Engineering Semantic Composability Based on Ontological Metamodeling

Zhi Zhu[1(✉)], Ning Zhu[2], Yongling Lei[1], Qun Li[1], and Huabing Wang[2]

[1] National University of Defense Technology, Changsha 410073, China
zhuzhi@nudt.edu.cn

[2] State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System, Luoyang 471003, China

Abstract. Combat system modeling generally includes two aspects: structure and behavior. Structural modeling is to build the structure of entities and their internal and external static relationships. Behavioral modeling aims at the typical working processes. The simulation modeling definition language (SMDL) of Simulation Modeling Platform (SMP2) can describe system structure well, but it is difficult to support behavioral modeling. The modeling and simulation tool named Ptolemy is able to support the description of system behavior, but it lacks effective support for structural modeling. For this reason, the goal of the paper is to explore a mechanism to combine structural and behavioral modeling, which can not only effectively support the architecture design but also support the expression of system behaviors. As a proof of concept, we extend the SMDL by adding the synchronous data flow (SDF) elements, and use this new language, namely, the extended Simulation Model Definition Language (ESMDL), to specify a radar system for both of structural and behavioral modeling.

Keywords: Metamodeling · Semantic composability · SMP2

1 Introduction

Combat system presents the characteristics of large-scale and multiple subsystems. It have reached a complexity that requires inevitably the combined use of different formalisms to ensure correct domain descriptions and good model specifications [1]. Traditionally, pure theoretical analysis and experimental observation are not able to answer these complex questions because it is very difficult to observe the real operation process of an actual system, or the cost may be too expensive and restricted by various external factors [2]. Therefore, many researchers have paid attentions to simulation as an effective alternative method. Using the advantages that simulation brings, system experiment and analysis can acquire many benefits that theoretical analysis and experimental observation have not. In addition, in the face of new system or novel functional requirements, it is a real challenge that how to standardize the representation of simulation model, with the objective of maximizing the reuse of existing model and simulation resources, to achieve rapid and high-quality development of simulation applications.

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 369, pp. 582–592, 2021.
https://doi.org/10.1007/978-3-030-72792-5_46

Specifically, along the development of simulation application, this problem needs to address the following three aspects, which are also the long-term attention in the modeling and simulation community. Firstly, simulation model lacks the unified reusable model specification. Consequently, as a valuable knowledge-intensive asset, simulation model lacks reusability, which makes it difficult to support the rapid and high-quality development of simulation applications. Secondly, combat system modeling includes structure and behavior in general. The structural domain integrate behaviors tightly, which will lead to many issues such as unnecessary repetitive processes and a set of modeling and simulation resources of waste. Thirdly, a simulation model is traditionally viewed as an ordinary software and it lacks the support of a standardized simulation modeling language which can describe its structure and behavior simultaneously.

For above problems, the goal of the paper is to design a new simulation modeling language called ESMDL (Extended Simulation Model Definition Language). This language is established by extending the SMP2, which can support typical behavioral modeling paradigm like synchronous data stream (SDF). Furtherly, the simulation scheduling operation mechanism is designed to seek a simulation modeling method with high efficiency, strong reusability and easy expansion, which can effectively support the radar system construction in various environments. Finally, taking a certain radar system as an example, this paper uses ESMDL to model and simulate the radar target scene, echo simulation, signal processing, data processing, display and evaluation subsystem, and verifies the method proposed in the paper.

2 Related Works

Since 1960s, the research on radar system simulation modeling technology has been started and a large number of excellent simulation model specifications, modeling paradigms and simulation protocols have been formed, as well as a series of radar simulation models. However, the theory and method of simulation modeling have not yet formed a unified standard and specification, nor a general radar simulation application system [3]. Therefore, it is difficult to reuse the radar simulation model, and the combination and interoperability is not strong. Aiming at the problems of high level architecture (HLA) [4] such as complex architecture, difficult model granularity control and system maintenance, SISO (Simulation Interoperability Standards Organization) proposed the base object model (BOM). In the early 1990s, ESA (European Space Agent) recognized the problems of simulation reuse and model transplantation in different development stages and projects during the development of many spacecraft. In the mid-1990s, ESA initiated the formulation plan of simulation model portability standards (SMP) [5], with the goal of realizing the exchange of models among the simulation platforms.

