Study on Modeling Formalism and Co-simulation for the System-level Test of Maritime Components

Jeongsik Kim
Ulsan Intelligent Convergence Research Section
Electronics and Telecommunications Research Institute
Ulsan, Republic of Korea
j.s.kim@etri.re.kr

Woo-sung Jung
Ulsan Intelligent Convergence Research Section
Electronics and Telecommunications Research Institute
Ulsan, Republic of Korea
woosung@etri.re.kr

Namhun Kim
Department of Mechanical Engineering
Ulsan National Institute of Science and Technology
Ulsan, Republic of Korea
nhkim@unist.ac.kr

***Abstract*—The advance of information and communications technology has rapidly sophisticated maritime systems, while the development of a proper product for the enhanced system has become more difficult. System-level simulation has been becoming a key solution to find and verify the adequacy of products on a targeted complex system through the product life cycle. However, the reusability of simulation models is still limited. In this context, this paper studies the benefits from the introduction of a state-of-the-art modeling formalism to complement co-simulation approaches. This consideration can be utilized to facilitate the practical application of system-level simulation.**

Keywords*—*computational simulation, co-simulation, discrete event system specification, modeling formalism, system-level simulation

I. INTRODUCTION

Simulation-based approaches have been becoming pervasive but there are a few challenges to facilitate the practical applications of simulation-based approaches in modern industries [1,2]. The advance of information and communications technology has sophisticated the complexity of real world systems to model. It becomes more difficult when the target system consists of many components and manufacturers (e.g., maritime and automotive domains) by requiring the corresponding co-simulation of the components as an integrated system. Moreover, not a few simulation models have been developed in a systematic form and limits the reusability of models by requiring additional measures such as completing specification and building adapters.

To tackle this problem, co-simulation technologies have been developed for the system-level integration of different simulation models with little care of explicit specification for the reusability of each component and federated system. In this context, this paper studies the introduction of a modeling formalism, called discrete event system specification (DEVS), to complement co-simulation approaches in terms of the reusability of componential models and their federation. A brief review on key co-simulation techniques and basic modeling formalisms is provided as well.

II. RESEARCH BACKGROUND

A. Co-simulation

Co-simulation technology typically refer to tools that co-simulate heterogeneous and multi-domain systems. The major of them focuses on interface for the communication of different systems and operation for the synchronization of heterogeneous simulations without potential errors.

TABLE I. THE COMPARISON OF CO-SIMULATION STANDARDS [3,4]

Feature	Co-simulation Standards	
	Functional Mock-up Interface	***High-level Architecture Run-time Infrastructure***
Release year	FMI 2.0 (2014), FMI 3.0 (2022)	HLA 1.3 (1998), IEEE 1516-2000 (2000), HLA-evolved (2010)
Scope	Form of the container for simulation models and interfaces to manage them	Framework and rules, federate interface specification, object model template specification
Main target	Time-based unit models	Event-driven unit models
Support Method	Using application programming interface based on C to manage simulations in the form of ZIP file (.fmu)	Using a developed middleware and its functions to control federated simulations (federation)
Key tools	Matlab, Modelica	Pitch pRTI, MAK, CERTI

The basic features of two key techniques for co-simulation are described as below [3,4]:

- Functional Mock-up Interface (FMI) is a standard for the container, so-called functional mock-up unit (FMU), and interfaces to bridge dynamic models. Over 170 off-the-shelve programs, including Matlab and Modelica, provide the compatibility using the application programming interface (API) of FMI.
- High-level architecture (HLA) and run-time infrastructure (RTI) is a standard for three major specifications of co-simulation: i) the framework of federation and the rules to develop federates, ii) services of RTI to exchange information, iii) model template for a HLA object model to interact.

As listed in Table I, there are a few differences between FMI and HLA/RTI. FMI covers the container and interfaces while HLA/RTI deals with the broader management to realize the co-simulation. Thus, some engineers have applied the combined approach to embrace the both event-driven and time-based models such as [5] which uses RTI with an adapter for the FMI. Another popular technique in maritime industry is the application of an integration platform such as Matlab Simulink^{TM} to integrate third party simulation models based on co-simulation standards (FMI/FMU and S-functions) [6]. In the both standards, the description of components and their federation has been unnoticed, inhibiting not only their transparency and readability but the reusability of them.

