System of Systems Approach to Formal Modeling of CPS for Simulation-Based Analysis

Kyou Ho Lee, Jeong Hee Hong, and Tag Gon Kim

This paper presents a system-of-systems (SoS) approach to the formal modeling of a cyber-physical system (CPS) for simulation-based analysis. The approach is based on a convergence technology for modeling and simulation of a highly complex system in which SoS modeling methodology, hybrid systems modeling theory, and simulation interoperation technology are merged. The methodology maps each constituent system of a CPS to a disparate model of either continuous or discrete types. The theory employs two formalisms for modeling of the two model types with formal specification of interfaces between them. Finally, the technology adapts a simulation bus called DEVS BUS whose protocol synchronizes time and exchange messages between subsystems simulation. Benefits of the approach include reusability of simulation models and environments, and simulation-based analysis of subsystems of a CPS in an inter-relational manner.

Keywords: Cyber-physical system, CPS, system of systems, hybrid system, formal model, simulation-based analysis.

Manuscript received July 16, 2014; revised Oct. 1, 2014; accepted Nov. 21, 2014.

This work was supported by the Inje Research and Scholarship Foundation in 2011.

Kyou Ho Lee (corresponding author, kyou@inje.ac.kr) is with HSV-TRC, UHRC, and the Department of Information and Communications Engineering, Inje University, Gimhae, Rep. of Korea.

Jeong Hee Hong (jhhong@smslab.kaist.ac.kr) is with the Department of Industrial and Systems Engineering, KAIST, Daejeon, Rep. of Korea.

Tag Gon Kim (tkim@ee.kaist.ac.kr) is with the Department of Electrical Engineering, KAIST, Daejeon, Rep. of Korea.

I. Introduction

A cyber-physical system (CPS) is an integration of physical subsystems together with computing and networking [1]. It can be seen as an intelligent real-time distributed monitoring and control system with multiple feedback loops. Since disparate physical and cyber subsystems of a CPS are in operational independence, a CPS is a system of systems (SoS). From the point of view of system taxonomy, physical subsystems can be either continuous or discrete dynamic systems; computation subsystems are discrete dynamic systems. Thus, a CPS itself is a hybrid dynamic system. Since a CPS is a hybrid dynamic SoS, the analysis of its behavior and performance with either formal or informal models is highly complex. Being done by interaction between disparate subsystem models, a simulation-based technique may be more advantageous than formal methods in the analysis.

This paper presents an SoS approach to formal modeling of CPSs for simulation-based analysis. The approach first maps each subsystem of computation or physical process in a CPS to an independent model, which is similar to the SoS concept. Then, it provides an interface between component models for their interaction. Thus, a set of component models in either continuous or discrete dynamics along with their interfaces constitutes a hybrid model. To be unified in a model representation of continuous and discrete dynamic models, we employ the system-theoretic view in the specifications of both models. More specifically, the view represents a system model in three sets (namely, input, output, and state) and associated operations (namely, state transition function/equation, and output function/equation).

With the proposed modeling approach, simulation of the hybrid model would be done by interoperation between simulators, each being associated with a disparate component model. Each simulator interprets formal semantics for the model. Interoperation between simulators can be performed via DEVS BUS [2], which may be implemented in a standard simulation middleware of HLA/RTI [3]–[5]. The main advantage of simulation interoperation is reusability of modeling and simulation (M&S) environments for continuous and discrete event dynamic systems. Moreover, analysis of CPSs in simulation interoperation would be done based on independent subsystems. Indices for a computation model are a function of the variables contained within, with variable parameters in the physical process model; and vice versa. We call such an analysis a joint analysis, because indices for the computation and physical process models are related to each other; thus, they should be analyzed in a joint manner.

The remainder of this paper is organized as follows. The next section presents types of systems and modeling formalisms. Section III describes the simulation-based analysis of complex systems such as CPSs. Sections IV, V, and VI present our approach to the formal modeling of a CPS as an SoS and associated concepts of simulation interoperation using DEVS BUS. A case study is shown in Section VII. Finally, a conclusion is drawn in Section VIII.

II. Types of Systems and Modeling Formalisms

Systems M&S processes vary according to the types of systems and modeling objectives. The state of a system can be either continuous or discrete — a factor that can change over time. Table 1 shows the system-theoretic view of the input, state, and output of a system with continuous, discrete, and hybrid dynamics.

A continuous (dynamic) system is one that is operating in continuous time and in which its input, state, and output variables are all real values. Examples of a continuous system (CS) include analog circuits/systems, vehicle dynamics, and continuous physical processes in a CPS such as variation of temperature.

A discrete (dynamic) system is one that changes its state in a piecewise constant manner by either time-based or event-based time advances. Input, state, and output variables of the system are all discrete values. Viewing time-based advances as a special case of event-based advances, we classify a discrete system as a discrete event (dynamic) system. Examples of a discrete system include digital systems for time-based, communication networks for event-based, and CPS computation at the H/W level (that is, digital system) for time-based and at the S/W level (that is, algorithm) for event-based. A hybrid (dynamic) system is a combination of continuous and discrete (dynamic) systems. A typical example of a hybrid system includes a CPS in which a computation subsystem is a DES and a physical subsystem is a CS. Of course, a physical subsystem of a CPS may be a DES, such as a flexible manufacturing process. The physical process in combination with the computation of intelligent decision-making, constitutes a CPS that can be classed as a DES.

The representation of systems may be formal or informal. Informal models depend on a modeler's views and experiences of systems. One of the main problems of informal modeling is the lack of mathematical semantics. Thus, such models should not be employed in the modeling of complex systems such as CPSs. On the other hand, formal modeling should be based on sound semantics, called formalisms, to specify the model's behavior without incompleteness or ambiguity.

The process of modeling formalisms in CS and DES differs depending on the type of system. This is due to the heterogeneous types of input, output, and state variables of the

Table 1. Types of systems: CS, discrete event system (DES), and hybrid system (HS = CS + DES).

			Input	State	Output	Example	Modeling formalism
HS (CS+ DES)	CS					Continuous physical process (e.g. variation of temperature)	Differential equation
	DES	Time-based				Computation at H/W level	See Fig. 1
		Event-based				Computation at S/W level	

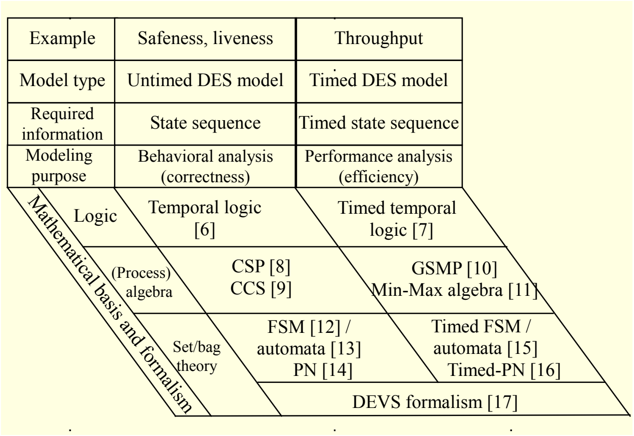

Fig. 1. Math formalisms for DES modeling [communication sequential process (CSP), calculus of communicating system (CCS), generalized semi-Markov process (GSMP), finite state machine (FSM), and petri net (PN)].

http://dx.doi.org/10.4218/etrij.15.0114.0863

two types of systems. Although modeling formalisms for CS relies on differential equations, a variety of formalisms have been employed in DES modeling. This is mainly because each formalism has its own semantics, which is convenient for modelers to express their domain-specific system to be modeled. Furthermore, such semantics are based on different mathematical bases. Figure 1 summaries such formalisms along with their mathematical bases.

