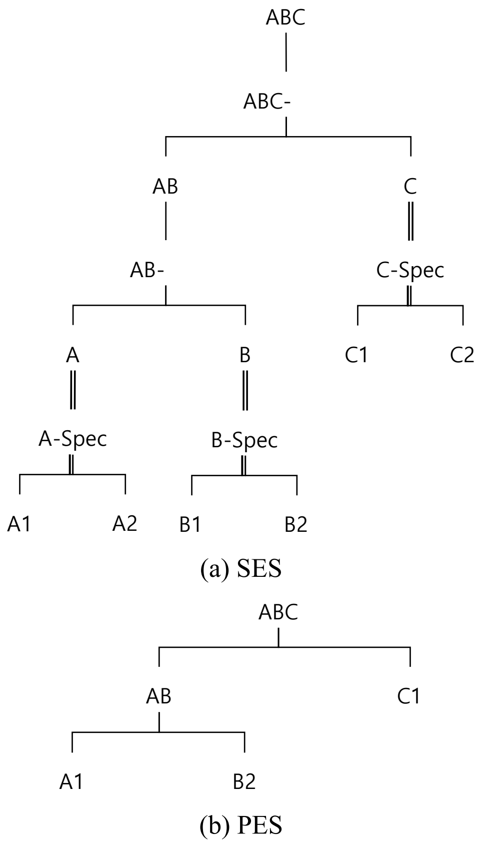


Figure 2 SES pruning example

For example, if an arbitrary user conducts pruning to implement a system expressed as SES in Figure 2 (a), the PES in Figure 2 (b) is the result that selects A, B, and C as A1, B2, and C1, respectively. The PES can be generated variously according to the pruning algorithm or according to the user's selection in the pruning process.

MB is a database that manages the verified models through unit testing, and user can implement the PES as a real simulation model by taking each model constituting the PES from the MB and synthesizing them (Park, 1997; Zeigler, 1991).

3. TEST CASE BASE FOR DYNAMIC VERIFICATION OF DEFENSE SIMULATION MODEL BASED ON SYSTEM ENTITY STRUCTURE AND MODEL BASE

3.1. Test case Base

The TB proposed in this paper is a database that manages test cases for the dynamic verification of each model, which is corresponding to each node constituting the SES of defense simulation. TB manages test cases of models existing in MB, and test cases can be written in various formats. The user can use the test cases existing in TB in some ways, and can add/delete/edit test cases. However, to increase the reliability of TB, only the user who has privileges must be able to add/delete/edit test cases.

ISBN 978-88-97999-90-4; Bruzzone, Cayirci and Sottilare Eds.

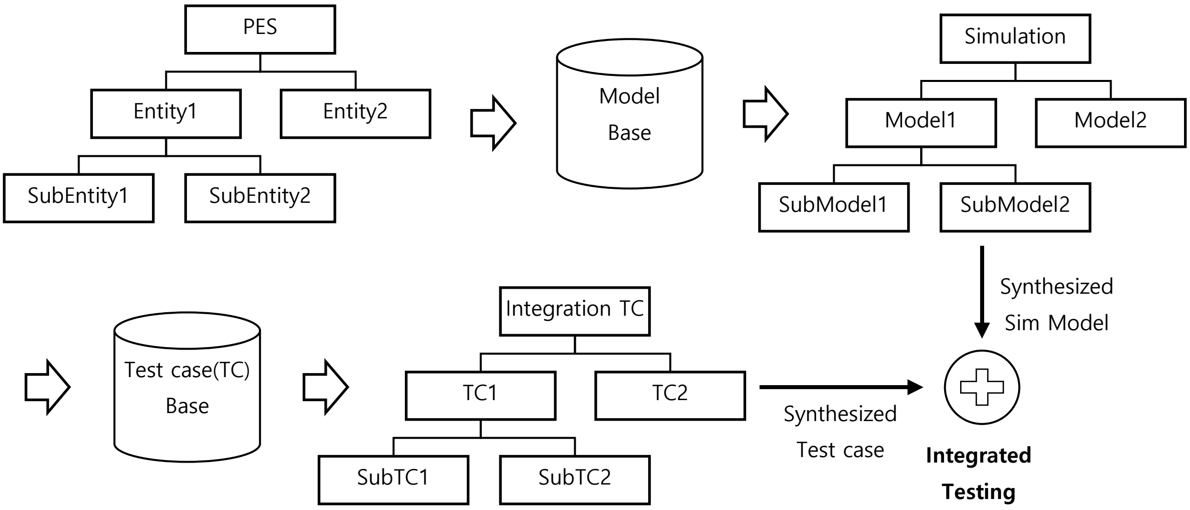


Figure 3 Blueprint for dynamic verification method

3.2. Test case Base

The dynamic verification of the defense simulation model consists of two parts: unit testing of the sub-models that make up the defense simulation model and integration testing of the synthesized model. Unit testing is necessary to ensure the reliability of the sub-models. Even if the defense simulation model is composed of verified sub-models, integration testing of the synthesized simulation model is required due to various errors that may occur during the synthesis process of the simulation model. (Gao, 2003)