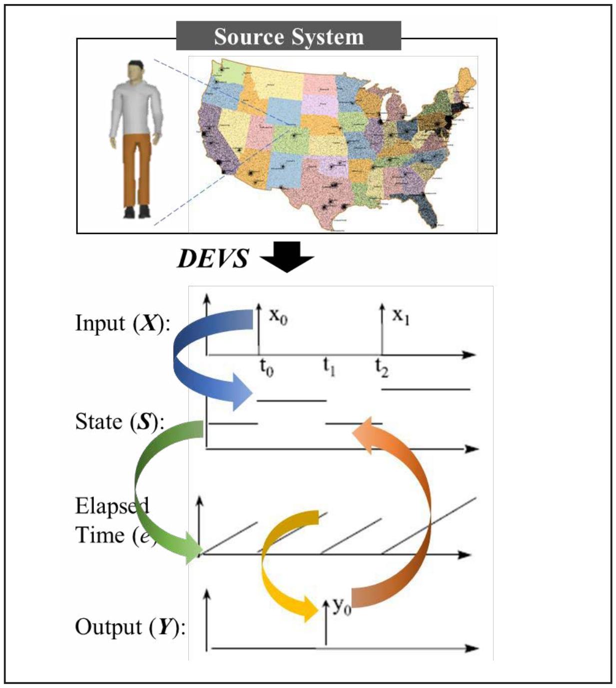


Fig. 1. The basic mechanism of discrete event system specification

B. Modeling Formalism

One of the main challenges in modeling has been the complete and clear specification of models. DEVS is one of the most popular specification languages based on set theory in mathematics [7]. DEVS allows modelers to specify system models in a systemactic form and reuse them for other purposes.

A classic DEVS formalism is structured as three sets and four functions as below.

$$DEVS=<X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta> \tag{1}$$

where the static information of a system is represented as *X*, *Y*, and *S*, which are the sets of inputs, outputs, and sequential states. While the dynamic information is represented as 4 functions (i.e., internal function, external function, output function and time advance function). Fig.1 shows how system changes with the functions. Simply, inputs and elapsed time invoke state transitions which reset the elapsed time and send outputs.

III. DEVS FOR MULTICOMPONENT SYSTEM

The classic DEVS exhibits the traditional concept of systems theory in which a system is considered as the sum of components. However, some modern systems used to be considered as a collection of interacting decentralized sub-systems. From the new viewpoint on systems, diverse variants of DEVS have been developed for more than two decades. One of the latest DEVS formalisms is MultiPDEVS for the specification of multicomponent and parallel system. A brief comparison between classic DEVS and MultiPDEVS is listed in Table II. The definition, interpretation, and example of MultiPDEVS formalism is specified in [8].

TABLE II. THE COMPARISON OF CLASSIC DEVS AND MULTIPDEVS

Feature	**DEVS Type**	
	Classic DEVS	***MultiPDEVS***
Structure	3 Sets, 4 Functions	3 Sets, 6 Functions
New Item	-	Confluent and Reaction Functions
Advantage	The Simplest Specification	Specification of Parallel & Nonmodular Interactions
Example	A Bank Cashier	Fire Spread on Cellular Map

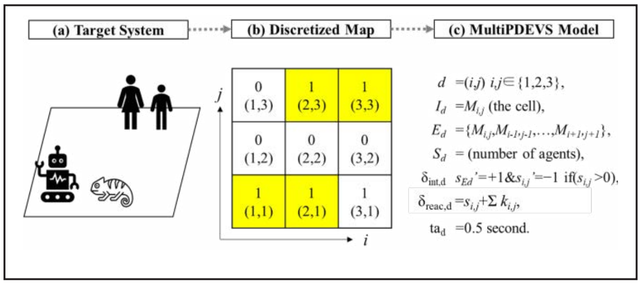


Fig. 2. The example of transfomrations based on MultiPDEVS

A MultiPDEVS formalism is structured as below:

$$MultiPDEVS=<X, Y, D, \{M_d\}> \tag{2}$$

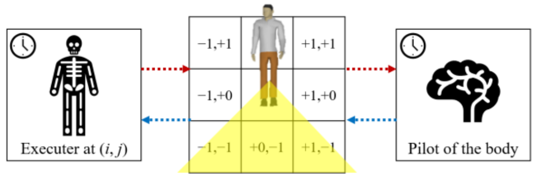
where *X* is the set of input events, *Y* is the set of output events, *D* is the set of component references for $d\in D$, and an atomic component, M_d, for each *d* is structured such that:

$$M_d=<S_d, I_d, E_d, \delta_{int,d}, \delta_{ext,d}, \delta_{con,d}, \delta_{reac,d}, \lambda_d, ta_d> \tag{3}$$

where $\delta_{reac,d}$ is the reaction transition function to define the next transition according to the current state and a set of bags of suggested states (k_d) for *d* over elements.

With the edited structure and two added functions at the atomic component, a MultiPDEVS formalism specifies the parallel events and nonmodular multiple components. In other words, a new phenomenon could be represented when two events occurs at the same time (recorded on simulations) with the confluent function while interactional effects for the both components are represented with the reaction function. For instance, Fig. 2. illustrates the transformation based on MultiPDEVS. Four objects on a floor of a system can be discretized into the form of cellular automata. Each cell can be represented by mathematical notations. The number of objects in a cell is directly calculated with the defined functions. Each cell is updated by an internal event (i.e., a mover on the cell leaves) as well as reactional events (i.e., movers enter the cell).

In fact, the representation on the componential system has been appeared in prior formalisms while the representation on their direct influences in MultiPDEVS contributes to the theory of system modeling. The direct influence lets each componential system to change another without an indirect function from the upper level of system. Classic formalisms such as ml-DEVS [9] aims at representing multiple components but sticks to top-down event handling between atomic models from the viewpoint of the traditional systems theory. This indirect message could cause overhead cost with potential source of unintended operations and errors by incomplete representation. Thus, MultiPDEVS helps the concise specification of atomic systems and their atomic relationships become clearly specified and easier to reuse.



d=ad (Executer)	Items	d=id (Pilot)
$signal=(a,b)$, $position=(i,j)$, $pilot=id$	*State*	$percept=\{\ldots, \text{available}_{+0,-1}, \ldots\}$
$M_{ad}.position'=(i+a, j+b)$, $M_{id}.percept'=\delta_{\text{int,ad}}(M_{ad}.s)$	Transition	$M_{ad}.s'=\delta_{\text{int,id}}(M_{id}.percept)=(a',b')$
$s'=(a',b')$	Reaction	$percept'=\{\ldots, \text{available}_{+0,-1}, \ldots\}$
$\mathbb{R}_0^+ \cup \{\infty\}$	Frequency	$\mathbb{R}_0^+ \cup \{\infty\}$

Fig. 3. Representational seperation of a mover agent [9]

Moreover, each component exists or not on a specific space when there are multiple communication spaces (e.g., physical world and wireless networks). For example, [10] applies the feature of MultiPDEVS to separate an individual of a certain system into a brain component (i.e. pilot) for the decision making and a body component (i.e. executer) for the heuristic and physical actions as shown in Fig. 3. So, brain components exist and interact in an event-driven social space while the body components exists and interacts in a time-based physical space. This representational separation could help the conceptual structure of the target system similar to the viewpoint of stakeholders and engineers with the reuse of each partial component in a targeted domain. The reuses based on this representational separation has been illustrated in [10].

IV. DISCUSSION

To develop a proper product in the rapidly sophisticated maritime system, the system-level simulation has been becoming a promising solution through the product life cycle. Many researchers have developed co-simulation tools such as FMI, HLA/RTI, and Matlab Simulink$^{\text{TM}}$ to focus on the interface for the communication of different systems and operation for the synchronization of heterogeneous simulations without potential errors. However, there are still a few challenges to facilitate the practical applications of the system-level simulation for the product development in the maritime system [2].