III. Simulation-Based Analysis of Complex Systems

Analysis of complex systems, such as CPSs, should be performed for the whole life cycle of a system, ranging from the design phase to the operational/maintenance phase. The purpose of such analysis should be design/correctness verification and performance/effectiveness evaluation. In general, a system model for behavioral analysis and that for performance analysis do not have to be the same. Accordingly, formal models for the two may be different, as is also shown in Fig. 1. Note that behavioral analysis with formal models should be done either by formal methods or by simulation, but performance analysis should be done mainly by simulation.

The main advantage of formal methods, such as the checking of models [18]–[19] and the proving of theorems [20]–[21], for behavior analysis is completeness — in the sense of the formal methods' coverage of the verified state trajectories of a system model. However, formal methods are not practically applicable to a large-scale SOS such as a CPS, due to the cost in both time and space for a complete exploration of the state space of the system model. It is known that formal methods are applied to the correctness/design verification of hardware (in most advanced industries) and moderately sized software (in academia). Moreover, formal methods have limitations in performance analysis, for which simulation is the most suited. This is why the proposed method employs simulation-based analysis as opposed to formal methods.

Simulation is a process to execute a formal model, in which a predefined input scenario is given and a corresponding output is observed. Thus, a range of input scenarios is limited to the coverage of state trajectories of the model under simulation. Simulation of a formal model requires simulation algorithms, or simulation engines, which may not be unique for a given formal model. A main function of any simulation engine is to interpret the semantics of a model, which is divided into three subfunctions — namely, execution of each component (model), time synchronization, and data exchange between components. Thus, each formal model has to have its own simulation engine to interpret the semantics specific to the model.

Consider a CPS model that has a collection of component models in different semantics, which is very natural in CPS modeling. Then, CPS simulation would require a collection of local simulation engines, whereby each of which would have to interpret an associated component model. Moreover, the simulation needs to coordinate all of the local simulation engines, for which time synchronization and data exchange between them are necessary [22]. Alternatively, a CPS model may be specified by a unified set of semantics, which is applicable to all component models. In such a case, a simulation engine may be enough to interpret the semantics [23]. The latter is a single-system M&S approach; the former is an SoS M&S approach, which will be discussed in the next section.

IV. Convergence Technology for CPS M&S

1. CPS as Hybrid SoS

A CPS has two basic components: physical process and computation. Figure 2 shows various logical coupling structures of such components to form a CPS. A CPS implementation would map such structures in a physical network in an appropriate manner. Figure 3 shows one such mapping of a CPS in a network-centric SoS. Such a CPS interconnects CPSs and components of CPSs via networks, each of which is considered to be independent and disparate, yet they should work together to achieve a common goal [24]–[27]. Note that computations and physical processes may be either centralized or distributed.

A CPS is an HS with diverse subsystems of heterogeneous type [28]. Figure 4 shows the types of such subsystems. Formal modeling of these subsystems requires clear, well-defined semantics that are specialized enough to represent their behavior. Whereas differential equations are used for CS modeling, various formalisms are in existence for DES modeling (see Fig 1). Formalisms for the modeling of an HS should have sound semantics to specify both CS and DES dynamics, and interfaces between them.

Several efforts have been made in the field of modeling and simulation of HS. Basically, two approaches are possible from the point of view of a number of formalisms used in modeling. Figure 5 summarizes the concepts and features of the two

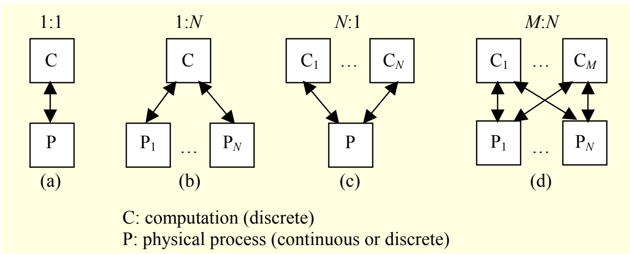

Fig. 2. Coupling structure of C and P in CPS.

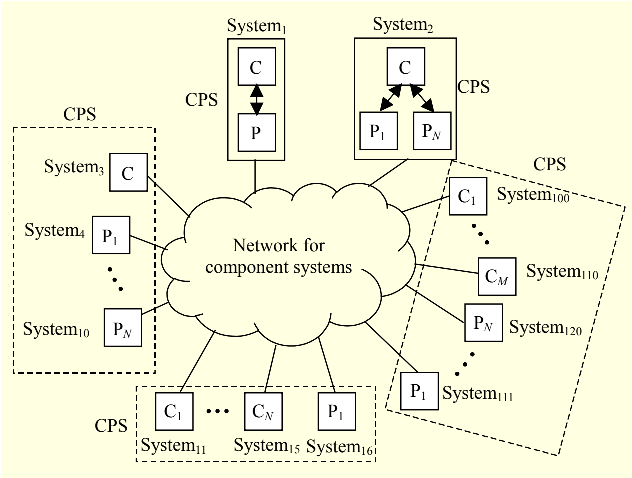

Fig. 3. CPS as network-centric SoS.

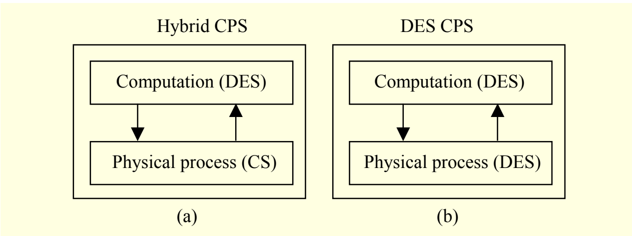

Fig. 4. CPS system types in modeling view: (a) hybrid CPS and (b) DES CPS.

approaches. The first approach employs a formalism that represents a mixture of continuous and discrete event dynamics [29]–[31]. As shown in Fig. 5(a), this approach should have a new tool to support modeling and simulation of HSs using the formalism. Examples of this particular approach with associated tools include MATLAB/Simulink, Ptolemy II [32], AnyLogic [33], and PowerDEVS [34]. One of the main hurdles for domain-specific simulation practitioners wishing to use this approach is the difficulty in understanding the semantics for the formalism.

