Collaborative Work in Domain-Specific Discrete Event Simulation Software Development: Fleet Anti-air Defense Simulation Software

Chang Ho Sung*, Il-Chul Moon†, and Tag Gon Kim‡
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
Daejeon, Korea
Email: chsung@smslab.kaist.ac.kr, icmoon@smslab.kaist.ac.kr†, tkim@ee.kaist.ac.kr‡*

***Abstract*—Modeing and simulation (M&S) is an important method to evaluate numerous designs and operational concepts for a real-world system. If a system to be modeled is domain-specific, developing the simulation software of the system will require domain knowledge about the system as well as understanding the M&S methodology. This paper describes M&S stakeholders and proposes a collaborative work process in the development of domain-specific simulation software. M&S stakeholders are persons with their own professional knowledge: subject matter experts (SME), domain engineers, M&S engineers, platform engineers, and simulation data analysts. The M&S process consists of eight activities from defining modeling objectives to analyzing simulation data, and diverse M&S stakeholders work together in each activity. The M&S process is applied to develop a real-world M&S software development experience in a military domain. Through the proposed collaborative work process, the capabilities of the M&S stakeholders can be utilized maximally by seamlessly separating yet correlating their works.**

***Keywords*-Modeling and simulation (M&S), Software development process, M&S stakeholders, Collaborative work**

I. Introduction

Discrete event modeling can be considered a process of abstract knowledge representation of a real-world system. As a model, the representation should be executable within a simulation environment to analyze the modeled system with respect to modeling objectives. The model can vary according to the modelers' different viewpoints, such as event-oriented, process-oriented, object-oriented, etc. Among them, the object-oriented (OO) approach may be most compatible with a real-world system from the system-theoretic viewpoint of model representation [1]. The DEVS formalism [2], which represents a discrete event system from the system-theoretic viewpoint, is known to be compatible with the OO world view.

Given modeling objectives, modeling and simulating domain specific systems require a great deal of domain knowledge and modeling and simulation (M&S) expertise. Some may approach developing simulation software as an ordinary software development. However, we often find that there more expertise is involved in the development. If we were to assume that a software engineer has in-depth knowledge in software (SW) technologies, i.e. programming, and if the engineer develops defense simulation software, the software engineer needs information about military science. Even though the software engineer may have extensive data about weapons, strategy, tactics, etc., the engineer cannot make the best use of the data in developing simulation software because of the lack of domain-specific knowledge about military science. In addition, the software engineer may still fail to develop a simulation model because of a lack of M&S knowledge. The simulation model is defined by a set of instructions, rules, equations, and constraints for generating output behaviors from inputs. No matter how much the software engineer may have learned from military science, the engineer cannot develop the defense simulation model without certain M&S theory.

To overcome the difficulty of developing domain-specific simulation software, a collaborative work process is required, and this paper proposes such a work process found to be efficient from our experiences in developing defense simulation software. The essence of the work process is assigning the right participants with the right expertise in the right place and time. This paper specifies M&S stakeholders as participants with expertise, and this paper also proposes a collaborative work items and processes among the participants for domain-specific simulation software development.

This paper is organized as follows. Section II presents the M&S process and stakeholders for domain-specific simulation software development. Our proposed collaborative modeling process and case studies are described in Section III. Section IV concludes the discussion.

II. M&S Software Development

A. M&S Process

Generally, we have several alternative software development processes, i.e. waterfall, spiral, agile process, etc. If we follow the waterfall process, the most popular method, the process is composed of the following development activities: *planning, requirements analysis, design, implementation, testing, deployment, and maintenance*. Figure 1 provides a side-by-side comparison between a general software development process and the M&S process for the development of simulation software. The M&S process

978-0-7695-4063-4/10 $26.00 © 2010 IEEE
DOI 10.1109/WETICE.2010.31

may correspond to the software development cycle, but the M&S cycle has distinct activities in the model, such as *motivation, requirements analysis, conceptual model design, detailed model design, simulation software implementation, verification and validation, simulation experiment design, and simulation data analysis*. Software models for general software engineering are structural and behavioral specifications of executable software. On the other hand, simulation models of M&S engineering are concerned with dynamics beyond what could eventually be supported by the real system [3]. The activities of the M&S process will be described in detail in Section III.