The reusability of componential models and their structure is spotlighted in this paper. Theorists in system modeling and simulation have developed various alternatives to enhance the reusability of simulation models in terms of the clarity and readability of system specification [7]. One state-of-the-art modeling formalism with the modern view of systems theory is proposed to supplement the current co-simulation approaches in terms of better representing the system of systems. The introduction of the multicomponent formalism is expected to provide two benefits:

- Mathematical representation: this feature of MultiPDEVS allows for the concise expressiveness and complete representation for atomic models and their interactions rather than verbose description such as the ODD protocol [11]. In other words, the componential systems and interactions of a certain system could be more systematically defined and explicitly specified.
- Representational separation: this feature of MultiPDEVS enables to differentiate atomic models and their interactions corresponding to communication spaces in the viewpoint of stakeholders and engineers. For instance, the simulation-based test of a gateway device using integrated ship-to-shore communications requires multiple communication networks (i.e., 5G, LTE, and etc.) on the simulation. System engineers could develop the several communication spaces with the representational separation of MultiPDEVS.

In recent years, a few projects (e.g., Open simulation platform [12]) foster meaningful achievements to develop and verify a system in the higher complexity of maritime system while the relatively less attention on the reusability in terms of human engineers has been paid. Notwithstanding the limited verification, this study notices the importance of the readability as well as systematic error to supplement the reusability of simulation models. In this sense, this study is expected to be a cornerstone for the better practical applications of the simulation-based approach in the maritime system.

ACKNOWLEDGMENT

This research was supported by Korea Institute of Maritime Science & Technology Promotion(KIMST) funded by the Ministry of Oceans and Fisheries, Korea(20220534, The Development of Simulation-based Evaluation Technology).

REFERENCES

[1] D. Yang, L. Wu, S. Wang, H. Jia, and K. X. Li, "How big data enriches maritime research–a critical review of Automatic Identification System (AIS) data applications," Transp. Rev., vol. 39(6), pp.755-773. 2019.

[2] N. Taylor, C. Human, K. Kruger, A. bekker, and A. Basson, "Comparison of Digital Twin Development in Manufacturing and Maritime Domains," in Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future: Proceedings of SOHOMA 2019, T. Borangiu, D. Trentesaux, P. Leitão, A. G. Boggino, amd V. Botti, V, Eds. Springer, 2019, pp.158-170.

[3] FMI Association, "Functional Mock-up Interface Specification," available at: https://fmi-standard.org/ (accessed: Aug. 14th, 2023)

[4] B. Möller, K. L. Morse, M. Lightner, R. Little, and R. Lutz. "HLA evolved–a summary of major technical improvements," In Proceedings of Spring Simulation Interoperability Workshop, 08F-SIW-064, 2008.

[5] A. Falcone and A. Garro, "Distributed Co-Simulation of Complex Engineered Systems by Combining the High Level Architecture and Functional Mock-up Interface," Simul. Model. Pract. Theory, vol. 97, p. 101967, 2019.

[6] F. Perabo, D. Park, M. K. Zadeh, Ø. Smogeli, and L. Jamt, "Digital twin modelling of ship power and propulsion systems: Application of the open simulation platform (osp)," In 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), pp. 1265-1270, June 2020.

[7] B. P. Zeigler, "DEVS and MBSE: a review," Int. J. Model. Simul. Sci. Comput., vol. 13(02), 2230001, 2022.

[8] D. Foures, R. Franceschini, P. A. Bisgambiglia, and B. P. Zeigler, "multiPDEVS: A Parallel Multicomponent System Specification Formalism," Complexity, vol. 2018, 3751917, 2018.

[9] A. Steiniger, F. Krüger, and A. M. Uhrmacher, "Modeling agents and their environment in multi-level-DEVS," In Proceedings of the 2012 Winter Simulation Conference, pp. 1-12. December 2012.

[10] J. Kim, "Formal Modeling and Simulation of Availability-based Agent Behavior: Case Studies of Human Movement and Imitation," UNIST, 2021.

[11] V. Grimm et al., "The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism," J. Artif. Soc. Soc. Simul., vol. 23(2), 2020.

[12] Open Simulation Platform, "Open Simulation Platform," available at: https://opensimulationplatform.com/ (accessed: Aug. 14th, 2023).