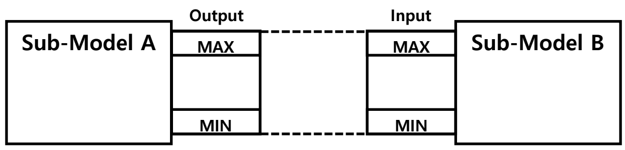
As mentioned above, the target of the dynamic verification method using TB is the defense simulation model developed by using the defense simulation model management system using SES / MB. Therefore, it means that the sub-models constituting the developed defense simulation model are the models taken from the MB, and the MB is the database managing the verified models through the unit test, so unit testing for the sub-models do not need to be performed separately. Therefore, dynamic verification of defense simulation model using TB is integration testing of simulation model composed of verified sub-models.

Figure 3 shows the dynamic verification process using the TB of the simulation model developed using the defense simulation described in the SES formalism. The process of development and dynamic verification of simulation is as follows. The user takes the models corresponding to the entities of the PES generated through appropriate pruning process from the model base and synthesizes them to create a simulation model suitable for the simulation purpose. Then, we take the test cases corresponding to each model from TB and synthesize them or add user-defined test cases to create integration test cases. And we apply the integration test case to the synthesized simulation model to perform dynamic verification of the synthesized simulation model.

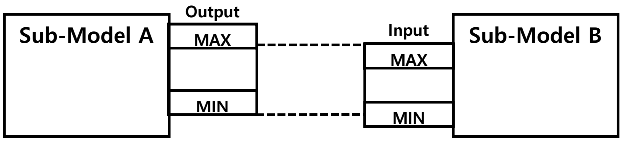
3.3. Use of Testcase Base

The defense simulation represented using SES is hard to describe the range of values of input and output of sub-models due to the limitation of representation of SES. Therefore, it was difficult to judge whether synthesis is possible between certain sub-models. For example, if two models are model A that outputs a value between 1 and 5 and model B that operates with input value between 6 and 10, the model A and B cannot be combined without a separate process of converting values. However, since the model A and the model B represented using the SES do not describe information about the input and the output range, the user cannot know that the model A and the model B can not be combined with each other. However, by using TB, we can reverse-trace the range of the output value of the model A and the range of the input value of the model B through the test case of the model A and the model B existing in the TB. Thus, TB can be used as a way of diagnosing the minimum possible combination between any sub-models.

In the process of importing and combining the test cases corresponding to each model from the TB, two cases occur depending on the attributes of the sub-models to be integrated. Figure 4 shows two cases that occur according to the input/output range of sub-models A and B when we try to make an integration test case for the integrated model AB by synthesizing the test cases of sub-models A and B.



(a) IO range matching case



(b) IO range not matching case

Figure 4 Two case of integrating sub-models

Figure 4 (a) shows the case where the output of sub-model A matches the range of input of sub-model B. Suppose that one of the test cases for sub-model A, $A\_Testcase_n$, exists and its input and output are $input_n$ and $output_n$, respectively. At this time, $A\_Testcase_n$ can be expressed as (A_input_n / A_output_n). The output of $A\_Testcase_n$, $output_n$, is the input of B, and the resulting output is AB_output_n. At this time, $AB\_Testcase_n$, which is an integration testcase for the integrated model AB, can be defined as (A_input_n / AB_output_n). Figure 5 is simplifying of above description.

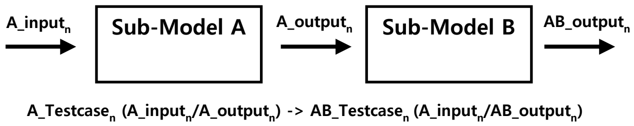


Figure 5 Integration test case generation in the case of IO matching case

Thus, in case of IO matching case, various integration test cases can be created by using existing test cases. Also, if the user determines that the generated testcase $AB\_Testcase_n$ has a meaningful result, it can be registered in the TB and can be recommended as a testcase for the future integrated model AB.