Some research work has also done a lot of research work in radar system simulation modeling, and achieved many outstanding results. In the early 1990s, the National University of Defense Technology has begun to organize forces to develop distributed simulation support software. In 1996, DIS-Link, the first distributed simulation support software in China that met the DIS standard, solved the time synchronization and management problems of large-scale DIS, and surpassed VR link in terms of function and performance. In 1999, KD-RTI software, the first HLA compliant software in China, was developed. A system method for large-scale combat simulation system integration based on shared resource library and general simulation environment was proposed. The HLA simulation environment and platform software have been successfully used in hundreds of distributed simulation systems inside and outside the army.

In fact, the above-mentioned simulation modeling technologies fall into standards and specifications based and specific domain oriented. The former focuses on the syntactic heterogeneity of simulation models, while the latter focuses on the differences in semantic expression [6]. These two types of simulation modeling methods complement with each other, and build reusable simulation model library from syntax and semantic levels respectively to achieve simulation resource reuse, save simulation development cost and improve efficiency of simulation application development. The first mock exam is to attempt to unify the formal representation and formal definition of models from the technical or grammatical level, such as unified model specification and simulation protocol like HLA and SMP. Such as unified modeling methods and platforms like discrete event system specification (DEVS) [7, 8]. The second type is from domain or semantics. For example, the application system based on specific domain, including extended air defense system (EADSIM) [9], system effectiveness analysis simulation of air force (SEAS) [10], etc.

The above two kinds of simulation modeling technologies solve the problem of combination reuse of simulation models to a great extent from the perspective of standard specification and specific domain oriented, and have achieved good results, which also lay a solid foundation for current and future modeling and simulation research [11]. However, the following problems still exist in the field of radar simulation modeling. Firstly, most of the above simulation modeling standards and simulation systems are commercial, and their application fields are relatively sensitive. There is no effective model sharing mechanism among various units, and there is not enough support for new requirements in the new system radar and the expansion and development of new models. Secondly, most simulation systems have adopted relatively high performance. However, the radar models in these simulation systems either pay attention to the radar composition structure and external static interface relationship, or pay attention to the single representation of radar behavior. There is no effective and flexible integration mechanism between the two aspects. Thirdly, many radar simulation systems support a single electronic warfare style and lack of rapid combination of multiple scenarios in complex system combat environment. A set of integrated simulation software tools, including scenario editing, model development, experiment design, decision modeling, two-dimensional and three-dimensional situation display are not able to accurately and effectively evaluate the performance of radar in complex combat system [12].

3 Engineering Semantic Composability of Simulation Models

3.1 The Proposed Framework

The overall technical scheme of the paper is based on the system modeling and simulation platform, integrating ESMDL simulation operation scheduling mechanism, ESMDL simulation modeling language, and system application, as shown in Fig. 1.

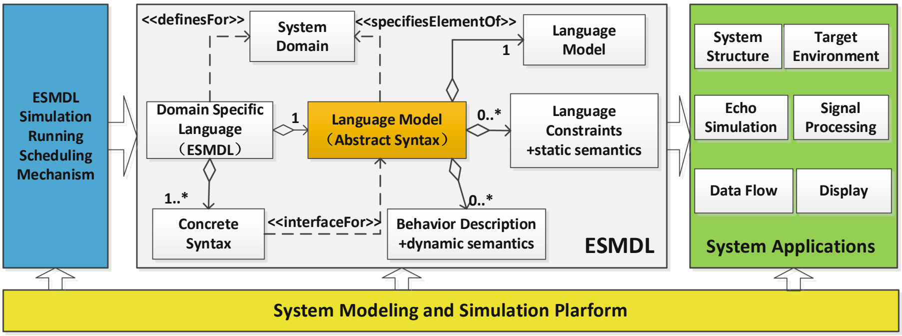

Fig. 1. The proposed framework.

Firstly, system modeling and simulation platform provides the operation environment and related tools for radar system simulation, including the whole process of equipment design and demonstration from the integrated development of simulation model, data preparation, scenario editing, experimental design, etc.