The remaining approach employs two formalisms with interfaces between them — one for continuous models, and the other for discrete event models (as shown in Fig. 5(b)). Unlike the first approach, this approach aims to explicitly separate different types of dynamics in different formalisms. However, a simulation approach may not be unique. One approach simulates continuous and discrete event models with appropriate interfaces in an environment [35]–[36]. The other approach simulates the two models in a set of interoperable environments in which each model is simulated by its own environment and is capable of communicating with the other model [37]–[38]. Note that the former reuses only models and that the latter reuses both models and simulation environments. Of course, both approaches need to define formal semantics for interfaces, which will be presented later.

2. Joint Analysis of CPS Subsystems

Constituent systems of CPSs perform disparate functions of their own to achieve a given CPS goal [39]–[40]. Thus, analysis of CPSs should be done in such a way that a computation system is analyzed as a function of a physical system, and vice versa. We call such an analysis a joint analysis for an SoS, as opposed to a single analysis for systems [41]. Figure 6 shows the differences between single analysis and joint analysis with a modeling and simulation process. Associated with single analysis is a conventional modeling and simulation approach in which a modeler views a complex system as one (Approach I in Fig. 6). In the approach, an analysis should be carried out either for computation or for physical process.

On the other hand, joint analysis employs a set of disparate simulations, each of which simulates its own constituent system in an independent manner. Thus, an appropriate experimental design for simulation allows modelers to measure analysis indices for one subsystem model with the parameters of another. For example, the performance of a computation (or physical process) subsystem is measured by simulating a computation model while simulating a physical process (or computation) model with various parameters (Approach II in

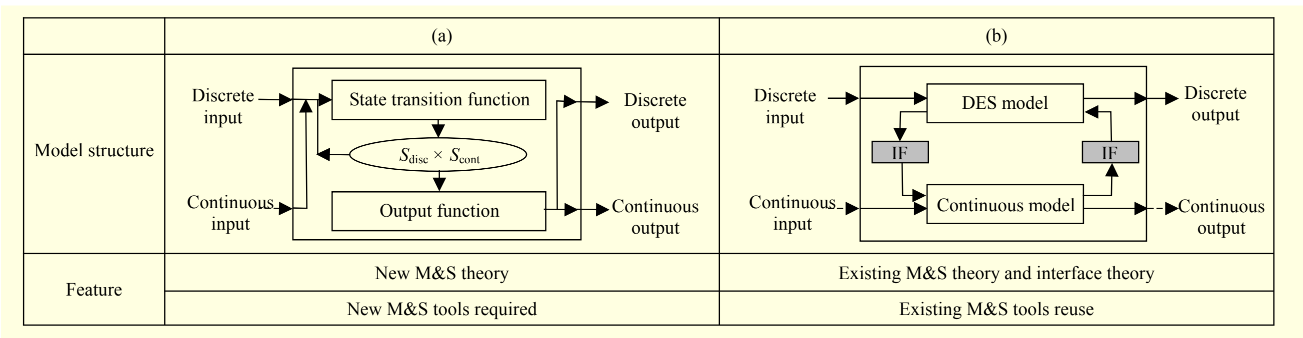

Fig. 5. HS modeling approaches: (a) use of a unified formalism and (b) use of two formalisms.

http://dx.doi.org/10.4218/etrij.15.0114.0863

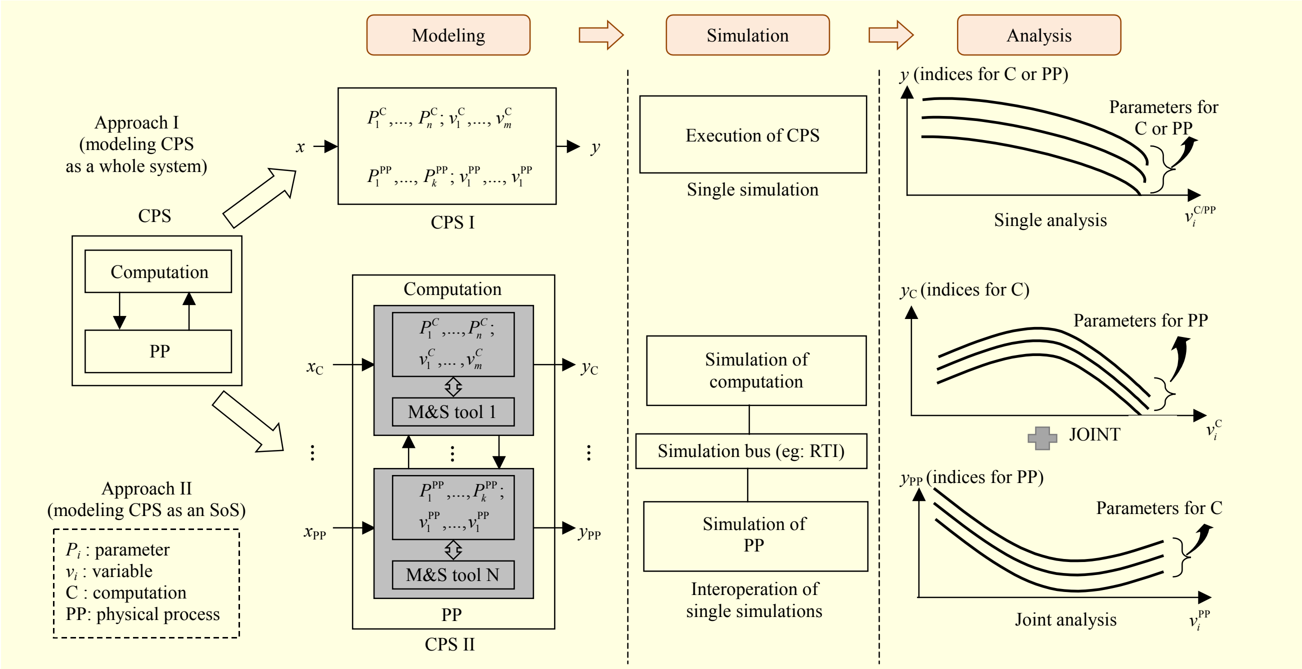

Fig. 6. Analysis view of CPS: single analysis vs. joint analysis (taken from [41]).