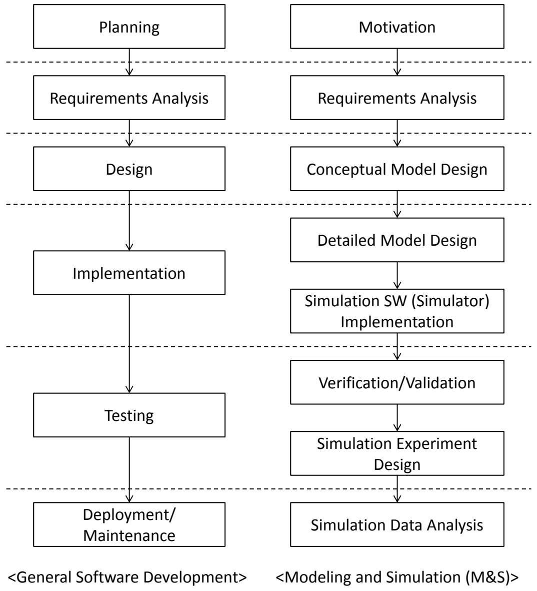


Figure 1. Comparison of activities between general software development and M&S process

B. M&S Stakeholders

In this paper, our definition of an M&S stakeholder is anyone who participates in developing domain-specific discrete event simulation software. In traditional software engineering, the stakeholders include software customers, software developers, testers, etc. These stakeholder categories will be mapped to five groups when we consider the characteristics of the M&S work process. The five groups are *subject matter expert*, *domain engineer*, *M&S engineer*, *platform engineer*, and *simulation data analyst*. These groups are distinct in their interests, activities, and expertise during the M&S work process.

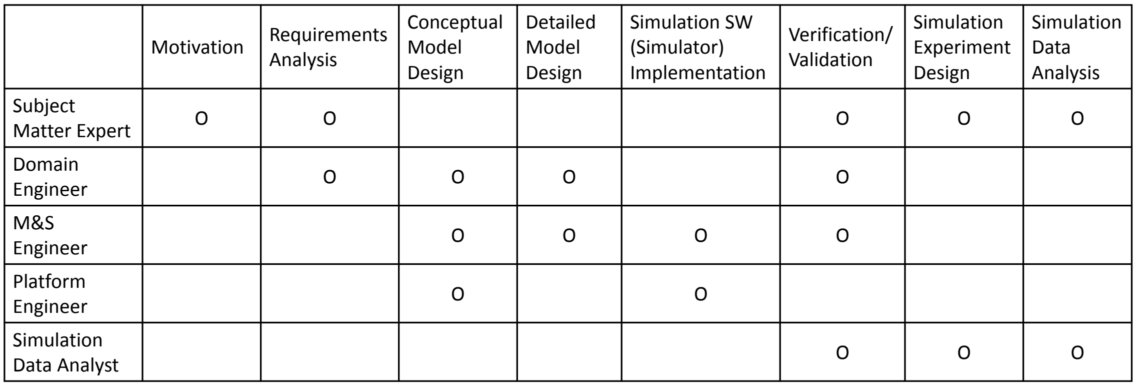
1) Subject matter expert: In software engineering, a subject matter expert (SME) is a person who has expertise in the field of domain application, yet no technical knowledge. The definition of SME in our proposed collaborative M&S process is the same. SMEs are expected to tell the simulation software developers what needs to be done as well as the SMEs' motivation for the development from the software user's perspective. SMEs are also involved in validating the developed simulation software by providing a dataset for the statistical validation process.

2) Domain engineer: Domain engineering (DE) is a software engineering discipline that includes the identification, analysis, and design of domain-specific software capabilities. In general, DE builds software architectures and reusable assets to address the problems of system development within a domain [4]. A domain engineer is involved in performing the domain requirement analysis and design. Based on the domain requirements, the domain engineers generate the behavioral level model, which includes mathematical equations, algorithms, and strategies for domain-specific modeled entities.

3) M&S engineer: M&S engineering refers to the development and use of simulation models to generate data as a basis for making managerial or technical decisions [5]. The activities of M&S engineering include requirement engineering, modeling theory, simulator implementation/verification, model validation, model behavior analysis, and performance evaluation. In the process described in this paper, an M&S engineer is in charge of the overall process related to the modeling and simulation of discrete event systems satisfying domain requirements. The M&S engineers are especially involved in designing the high-level abstract behavior of objects. They generate the discrete event level model, which can be specified as the DEVS formalism.