Secondly, the core of the project is the design of ESMDL. It is combination of SDF and SMP2. Like the new domain specific language (DSL), it mainly involves the definition of syntax and semantics [13]. The core language model, language model constraint and behavior description are aggregated together to form the core of ESMDL, namely language model. In addition, the concrete syntax is separated and is used to create a friendly human-computer interaction form of ESMDL.

Thirdly, the research of ESMDL simulation running scheduling mechanism supports the simulation execution algorithm of ESMDL model. It mainly extends the execution part of SDF based on the original basic model execution algorithm of SMP2, such as pipelined serial and parallel simulation scheduling algorithms, so as to improve the simulation execution efficiency of ESMDL model. Finally, taking a certain radar system application as an example, the new ESMDL simulation modeling language is used to design the radar model, including structure and behavior, to verify the feasibility and effectiveness of the project research content.

3.2 ESMDL Metamodeling

ESMDL is a new domain specific language based on SMP2 for behavioral computing models such as SDF. The key of this technology is to find out the extension points for SDF based on the deep understanding of SMP2 metamodel, and straighten out which elements need to be extended and which elements need mapping. Generally, domain specific language consists of language model and concrete grammar. Language model, also known as abstract grammar model, abstracts domain knowledge by defining elements of the target domain. It is composed of three parts: core language model, language model constraint and behavior description.

Firstly, the core language model defines the concepts related to a specific target domain and the relationship between them, which is usually described by appropriate modeling language, such as UML class diagram. Secondly, language model constraints, also known as static semantics, are a necessary part of ESMDL language model. These constraints define additional semantics by creating invariants on concepts or relationships in the core language model and creating pre - and post conditions on operations, which are usually expressed in formal constraint languages. For example, the core language model defined by UML class diagram uses OCL to express constraints. For constraints that cannot be expressed in formal language, they can also be given in natural language, but such constraints cannot be automatically processed and debugged by relevant tools. Thirdly, ESMDL behavior description, also known as dynamic semantics, defines a series of effects caused by the use of language elements, such as language elements. Usually, the real-time interaction can be expressed by UML activity diagram and sequence diagram [14].

The Synchronous Data Flow Metamodel. It is known as static data flow, is a special form of data flow model. In the data flow model, as long as the input data of the role meets the requirements. SDF is a simpler data flow, whose role execution order is static and does not depend on the data.

As shown in Fig. 2, the root node is schedule, which is composed of one or more nodes and transitions. One node can be connected to multiple transitions; on the contrary, a transition can only be connected to one node. A node can contain multiple input/output ports, and each port is associated with at least one token (It is an abstract class of all data, which is divided into two types required before and after the role is executed. The data transferred between roles are instances of token, making the real data invisible. The specific type of token is declared by the user in the role definition). A transition can have a buffer buffer (data transfer between nodes in a data flow graph can be implemented by using a FIFO queue buffer, and the size of this buffer will vary with the execution of the graph).

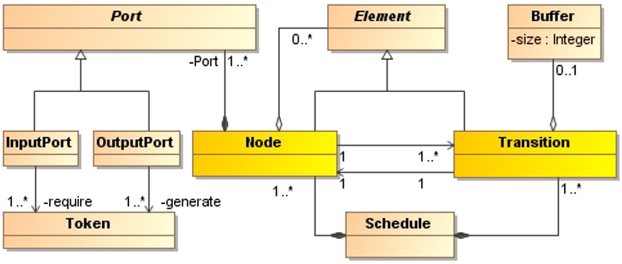

Fig. 2. The SDF metamodel.

The SMP Metamodel. It consists of three parts: catalog, assembly and schedule. The extension points related to SDF exist in the latter two parts, assembly (as shown in Fig. 3) and schedule (as shown in Fig. 4). In the assembly metamodel, links allows you to connect all model instances to the assembly itself.

1. Interface chain. It is responsible for connecting references and provided interfaces, used for interface based design and component-based design.

2. Event chain. It connects event slot with event source of the same type. For event based design.
3. Field chain. It connects input and output fields of the same type.

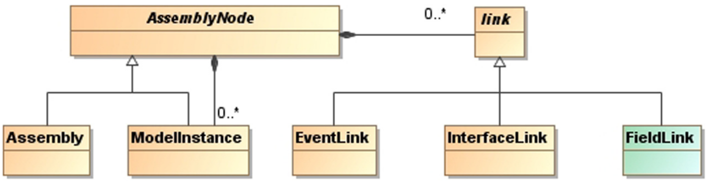

Fig. 3. SMP2-Assembly metamodel (part).