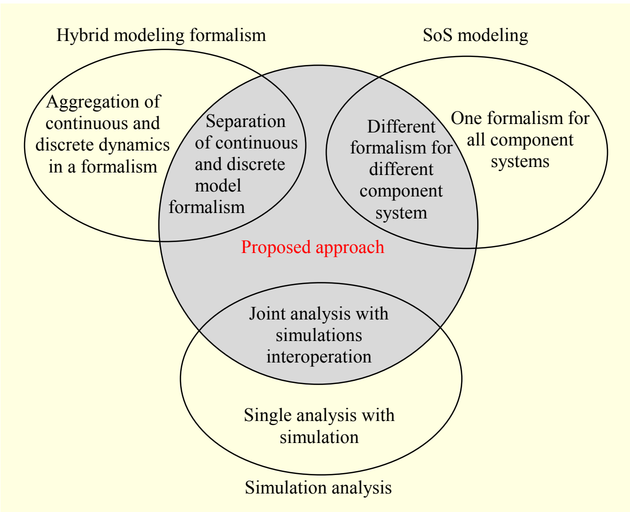

Fig. 7. Proposed convergence approach to CPS M&S.

Fig. 6). Joint analysis is effective when one is interested in the analysis of a constituent system with a function of parameters for another constituent system [42].

3. Overview of Proposed Approach

We are now ready to propose our approach to CPS modeling for simulation-based analysis. As described earlier, CPS modeling is closely related to such perspectives as SoS modeling, hybrid modeling formalism, and joint analysis. Although each of these three perspectives has been studied rigorously in its field, requirements for CPS modeling cannot be attained by one perspective alone. Taking the three perspectives into consideration, we propose an approach to CPS modeling (see Fig. 7).

As shown in the figure, our approach in SoS modeling is to employ different formalisms for the modeling of disparate, constituent systems. Accordingly, our approach to hybrid modeling is explicit separation of continuous and discrete event models with sound interfaces. Finally, simulation of hybrid models is done by simulations interoperation in which continuous models, and that for discrete event models, are simulated in their own environment. The M&S approach allows one to perform a joint analysis of a CPS in which the analysis of subsystems is done inter-relationally.

V. Formal Specification of CPS Model

The proposed approach to the formal modeling of a CPS requires mathematical semantics for continuous models and discrete event models, as well as coupling schemes between them. Figure 8 shows a system-theoretic approach to the modeling of a system where continuous and discrete event dynamics are modeled in a unified view. The view is unified in that a model is represented by three sets (namely, input, output, and state) and operations on them (called state equation and output equation). Once a model is defined by the three sets and their respective associated operations, it should be refined to appropriately represent continuous and discrete event dynamics.

More specifically, the types of input, output, and state sets for a continuous model are all given in terms of real values; those for a discrete event model are all discrete (ranged) values. Accordingly, operations on the three sets for a continuous model are specified by a differential equation; those for a discrete event model are specified by a DEVS equation. Note that in Fig. 8 the state equation (2) in the DEVS equation for a discrete event model is a quantized form of that of the differential equation (1) in a continuous model. Likewise, output equations for the differential equation and the DEVS equation have the same form. Although Fig. 8 shows linear state/output equations for a CS, it may have nonlinear dynamics represented by nonlinear differential equations.

Note that the state equation (1) of a linear differential equation model has the form $\mathrm{d}Q/\mathrm{d}t = AQ(t) + BX$, where $\mathrm{d}Q/\mathrm{d}t = AQ(t)$ represents a state transition with no input and $\mathrm{d}Q/\mathrm{d}t = BX$ represents a state transition with an external input. Discrete forms corresponding to $\mathrm{d}Q/\mathrm{d}t = AQ(t)$ and $\mathrm{d}Q/\mathrm{d}t = BX$ are $q' = \delta_{\mathrm{int}}(q)$ and $q' = \delta_{\mathrm{ext}}(q, x)$, in the DEVS equations, respectively. Such a correspondence is natural because both the differential equation and DEVS equation are based on system-theoretic representation in system modeling. Recall that Fig. 1 listed a variety of formalisms for modeling DES. However, only DEVS formalism corresponds to the differential equation model in the system-theoretic state transition representation. Moreover, DEVS formalism should be applicable to both behavior and performance analysis using simulation.

DEVS formalism specifies DES in a hierarchical modular manner. The formalism has two classes — atomic model and coupled model. An atomic model represents state transition and output for a DES at a non-decomposable level. The DEVS equation shown in Fig. 8 is an equation form of an atomic model. A coupled model represents how models of atomic or coupled are coupled together to construct another model in a hierarchical form. The DEVS formalism itself is not the main topic of this paper. Details for the formalism can be found in [43].

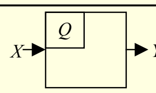	System theoretic model specification	
	State equation	Output equation
System representation - X: input - Y: output - Q: state	$\frac{\mathrm{d}Q}{\mathrm{d}t} = f(Q, X)$	$Y = g(Q, X)$
Differential equation for linear CS - X: real-values set - Y: real- values set - Q: real-values set	$\frac{\mathrm{d}Q}{\mathrm{d}t} = AQ + BX$ (1)	$Y = CQ + DX$
DEVS equation for DES - X: discrete events set - Y: discrete events set - Q: discrete events set	$q' = \delta_{\mathrm{int}}(q) \oplus \delta_{\mathrm{ext}}(q, x)$ (2)	$y = \lambda(q)$

Fig. 8. System-theoretic modeling of dynamic system.

An overall CPS model requires the specification of a coupling scheme that describes how one model is connected to the other. The scheme specifies a map from an output of a model to an input of another. The system coupling scheme (SCS) includes internal coupling (IC), external input coupling (EIC), and external output coupling (EOC) relations. These relations were adopted by DEVS coupled-model semantics, which do not consider the heterogeneity of input and output types.

Figure 9 shows the semantics for the interfaces for all possible cases to couple between a discrete event model and a continuous model.

We now present the proposed formalism for CPS modeling.

1. Formal Specification of CPS Model

$M_{\mathrm{CPS}} = \langle X, Y, M, \mathrm{SCS} \rangle$.
$X = X_{\mathrm{disc}} \cup X_{\mathrm{cont}}$: a set of hybrid inputs.
X_{disc}: a set of discrete event inputs.
X_{cont}: a set of continuous inputs.
$Y = Y_{\mathrm{disc}} \cup Y_{\mathrm{cont}}$: a set of hybrid outputs.
Y_{disc}: a set of discrete event outputs.
Y_{cont}: a set of continuous outputs.
$M = M_{\mathrm{DES}} \cup M_{\mathrm{CS}} \cup M_{\mathrm{HS}}$: a set of all component models.
M_{DES}: a set of discrete event models.
M_{CS}: a set of continuous models.
M_{HS}: a set of hybrid models.
$\mathrm{SCS} \subseteq (\mathrm{IC} \cup \mathrm{EIC} \cup \mathrm{EOC}) \times \mathrm{IF}$: system coupling scheme.
$\mathrm{IC} \subseteq \cup M_{\mathrm{i}}.Y \times \cup M_{\mathrm{j}}.X$: internal coupling relation.
$\mathrm{EIC} \subseteq (X_{\mathrm{disc}} \times \cup M_{\mathrm{DES_i}}.X) \cup (X_{\mathrm{cont}} \times \cup M_{\mathrm{CS_i}}.X) \cup (X \times \cup M_{\mathrm{HS_i}}.X)$: external input coupling relation.
$\mathrm{EOC} \subseteq (\cup M_{\mathrm{DES_i}}.Y \times Y_{\mathrm{disc}}) \cup (\cup M_{\mathrm{CS_i}}.Y \times Y_{\mathrm{cont}}) \cup (\cup M_{\mathrm{HS_i}}.Y \times Y)$: external output coupling relation.
$\mathrm{IF} \in \{f_{\mathrm{EE}}, f_{\mathrm{SS}}, f_{\mathrm{SE}}, f_{\mathrm{ES}}\}$: data conversion interface.