4) Platform engineer: A platform engineer is a general program developer who implements behavioral models specified by the domain engineers. The platform engineer also takes charge of integrating simulation environments, i.e. GIS, database, and user interface. He also possesses expertise in the platform technologies where simulation models will be deployed. The distinction between the domain engineer and the platform engineer is that the domain engineer will distill domain-specific knowledge into implementation-related knowledge, so the platform engineers do not have to know the details of the overall domain-specific concepts.

5) Simulation data analyst: A simulation data analyst is a person who analyzes the simulation results statistically and provides alternatives to the SME. The data analysts will acquire simulated data, perform the validation of the results compared to the real world, and provide alternative policy recommendations, training methods, acquisition plans to the SMEs. The data analysts are adept in designing simulation experiments to test SMEs' various hypotheses, i.e. whether acquiring certain equipment will make a statistically significant improvement in the domain or not. They use various statistical techniques ranging from the simple t-test to the ANOVA, non-linear regression, etc.



	Motivation	Requirements Analysis	Conceptual Model Design	Detailed Model Design	Simulation SW (Simulator) Implementation	Verification/ Validation	Simulation Experiment Design	Simulation Data Analysis
Subject Matter Expert	O	O				O	O	O
Domain Engineer		O	O	O		O		
M&S Engineer			O	O	O	O		
Platform Engineer			O		O			
Simulation Data Analyst						O	O	O

Figure 2. Different stakeholders for various activities of the M&S process

C. *Collaborative Work in M&S Software Development*

These stakeholders are responsible for different parts of the M&S process, which cause different stakeholder groups to engage in different activities. We specified the activity group via the M&S process in Figure 2. For instance, the activity of conceptual model design will require the participation of the domain engineer, the M&S engineer, and the platform engineer because the activity will result in the overall behavior of the M&S software. However, the detailed model design will specify the behavior, not the implementation, of the software, so the platform engineers' involvement will be limited.

III. Collaborative M&S Process and Application

The development of domain-specific discrete event simulation software is a complex task because the task requires in-depth professional knowledge from diverse stakeholders. To mitigate the complexity, we previously proposed a collaborative modeling work process [6], and the process is focused on the development of M&S software. This paper extends the previous work by including the requirements analysis; the verification and the validation analysis; the simulated data analysis; and the real-world application, which are important parts of the M&S software process. As shown in Figure 3, the overall modeling and simulation process is partitioned into eight stages, and diverse M&S stakeholders work together in each stage. At the end of each activity description, we also add a real-world M&S software development experience in a military domain. Our story explains our cooperative modeling effort made to develop fleet anti-air defense simulation software in 2009.

A. *Activity 1: Define Modeling Objectives*

The first activity that initiates the M&S software development is the *motivation* to simulate a real-world situation from a certain perspective. For instance, SMEs often require simulation software to measure the effectiveness and performance of a domain system, or to train personnel in a domain environment. These motivations will provide the M&S objectives.

Case study application: The Republic of Korea Navy (ROKN) launched a new destroyer that is capable of providing anti-air defense for a fleet. Thus, the ROKN needed a new doctrine in the fleet formation that provides better air cover by placing the new destroyer in the right position in the formation.

B. *Activity 2: Requirements Analysis*

Requirements analysis is done in order to analyze SMEs' needs for simulation software. This stage starts with gathering domain information by concentrating on what the domain system is and requires. Domain information often is gathered through questionnaires or direct interviews with SMEs [7]. Domain engineers define the purpose and the overall functions of the simulation software by distilling the domain information. As a result of this stage, the domain engineers develop textual descriptions, called the *requirements specifications*, of the software.

Case study application: The research branch of the ROKN surveyed the operators and the commanders and provided the overall requirements of the M&S software. The branch is versed in the military domain, and the branch also had superficial knowledge about M&S theories and practices. Eventually, the branch produced a Software Request Specification (SRS) to be sent to the M&S experts.