In the schedule metamodel, tasks are containers for activities. The order of activities defines the calling order of the entry points referenced by the activity element. Events are elements in a schedule that trigger tasks that themselves refer to entry points for model instances. Events can be timed events or periodic events. The timed event and timed event belong to the trigger that is called only once by the scheduler. The deltatime element is a relative time, which specifies when the event is executed; the kind attribute specifies the time type and supports the following four time formats.

1. Simulation time: start from 0 moment of simulation start, and advance in the execution mode of simulation.
2. Epochal time: it is an absolute time, which is typically advanced with simulation time. And simulation time offset may change during simulation.
3. Mission time: it is a relative time, expressed as the duration from a certain beginning, which is typically advanced together with epoch time.
4. Zulu time: is the computer clock time, also known as the wall clock time, regardless of the simulation stage will advance forward.

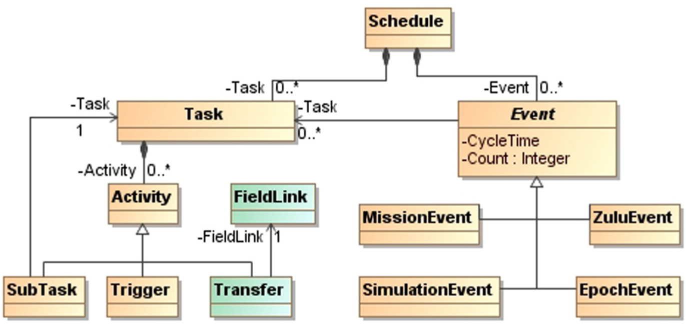

Fig. 4. SMP2-Schedule metamodel (part).

SMP2-SDF Metamodel Mapping. Based on the analysis of the above SMP2 and SDF meta models, the extended points for SDF in SMP2 metamodel are found out, and the mapping relationship of smp2-sdf meta model is defined, which elements need to be used, which elements need to be extended or added, as shown in Table 1.

Table 1. SMP2-SDF metamodel mapping.

SDF	SMP2	ESMDL	Note
Schedule	Schedule	Schedule	Original
Node	Task	Task	Original
Port	FieldLink/EventLink/InterfaceLink	Port	extended
Transition	Sink/Source, output/input	Transition	extended
Token	None	Token	Added
Buffer	None	Buffer	Added

3.3 ESMDL Scheduling Strategy

In SMP2, schedule is a document, which contains any number of tasks (task elements) and the events that trigger these tasks. It is necessary to specify the event triggering strategies to get the execution order of tasks. These tasks refer to the entry points of model instances. In SDF, role triggering requires multiple input tokens and does not necessarily produce a single token, which is called multi rate SDF. In this case, it is necessary to define the execution order of roles and the execution times of each role, that is, the model execution scheduling problem.

Event Scheduling. The event scheduling method first appeared in the early version of SIMSCRIPT language, which was introduced in 1963. Its basic idea includes:

1. The event routine is regarded as the basic model unit of simulation model, and the corresponding event routine is executed continuously according to the sequence of events.
2. Every event that can be predicted in advance has an event routine to deal with the impact of the event on the entity and arrange subsequent events.

Serial Scheduling. At present, the most classic and used static scheduling method for synchronous data stream is the S-class static scheduling method. This algorithm can find a feasible period serialization execution schedule of synchronous data flow graph under the condition of single processor scheduling, also known as pass (periodic adaptive sequential schedule). An effective pass must satisfy the following three conditions:

1. Full participation. This ensures that each node is executed, that is, each node in pass appears at least once.

2. No deadlock. When there is delay unit, deadlock will not occur.
3. Buffer bounded. The output data generated by each node can be consumed equally by other nodes without causing buffer accumulation.

Parallel Scheduling. As shown in Fig. 5, in the parallel scheduling method, it is not necessary to wait for the completion of one iteration to start the next iteration, but it can be executed in parallel with the nodes in other iterations after the execution of one node in one iteration. Compared with the serial scheduling method, especially for the radar system which needs many Monte Carlo simulation experiments, the time utilization is greatly improved.