Figure 4 (b) shows the case where the output of sub-model A and the input of sub-model B do not agree with each other. In this case, integration test cannot be performed using existing testcase stored in TB, so user must add user-defined integration test case directly.

4. CASE STUDY: DYNAMIC VERIFICATION O F SIMULATION MODEL MODELED IN ADD SIM ENVIRONMENT

As the case study of proposed dynamic verification method, dynamic verification of defense simulation model developed in AddSIM2.0 environment which is used at Korea Defense Science Institute proceeds.

The target model of this case study consists of 8 components in total and consists of aircraft component, detection radar component, launcher component, and four missile components. Simulation scenario is as follows: when the enemy aircraft is within the range of the allied detection radar, the allied detection radar detects it and transmits the coordinates to the launcher, and the allied launcher fires the missile to shoot down the enemy aircraft. In this case study, an integrated model combining the launcher model and the four missile models in the simulation model was verified.

The launcher model receives the location information and speed of the enemy aircraft as inputs and transmits the information received as input to the available missile model. And the missile model calculates and shoots the enemy aircraft based on the information of the enemy aircraft. The input and output formats of the launcher model and the missile model used in the case study are as follows.

- Launcher_input: (X, Y, Z, Vd)
  Launcher_output: (Mn, X, Y, Z, Vd)
- Missile input: (Mn, X, Y, Z, Vd)
  Missile output: ifSuccess

The output format of the launcher and the input format of the missile, Mn, are the id of the missile to which the enemy airplane is being hit. In this scenario, a total of four missile models are assigned, so Mn is a value between 1 and 4. Also, ifSuccess, the output format of the missile, compares the coordinate value of the enemy aircraft with the target coordinate value of the missile and applies the probability to indicate whether the enemy aircraft is shot down as True / False.

The target model of the case study belongs to the IO matching case described above because the output of the launcher and the input range of the missile are matched. The integration testcase is created as many times as the number of test cases for the launcher model as described in the previous chapter. For each generated integration testcase, an experiment that entering the input of the launcher according to the generated integration test case, recording the output of the missile using the probe code, and compare the recorded output of missile with the output of generated integration test case was performed n times.

As the value of n increases, the dynamic verification result of the target model for each integration test case converged to a similar value of the collision probability value specified by itself in the missile in the error rate of about 3%. Therefore, the integration testcase generated based on the existing testcase stored in TB is valid, and it means dynamic verification can be performed using TB.

5. CONCLUSION

In this paper, we propose test case base for dynamic verification of defense simulation model based on system entity structure and model base. The proposed TB can increase the reusability of the test cases of models by efficiently managing the test cases of the models developed for defense simulation. Also, the user can create various values by reusing test cases managed in TB. Specifically, we can perform dynamic verification by generating an integration test case for a specific case among synthesized simulation models by combining test cases of existing models managed through TB. Also, TB can be the basis for backtracking the attributes of the model that cannot be described due to the limitation of SES, and it is possible to determine whether synthesis can be performed between specific models using the properties of the backward traced model.

In this paper, only the verification of the case of implementing the defense simulation by synthesizing the model stored in the MB is performed. However, the test case stored in the TB can be usefully used even when the stored model is modified. For example, if the user has a model A stored in the MB and user want to verify it by defining a model A, which is similar to A, user can get a hint from A's existing testcase stored in TB. This will be discussed in future research.

ACKNOWLEDGMENTS
This work was supported by BSM-based verification and user code verification module(UD160075BD) funded by the Agency for Defense Development.

REFERENCES

T. G. KIM, C. LEE, E. R. CHRISTENSEN and B. P. Z EIGLER, "System Entity Structuring and Model B ase Management," IEEE Transactions on Systems, Man and Cybernetics, vol. 20, no. 3, pp. 1013-102 4, Oct., 1990.

Park, Hyu Chan, Wan Bok Lee, and Tag Gon Kim. "RA SES: A database supported framework for structur ed model base management." Simulation Practice and Theory 5.4 (1997): 289-313.

Zeigler, Bernard P., Cheng-Jye Luh, and Tag-Gon Kim. "Model base management for multifacetted syste ms." ACM Transactions on Modeling and Comput er Simulation (TOMACS) 1.3 (1991): 195-218.

Gao, Jerry, H-SJ Tsao, and Ye Wu. Testing and quality assurance for component-based software. Artech House, 2003.