C. *Activity 3: Conceptual Model Design*

The next activity is the conceptual model design. A conceptual model is a specification of the modeled entities and their links from the domain engineers' requirements specifications. The conceptual model is the basis for subsequent M&S software development in the domain [8] by providing a high-level description of implementation details. The conceptual model is often represented by the Systems Modeling Language (SysML), the Unified Modeling Language (UML) and the Entity Relationships (ER). These diagrams are developed through discussions among domain, M&S, and platform engineers, and the discussions will distill the domain engineers' requirement specification into a structure of model entities.

Case study application: After the SRS was compiled, the ROKN research branch and the Korea Advanced Institute of Science and Technology (KAIST) discussed what entities would be modeled and how they would be linked. For example, we identified that the fleet command and control (C2) structure, the warships, and the air threats, i.e. missiles and aircraft, should be modeled, and we also recognized that the C2 structure and the warships should be linked to send and receive order information.

D. Activity 4: Detailed Model Design

While the conceptual model sketches the modeled entities and relations, the detailed model design describes how to decompose the model entities into sub-models and how to specify the behavior of the model entities in detail. The architectural design of the model decomposition proceeds from the object-oriented (OO) modeling viewpoint. The OO modeling approach for systems modeling regards a system as an object by explicitly defining the object's state and associated operations. This definition can be partitioned into two levels: discrete event-level model (DEM) and detailed behavioral-level model (BM). The detailed model will describe the dynamic behavior of a modeled object, and the behavior includes the input and the output (I/O) events of a model, state transitions, and operations processed by specific events and state transitions. The M&S engineer specifies the state transitions by I/O events as the DEM, and concurrently, the domain engineer describes the detailed operations for the specific states of the modeled object using algorithms and mathematical equations as the BM.

Case study application: The DEVS diagram specifies what models will be created and how they will be integrated. These models in anti-air defense are the C2 structure, the warships, and the air threats. KAIST and the ROKN research branch decomposed, for example, the air threats into propulsion, seeker, and targeting. Similar decompositions were applied to the C2 structure and the warships, as well. From the decomposition, KAIST specified the state changes via air threat events, such as cruising after launch, targeting after warship detection, and detonation after approach to the target. The ROKN research branch implemented the trajectory of the threat cruising, the targeting mechanism of the anti-ship missiles (ASM), and the blast radius of the ASM. The state transitions triggered by events are modeled using discrete event modeling by KAIST, and the behavior of the states are modeled using behavioral modeling by the ROKN research branch.

E. Activity 5: Simulation Software Implementation

The DEM and BM are implemented using their own implementation environment. Using the DEVS simulation environment, M&S engineers generate code artifacts of the specified DEVS models that represent system behaviors at the state-transition level. Similarly, platform engineers implement specific algorithms as individual functions, which are defined by domain engineers, using programming languages. Therefore, two stakeholders conduct the simulation software implementation simultaneously. This means that there is collaborative work in software programming. The completed simulation software is the integrated form of the DEMs and the BMs, and each model is coupled with an implementation of communication interfaces.

Case study application: The detailed model design produced the function specifications and the DEVS model specifications for KAIST, who undertook the implementation of both DEM and BM by playing the M&S engineer and the platform engineer roles at the same time. KAIST used the DEVSim++ [9] for the discrete event modeling implementation; the DEVSim++ is a C++ library for the DEVS formalism. Also, KAIST implemented the behavior of air-threats and warships as dynamically linked libraries. Then, the two implementations were integrated and became runnable under a simulation environment provided by DEVSim++.

F. Activity 6: Verification and Validation

The integrated DEM and BM consists only of the core of the simulation software, so the M&S engineers need to provide interfaces to utilize the simulation model. For example, a detailed human-computer interaction interface will be needed in the use of simulation training, or a statistical result organizer will be needed to run a simulation experiment. These interfaces will provide the simulation usage results, and we call this as an experimental frame. After the interface is established, the simulation software will be verified and validated. First, M&S engineers test the simulator to check the accuracy of converting a model representation into simulation software. We call this the simulation verification. After verifying that the model is implemented as designed, the statistical analysis will follow to compare the simulated data to the real-world data; this procedure is the simulation validation.

Case study application: Since the implementation from KAIST was verified by the research branch of the ROKN, the verification is done by observing the graphical user interface (GUI) showing the friendly behavior and the threat behavior. There was no formal validation comparing the simulation results to the real-world data. The target domain, the anti-air defense of a fleet, rarely happens, and realistic data is difficult to come by.