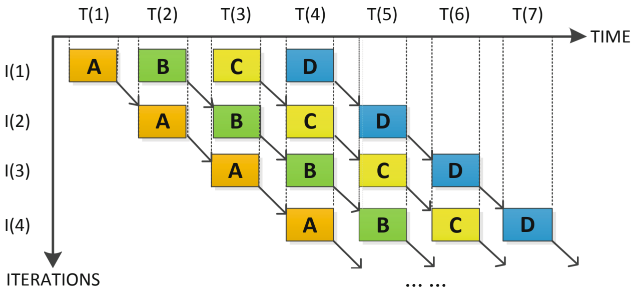

Fig. 5. Parallel scheduling.

4 Case Study

The system model framework based on ESMDL is divided into different domains and levels, which generally includes static structure, physical domain behavior and cognitive domain behavior horizontally, and system layer, subsystem layer, business component layer, algorithm support layer and computing model layer vertically [15, 16], as shown in Fig. 6. In this way, the flexibility, reusability and expansibility of radar system model are enhanced both vertically and horizontally, which supports the rapid development of radar system models with different functional characteristics and meets different radar system simulation requirements.

As can be seen from the above figure, the knowledge structure of model framework is divided into three parts: static structure, physical domain behavior and cognitive domain behavior.

1. Static structure mainly describes the related concepts and relationships in the domain, which is the most stable part and does not change with the change of human will or environment [17].
2. Physical domain behavior describes the concepts such as data flow, state change, event migration in the domain, which is the inherent behavior mode of an entity, and its flexibility is between the static organizational structure and the behavior of

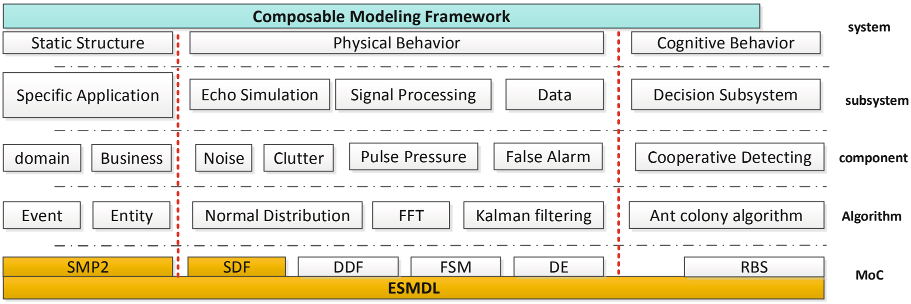

Fig. 6. CESS onto CMFs.

cognitive domain. It is usually described by synchronous data flow, dynamic data flow, finite state machine and discrete event (DE). The project plans to focus on synchronous data flow, and consider the future expansion requirements of other computing models when designing ESMDL metamodel.

3. Cognitive domain mainly describes tactical rules, which are the most flexible part of the three, and will vary with the changes of combat scenarios and commanders, and rule-based system (RBS) is usually used. The physical domain behavior can flexibly use entities and relationships in static structure, and achieve the purpose of reusing relatively flexible parts and relatively stable parts to a large extent without completely rewriting the model.

Vertically, the model framework is divided into five levels from top to bottom: system layer, subsystem layer, component layer, algorithm layer and computing model layer. When defining the upper layer, the appropriate concepts are selected from the next layer according to different applications to be specialized and extended to realize the reuse of concepts and relationships. This model can ensure high standards and quality.

1. The system layer is mainly for application, and users can easily build radar simulation system according to their needs by customizing various subsystems.
2. The subsystem is similar to the system layer, but with finer granularity, it can also customize and combine the lower level components to build subsystems with different functions freely.
3. The component layer is the module with specific functions. It can be an atomic module or a composite module assembled by lower level algorithms.
4. The algorithm layer is a composite function that encapsulates different mature algorithms into functions or a composite function constructed by simple algorithms, which is usually used to construct upper level components.
5. The computational model layer is mainly used to provide running environment and model semantics and control different levels of simulation model, and organize the running of simulation system.