2. Formal Specification of Interface (IF)

$\mathrm{IF} \in \{f_{\mathrm{EE}}, f_{\mathrm{SS}}, f_{\mathrm{SE}}, f_{\mathrm{ES}}\}$.
$\Sigma = X_{\mathrm{disc}} \cup Y_{\mathrm{disc}}$: a set of discrete events.
Ω: a set of time segment functions (intput and output of M_{CS} in time interval).
f_{EE}: event to event interface.
f_{SS}: signal to signal interface.
f_{SE}: signal to event interface.
f_{ES}: event to signal interface.
See Fig. 9.

3. Formal Specification of CS (M_{CS})

$M_{\mathrm{CS}} = \langle X_{\mathrm{cont}}, Q_{\mathrm{cont}}, Y_{\mathrm{cont}}, \delta_{\mathrm{cont}}, \lambda_{\mathrm{cont}} \rangle$.
X_{cont}: a set of continuous inputs.
Q_{cont}: a set of continuous states.

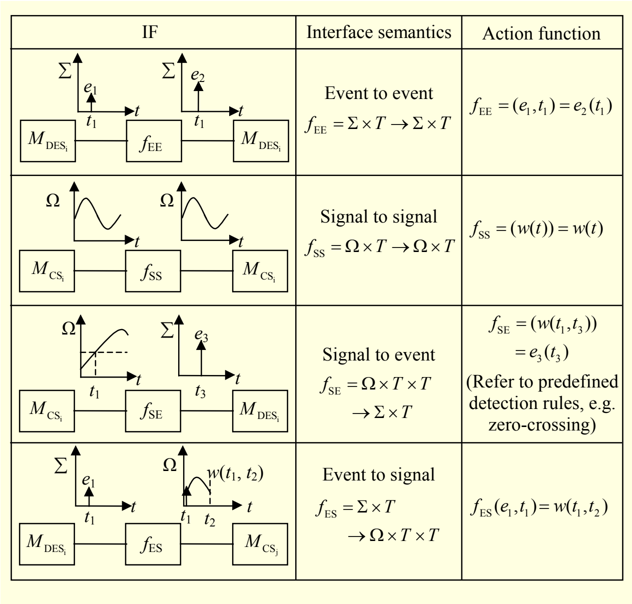

Fig. 9. Formal specification of data conversion interface.

Y_{cont}: a set of continuous outputs.

δ_{cont}: $\frac{d}{dt} Q_{cont}(t) = \delta_{cont}(Q_{cont}(t), X_{cont}(t), t)$: state transition function.

λ_{cont}: $Y_{cont} = \lambda_{cont}(Q_{cont}, X_{cont}, t)$: output function.

4. Formal Specification of DEVS Atomic Model (dAM of M_{DES}) [43]

dAM = <X_{disc}, S_{disc}, Y_{disc}, δ_{ext}, δ_{int}, λ, ta>.

X_{disc}: a set of discrete event inputs.

S_{disc}: a set of discrete event states.

Y_{disc}: a set of discrete event outputs.

δ_{ext}: $Q \times X_{disc} \rightarrow S_{disc}$: external transition function.

$Q = \{(s, e) \mid s \in S_{disc} \text{ and } 0 \le e \le ta(s)\}$

δ_{int}: $Q \rightarrow S_{disc}$: internal transition function.

λ: $Q \rightarrow Y_{disc}$: output function.

ta: $S_{disc} \rightarrow R^{+}_{0,\infty}$: time advance function.

5. Formal Specification of DEVS Coupled Model (dCM of M_{DES}) [43]

dCM = <X_{disc}, S_{disc}, M, EIC, EOC, IC, SELECT>.

X_{disc}: a set of discrete event inputs.

Y_{disc}: a set of discrete event outputs.

M: a set of all component models.

EIC $\subseteq X_{disc} \times \bigcup M_i.X$: external input coupling.

EOC $\subseteq \bigcup M_i.Y \times Y_{disc}$: external output coupling.

IC $\subseteq \bigcup M_i.Y \times \bigcup M_j.X$: internal coupling.

SELECT: $2^M - \varnothing \rightarrow M$: tie-breaking function.

VI. DEVS BUS for Simulation Interoperation

As shown in Fig. 10, our approach employs the interoperation of simulations, each of which simulates a constituent model corresponding to a disparate CPS subsystem. Such interoperation requires data exchange and time synchronization between simulations. The DEVS BUS has been proposed to provide a common simulation infrastructure for the interoperation [2]. The DEVS BUS architecture, shown in Fig. 10, consists of a time synchronization bus controller and a data bus controller. Table 2 presents four messages; namely, (*, t), (done, t_N), (x, t), and (y, t) used in the DEVS BUS protocol.

The DEVS BUS protocol underlies the simulation algorithm, or simulator, of DEVS models [43]. However, the protocol is for interoperation between heterogeneous simulators, whereas the algorithm is for the simulation of DEVS models. The protocol employs (*, t) and (done, t_N) for time synchronization and (x, t) and (y, t) for message delivery between heterogeneous simulators. The time synchronization bus controller maintains the global simulation time of interoperation. The controller receives (done, t_N) from simulators and generates one (*, t) at a time. When receiving (*, t), a simulator updates its local simulation time by t. Consequently, the global causality constraint can be easily obtained. Messages between simulators pass only through the data bus controller in DEVS

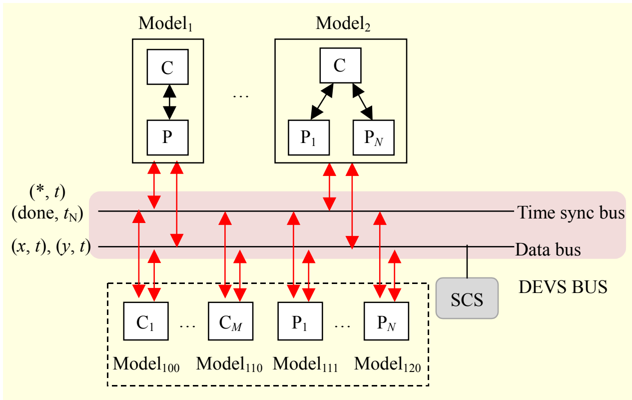

Fig. 10. DEVS BUS architecture (taken from [2]).