G. Activity 7: Simulation Experiment Design

After model verification and validation, stakeholders determine how the simulation software will be utilized. For instance, SMEs may generate a possible scenario to run the simulation training, and the data analysts will break down the results into the training performance. Another example is applying many alternative possible scenarios from SMEs and finding a better strategy through simulation runs. These virtual or constructive usages [10] will be run by SME scenarios and analyzed using statistical methods [11].

Case study application: The developed simulation software is executed by many scenarios that the ROKN suggested and KAIST designed. For example, a ROKN officer pointed out the possible number of warships and their formation alternatives. There were too many possible combinations, so KAIST applied a Nearly-Orthogonal Latin Hypercube (NOLH) design method to the possible combinations, and KAIST produced 34 scenarios and corresponding friendly survival ratios from the simulation experiments. Then, KAIST ran the scenarios multiple times to ensure statistical significance.

H. Activity 8: Simulation Data Analysis

The simulation runs generate a series of values for the analysis index and corresponding scenarios. Then, the question becomes how to analyze the corresponding scenarios by looking at likely outcomes or favorable outcomes. Simulation data analysts may build a meta-model to find a variable impacting the favorable outcomes or perform a t-test on the analysis indexes from two alternatives to find a better one. At the same time, the variance of simulation results should be significant enough to clearly distinguish favorable scenarios. For this purpose, the simulation analyst may perform ANalysis Of VAriance (ANOVA). At the end of the analysis, the analysts may report their findings, an important variable inducing favorable outcomes, a significant performance difference from one scenario to another, etc., to SMEs for the domain application.

Case study application: The 34 scenarios that KAIST executed generated corresponding analysis indexes, such as friendly warship survival rates and threat intercept rates. KAIST marginalized the 34 scenarios into two fleet formation cases and identified a better formation resulting in a better analysis index value. This identified formation was suggested to the ROKN, and the ROKN will apply this result to their doctrine development.

IV. Conclusions

This paper proposes collaborative work in domain-specific discrete event simulation software development and application. Because going through M&S software development and application is difficult to execute without understanding any domain knowledge or M&S methodologies, collaborative work among M&S stakeholders is imperative. The M&S stakeholders must work together in the whole process from the requirements analysis to the simulation data analysis for M&S software development. Through our proposed work process, we mitigated collaboration difficulties by allowing experts to do their jobs simultaneously with technical supports. We expect that this work process and story may provide insights into future M&S software development and application.

Acknowledgment

This work was supported by the Defense Acquisition Program Administration and the Agency for Defense Development under contract UD080042AD, Korea.

References

[1] C. A. Roberts and Y. M. Dessouky, "An overview of object-oriented simulation," *SIMULATION*, vol. 70, no. 6, pp. 359–368, 1998.

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

[3] A. E. Ferayorni and H. S. Sarjoughian, "Domain driven simulation modeling for software design," in *SCSC: Proceedings of the 2007 summer computer simulation conference*. San Diego, CA, USA: Society for Computer Simulation International, 2007, pp. 297–304.

[4] M. E. Stropky and D. Laforme, "An automated mechanism for effectively applying domain engineering in reuse activities," in *TRI-Ada '95: Proceedings of the conference on TRI-Ada '95*. New York, NY, USA: ACM, 1995, pp. 332–340.

[5] E. H. Page and R. Smith, "Introduction to military training simulation: a guide for discrete event simulationists," in *WSC '98: Proceedings of the 30th conference on Winter simulation*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1998, pp. 53–60.

[6] C. H. Sung, S.-Y. Hong, and T. G. Kim, "Layered Structure to Development of OO War Game Models Using DEVS Framework," in *SCSC 2005*, July 2005, pp. 65 – 70.

[7] W. Tracz, L. Coglianese, and P. Young, "Domain-specific software architecture engineering process guidelines," in *Domain-Specific Software Architecture Engineering Process Guidelines, Owego, NY: IBM Federal Systems Company ADAGE-IBM-92-02*, 1993.

[8] J. L. de la Vara, M. H. Fortuna, J. Sánchez, C. M. L. Werner, and M. R. S. Borges, "A Requirements Engineering Approach for Data Modelling of Process-Aware Information Systems," in *BIS*, 2009, pp. 133–144.