5 Conclusions

The behavioral modeling technology based on SMP2 combines the advantages of SMP2 and SDF. It explores a system simulation modeling method which can support the description of system structure and behavior at the same time. Thus, this method can acquire high efficiency, strong reusability and easy expansion, so as to improve the development efficiency and construction quality of simulation system, and save the development cost. Furthermore, this technology provides an important reference for SMP2 to provide more behavioral computing models. It can further consider the expansion of dynamic data flow, state machine, discrete event and other behavioral computing models, so as to solve the development needs of large-scale, multi system and complex functions of current and future radar systems. In the face of new combat system or demand, it can realize the modeling and simulation of new simulation system comprehensively, rapidly and high quality, provide support for the system demonstration, development, test identification, use training and comprehensive support of radar equipment, and improve the scientific level of radar equipment development. However, as a drawback more simulations about different scenarios are necessarily required as further illustrations.

Acknowledgement. We are grateful to the anonymous referees for their helpful reviews, and all the volunteers who wrote and provided helpful comments on previous versions of this document. We gratefully acknowledge the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System (CEMEE) (NO. CEMEE2020K0302A) for supporting this research.

References

1. Taylor, S.J.E., Khan, A., Morse K.L., et al.: Grand challenges on the theory of modeling and simulation. In: Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative M&S Symposium, pp. 1–8. Society for Computer Simulation International, San Diego (2013)
2. Lei, Y.L., Zhu, Z., Li, Q.: An ontological metamodeling framework for semantic simulation model engineering. J. Syst. Eng. Electron. **31**(3), 527–538 (2020)
3. Tolk, A.: The next generation of modeling & simulation: integrating big data and deep learning. In: Proceedings of the Conference on Summer Computer Simulation, pp. 1–8. Society for Computer Simulation International, San Diego (2015)
4. IEEE Computer Society. https://standards.ieee.org/standard/1516-2010.html. Accessed 25 Aug 2020
5. ESA. https://www.eurosim.nl/support/manuals/manual_4_2/pdf/SMP_2.0_Metamodel-1.2.pdf. Accessed 28 Aug 2020
6. Kim, B.S., Kang, B.G., Choi, S.H., et al.: Data modeling versus simulation modeling in the big data era: case study of a greenhouse control system. SIMULATION **93**(7), 579–594 (2017)
7. Zeigler, B.P., Praehofer, H., Kim, T.G.: Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, 2nd edn. Academic Press, New York (2000)

8. Tendeloo, Y.V., Vangheluwe, H.S.: Classic DEVS modelling and simulation. In: Winter Simulation Conference, Las Vegas, Nevadam, pp. 644–658. Society of International Modeling and Simulation (2017)
9. Azar, M.C.: Assessing the Treatment of Airborne Tactical High Energy Lasers in Combat Simulations. M.S. Thesis, Air Force Institute of Technology, Dayton, OH (2003)
10. Miller, J.O., Jason, L., Honabarger, B.: Modeling and measuring network centric warfare (NCW) with the system effectiveness analysis simulation (SEAS). In: 11th International Command and Control Research and Technology Symposium, pp.1–12 (2006)
11. Pidd, M.: Tools for Thinking-Modeling in Management Science. Wiley, New York (2009)
12. Antoine-Santoni, T., Poggi, B., Vittori, E., Van Hieux, H., Delhom, M., Aiello, A.: "Smart Entity" – how to build DEVS models from large amount of data and small amount of knowledge? In: Song, H., Jiang, D. (eds.) SIMUtools 2019. LNICSSITE, vol. 295, pp. 615–626. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32216-8_60
13. Grieves, M.W.: Product lifecycle management: the new paradigm for enterprises. Int. J. Prod. Dev. **2**(1–2), 71–84 (2005)
14. Deist, T., Patti, A., Wang, Z., et al.: Simulation assisted machine learning. Bioinformatics **35**(20), 1–1 (2018)
15. Atkinson, C., Kuhne, T.: Model-driven development: a metamodeling foundation. IEEE Softw. **20**(5), 36–41 (2003)
16. Zhu, Z., Lei, Y.L., Zhu, N., Zhu, Y.F.: Composable modeling frameworks for networked air & missile defense systems. J. Nat. Univ. Defense Technol. **36**(5), 186–190 (2014)
17. Zhu, Z., Lei, Y.L., Zhu, Y.F.: Model-driven combat effectiveness simulation systems engineering. Def. Sci. J. **70**(1), 54–59 (2020)