Table 2. Message type for DEVS BUS protocol.

Message	Implication
(*, t)	Time advance grant notification for the previous requested schedule reservation
(done, t_N)	Schedule reservation for the next (*, t)
(x, t)	Externally received input message at time t
(y, t)	Internally generated output message at time t

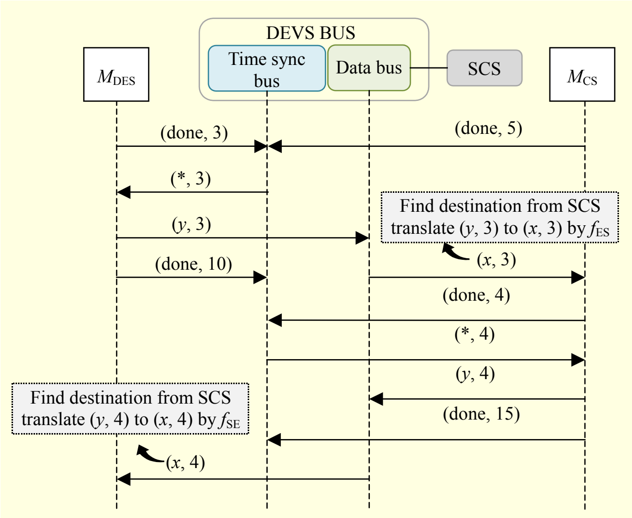

Fig. 11. Execution sequence of models in DEVS BUS.

BUS. A simulator that wants to send a message to another simulator is necessitated to send the message (y, t) to the data bus controller instead of to the destination simulator. Then, the data bus controller forwards (y, t) to the destination simulator as an input (x, t) by referring to the coupling scheme. The coupling scheme is a relation in which all pairs of source and destination simulators are specified. Such a coupling scheme is defined at the SCS of the proposed CPS modeling formalism. Figure 11 shows an execution sequence of the messages on DEVS BUS.

VII. Case Study: Defense CPS

1. Torpedo as CPS

A torpedo is a self-propelled underwater weapon carrying high explosives in its warhead. Being launched from submarines, warships, or aircrafts it tracks a target with its own search strategy. Our case study analyzes the dynamic behavior of an acoustic torpedo, which is launched from a submarine and homes in on the emissions of a target. The torpedo is mainly divided into two subsystems: a controller and a maneuver process. The controller, as a computation of the CPS, takes the role of a dynamic decision-maker under some uncertainty and tracks targets by its own algorithm. The dynamics of the maneuver process, as a physical process of the CPS, is represented by the continuous trajectory of the torpedo, which is controlled by the controller.

An objective of torpedo modeling is to analyze hybrid dynamics via simulation of a hybrid dynamic system model using the proposed approach. Figure 12 shows a simplified torpedo model. The controller model controls an elevation of the maneuver model with a feedback of a depth of the process to

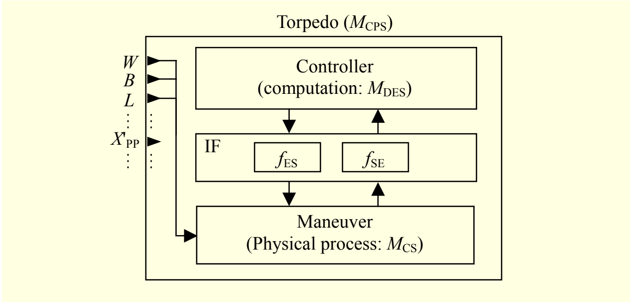

Fig. 12. Overall structure of torpedo CPS Model.

hit a target position. The maneuver model specifies the dynamic behavior of the torpedo over time. The model employs six state variables (u, v, w, p, q, r), which represent a velocity and an angular velocity of each of the x, y, and z axes. The state transition function δ_{cont} of the model is a differential equation based on the Newton equation and is employed with various parameters such as thrust force, gravity force, drag force, and so on [44]. The maneuver model delivers events to the controller model by the interface f_{SE} when crossing a predefined target path. Likewise, the controller model sends a control signal to the interface f_{ES} to guide the torpedo to the target. A formal specification of the torpedo model is given in the next subsection.

2. Model Formal Specification

A. Formal Specification of Torpedo Model (M_{CPS})

$M_{CPS} = \langle X, Y, M, SCS \rangle$.
$X = \{W, B, L, \ldots, X'_{PP}, \ldots\}$.
$Y = \varnothing$.
$M = M_{DES} \cup M_{CS}$.
M_{DES}: a controller model.
M_{CS}: a maneuver model.
SCS $\subseteq$ (IC $\cup$ EIC $\cup$ EOC) $\times$ IF: system coupling scheme.
IC = $\{(M_{DES}.E_{up}, M_{CS}.\delta_s), (M_{DES}.E_{down}, M_{CS}.\delta_s), (M_{CS}.Z, M_{DES}.E_{up}), (M_{CS}.Z, M_{DES}.E_{down})\}$.
EIC = $\{(W, M_{CS}.W), \ldots, (X'_{PP}, M_{CS}.X'_{PP}), \ldots\}$.
EOC = $\varnothing$.
IF = $\{f_{SE}, f_{ES}\}$.
$f_{SE}(Z(t)) = E_{up}(t)$ if $Z(t) < -15$ m.
$f_{SE}(Z(t)) = E_{down}(t)$ if $Z(t) > -15$ m.
$f_{ES}(E_{up}, t) = -\delta_s(t)$.
$f_{ES}(E_{down}, t) = \delta_s(t)$.

B. Formal Specification of Controller Model (M_{DES})

$M_{DES} = \langle X_{disc}, S_{disc}, Y_{disc}, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$.
$X_{disc} = \{E_{up}, E_{down}\}$.
$S_{disc} = \{$WAIT, UP, DOWN$\}$.
$Y_{disc} = \{E_{up}, E_{down}\}$.

http://dx.doi.org/10.4218/etrij.15.0114.0863

$\delta_{ext}(\text{WAIT}, E_{up}) = \text{UP}.$
$\delta_{ext}(\text{WAIT}, E_{down}) = \text{DOWN}.$
$\delta_{int}(\text{UP}) = \delta_{int}(\text{DOWN}) = \text{WAIT}.$
$\lambda(\text{UP}) = E_{up}.$
$\lambda(\text{DOWN}) = E_{down}.$
$\text{ta}(\text{WAIT}) = \infty.$
$\text{ta}(\text{UP}) = \text{ta}(\text{DOWN}) = 0.$