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

[10] D. D. Hodson and R. O. Baldwin, "Characterizing, measuring, and validating the temporal consistency of live-virtual-constructive environments," *Simulation*, vol. 85, no. 10, pp. 671–682, 2009.

[11] K. M. Carley, "On generating hypotheses using computer simulations," *Systems Engineering*, vol. 2, no. 2, pp. 69–77, 1999.

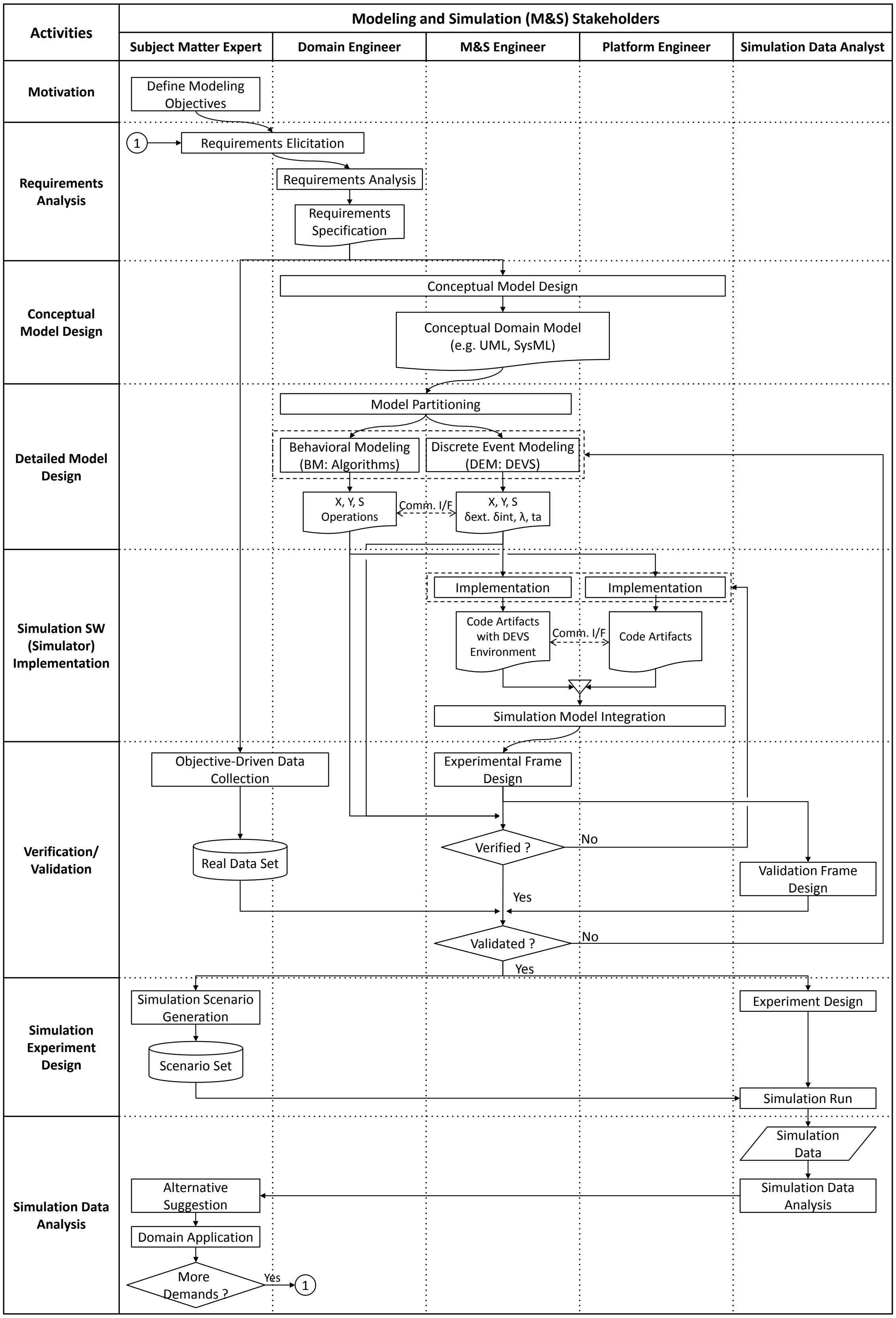


Figure 3. Overall collaborative modeling and simulation process