C. Formal Specification of Maneuver Model (M_{CS})

$M_{CS} = <X_{cont}, Q_{cont}, Y_{cont}, \delta_{cont}, \lambda_{cont}>.$
$X_{cont} = \{\text{RPM}, \delta_s, \delta_r, W, B, L, \ldots, X'_{PP}, \ldots\}.$
$Q_{cont} = (u(t), v(t), w(t), p(t), q(t), r(t)).$
$u(t)$: surge motion, $v(t)$: sway motion,
$w(t)$: heave motion, $p(t)$: roll motion,
$q(t)$: pitch motion, $r(t)$: yaw motion.
$Y_{cont} = \{X(t), Y(t), Z(t), \phi(t), \theta(t), \psi(t)\}.$

$$\delta_{cont}: \frac{d}{dt} Q_{cont}(t) = f(\text{RPM}, \delta_s, \delta_r, u, v, w, p, q, r, \phi, \psi, \theta).$$

$$\lambda_{cont}: \frac{d}{dt} X(t) = u(t)\cos\theta\cos\phi + v(t)(-\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi) + w(t)(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi).$$

$$\frac{d}{dt} Y(t) = u(t)\cos\theta\sin\phi + v(t)(\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi) + w(t)(-\sin\phi\sin\psi + \cos\phi\sin\theta\sin\psi).$$

$$\frac{d}{dt} Z(t) = -u(t)\sin\theta + v(t)\sin\phi\cos\theta + w(t)\cos\phi\cos\theta.$$

$$\frac{d}{dt} \phi(t) = p(t) + q(t)\sin\phi\tan\theta + r(t)\cos\phi\tan\theta.$$

$$\frac{d}{dt} \theta(t) = q(t)\cos\phi - r(t)\sin\phi.$$

$$\frac{d}{dt} \psi(t) = [q(t)\sin\phi + r(t)\cos\phi]/\cos\theta.$$

Refer to [44] for details of equations and coefficients.

3. Behavior Analysis of Torpedo CPS Model

The torpedo CSP model specified in the previous subsection is implemented and simulated. The torpedo is launched from a submarine and steered onto the target by controlling its position. When the depth of the maneuver model is located to be within 15 meters of the surface, a "down" event occurs and the controller model increases the elevation angle of the maneuver model. In contrast, when the torpedo sinks to below 15 meters deep, an "up" event is activated and the controller model controls the depth of the maneuver model by decreasing the elevation angle. The computation model (M_{DES}) and the physical process model (M_{CS}) of the torpedo CPS model (M_{CPS}) are implemented by DEVSim++ [45] and C++, respectively. Figure 13 shows the dynamic behavior of the torpedo CPS model during simulation.

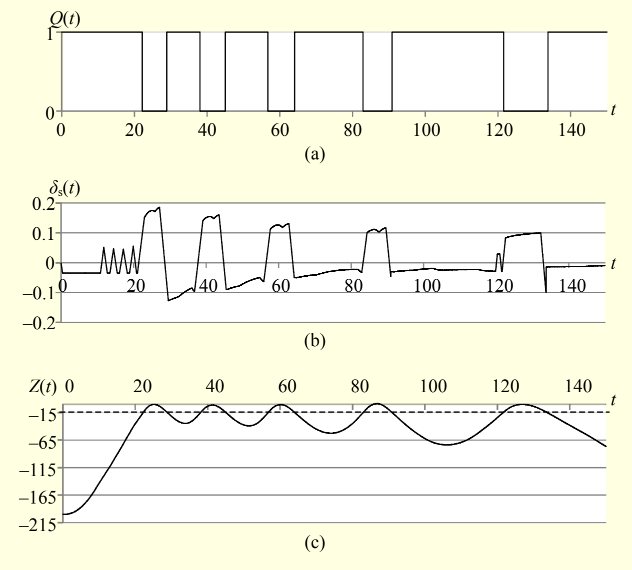

Fig. 13. Dynamic behavior of torpedo CPS model: (a) state transition of computation model (M_{DES}), (b) elevation angle ($\delta_s(t)$) applied to physical process model (M_{CS}), and (c) depth ($Z(t)$) of physical process model (M_{CS}).

VIII. Conclusion

A CPS consists of a collection of disparate subsystems of heterogeneous types. Since a CPS is a highly complex hybrid dynamic system of systems, simulation modeling is a practical means to analyze the behavior or performance of a CPS. The proposed formal modeling approach is a convergence technology in which concepts of SoS, modeling formalism of hybrid systems, and simulations interoperation are merged. The approach maps each subsystem of a CPS to an independent simulation model of either continuous or discrete event type, which should be simulated in a separate environment but interoperated together. Benefits of the approach include reusability of models and simulation environments/tools and analysis of subsystems in a flexible and inter-relational manner. The proposed approach would be still applicable if a non-DEVS formalism is used for the modeling of discrete event systems. In such a case, the formalism should support explicit input and output specifications to be compliant with the proposed interface specification. A simple case study shows an application of the proposed approach for modeling, simulation, and analysis of a CPS.

References

[1] E.A. Lee, "Cyber Physical Systems: Design Challenges," *IEEE*

Int. Symp., *Object Oriented Real-Time Distrib. Comput.*, Orlando, FL, USA, May 5–7, 2008, pp. 363–369.

[2] Y.J. Kim, J.H. Kim, and T.G. Kim, "Heterogeneous Simulation Framework Using DEVS BUS," *SIMULATION : Trans. Soc. Modeling Simulation Int.*, vol. 79, no. 1, Jan. 2003, pp. 3–18.

[3] IEEE Std. 1516-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Archit. (HLA) - Framework and Rules*, IEEE, New York, NY, USA, ISBN: 0-7381-2619-5, 2001.

[4] IEEE Std. 1516-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Archit. (HLA) - Federate Interface Specification*, IEEE, New York, NY, USA, ISBN: 0-7381-2621-7, 2001.

[5] IEEE Std. 1516-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Archit. (HLA) - Object Model Template (OMT) Specification*, IEEE, New York, NY, USA, ISBN: 0-7381-2623-3, 2001.

[6] Z. Manna and A. Pnueli, "*The Temporal Logic of Reactive and Concurrent Systems*," New York, NY, USA: Springer-Verlag New York, 1992.

[7] R. Koymans, "Specifying Real-Time Properties with Metric Temporal Logic," *Real-Time Syst.*, vol. 2, no. 4, Nov. 1990, pp. 255–299.

[8] C.A.R. Hoare, "*Communicating Sequential Processes*," Upper Saddle River, NJ, USA: Prentice Hall, 1985.

[9] R. Milner, "*Communication and Concurrency*," Upper Saddle River, Nj, USA: Prentice Hall, 1989.

[10] P.W. Glynn, "A GSMP Formalism for Discrete Event Systems," *Proc. IEEE*, vol. 77, no. 1, Jan. 1989, pp. 14–23.

[11] R. Cuninghame-Green, "*Minimax Algebra, Lecture Notes in Economics and Mathematical Systems 166*," New York, NY, USA: Springer-Verlag New York, 1979.

[12] A. Gill, "*Introduction to the Theory of Finite-State Machines*," New York, NY, USA: Mc-Graw Hill, 1962.

[13] Z. Kohavi, "*Switching and Finite Automata Theory*," 2nd ed., New York, NY, USA: McGraw-Hill, 1978.

[14] J.L. Peterson, "*Petri Net Theory and the Modeling of Systems*," Upper Saddle River, NJ, USA: Prentice Hall, June 1981.

[15] G. Noubir, D.R. Stephens, and P. Raja, "Specification of Timed Finite State Machine in Z for Distributed Real-Time Systems," *Proc. IEEE Workshop Future Trends Distrib. Comput. Syst.*, Lisbon, Portugal, Sept. 22–24, 1993, pp. 319–325.

[16] M.A. Holliday and M.K. Vernon, "A Generalized Timed Petri Net Model for Performance Analysis," *IEEE Trans. Softw. Eng.*, vol. 13, no. 12, 1987, pp. 1297–1310.

[17] A.I. Concepcion and B.F. Zeigler, "DEVS Formalism: A Framework for Hierarchical Model Development," *IEEE Trans. Softw. Eng.*, vol. 14, no. 2, 1988, pp. 228–241.

[18] E.M. Clarke, O. Grumberg, and D.A. Peled, "*Model Checking*," Cambridge, MA, USA: MIT Press, 1999.

[19] S. Owre et al., "PVS: Combining Specification, Proof Checking, and Model Checking," *Comput.-Aided Verification*, 1996, pp. 411–414.

[20] K. Havelund and N. Shankar, "Experiments in Theorem Proving and Model Checking for Protocol Verification," in *FME: Ind. Benefit Adv. Formal Methods*, Berlin, Germany: Springer Berlin Heidelberg, 1996, pp. 662–681.

[21] J. Rushby, "Theorem Proving for Verification," in *Modeling and Verification of Parallel Processes*, Berlin, Germany: Springer Berlin Heidelberg, 2001, pp. 39–57.

[22] J.-Y. Kim et al., "Abstracted CPS Model: A Model for Interworking between Physical System and Simulator for CPS Simulation (WIP)," *Proc. Symp. Theory Modeling Simulation - DEVS Integr. M&S Symp.*, SCS/ACM, Orlando, FL, USA, Mar. 26–29, 2012.

[23] D. Henriksson and H. Elmqvist, "Cyber-Physical Systems Modeling and Simulation with Modelica," *Proc. Int. Modelica Conf.*, Dresden, Germany, Mar. 2011, pp. 502–509.

[24] M. Jamshidi, "System of Systems Engineering - New Challenges for the 21st Century," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 23, no. 5, May 2008, pp. 4–19.

[25] J.E. Campbell et al., System of Systems Modeling and Analysis, SAND REPORT SAND2005-0020, Sandia National Laboratories, Jan. 2005.

[26] S.M. White, "Modeling a System of Systems to Analyze Requirements," *Annual IEEE Int. Syst. Conf.*, Vancouver, Canada, Mar. 23–26, 2009, pp. 83–89.

[27] P. Boily and N. Harrison, "A Simulation System of Systems to Assess Military Aircraft Protection," *IEEE Int. Syst. Conf.*, Vancouver, Canada, Mar. 19–22, 2012, pp. 1–6.

[28] B. Wang and J.S. Baras, "HybridSim: A Modeling and Co-simulation Toolchain for Cyber-Physical Systems," *Proc. IEEE/ACM Int. Symp. Distrib. Simulation Real Time Appl.*, Delft, Netherlands, Oct. 30–Nov. 1, 2013, pp. 33–40.

[29] P.J. Antsaklis and X.D. Koutsoukos, "Hybrid Systems: Review and Recent Progress," in *Softw.-Enabled Contr.: Inform. Technol. Dynamical Syst.*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2003, pp. 273–298.

[30] R. Goebel, R.G. Sanfelice, and A.R. Teel, "Hybrid Dynamical Systems," *IEEE Contr. Syst.*, vol. 29, no. 2, Apr. 2009, pp. 28–93.

[31] K.H. Johansson, J. Lygeros, and S. Sastry, "Modeling of Hybrid Systems," *Contr. Syst., Robot. Autom.*, vol. 15, UNESCO Encyclopedia of Life Support Systems (EOLSS).

[32] H. Zheng, "*Operational Semantics of Hybrid System*," Ph.D. dissertation, Department of EECS, University of California, Berkeley, CA, USA, 2007.

[33] A. Borshchev, Y. Karpov, and V. Kharitonov, "Distributed Simulation of Hybrid Systems with AnyLogic and HLA," *Future Generation Comput. Syst.*, vol. 18, no. 6, May 2002, pp. 829–839.

[34] E. Kofman, M. Lapadula, and E. Pagliero, "PowerDEVS: A DEVS-Based Environment for Hybrid System Modeling and

Simulation," School of Electronic Engineering, Universidad Nacional de Rosario, Tech. Rep. LSD0306, 2003.

[35] F. Bouchhima et al., "Generic Discrete-Continuous Simulation Model for Accurate Validation in Heterogeneous Systems Design," *Microelectron. J.*, vol. 38, no. 6–7, 2007, pp. 805–815.

[36] M. Wetter and P. Haves, "A Modular Building Controls Virtual Test Bed for the Integration of Heterogeneous Systems," in *SimBuild National Conf. IBPSA-USA*, Berkeley, CA, USA, July 2008, pp. 69–76.

[37] C. Sung and T.G. Kim, "Framework for Simulation of Hybrid Systems: Interoperation of Discrete Event and Continuous Simulators Using HLA/RTI," *IEEE Workshop Principles Adv. Distrib. Simulation*, Nice, France, June 14–17, 2011, pp. 1–8.

[38] S.J. Kwon et al., "Integrated Hybrid Systems Modeling and Simulation Methodology Based on HDEVS Formalism," *Proc. Summer Comput. Simulation Conf.*, Toronto, Canada, July 2013, pp. 410–417.

[39] M.W. Maier, "Architecting Principles for System of Systems," *Syst. Eng.*, vol. 1, no. 4, 1998, pp. 267–284.

[40] T.G. Kim, "Simulations Interoperation Approach for Modeling and Simulation of Defense System as System of Systems," *SpringSim*, *TMS Symp.*, *Plenary Talk*, San Diego, CA, USA, Apr. 7–10, 2013.

[41] T.G. Kim and D.S. Kim, "Joint Analysis of Combat Power and Communication System via Interoperation of War Game Simulator with Communication Network Simulator," presented at the *ROK-US Defense Anal. Seminar*, Seoul, Rep. of Korea, Apr. 23–25, 2012.

[42] J.H. Kim, I.-C. Moon, and T.G. Kim, "New Insight into Doctrine via Simulation Interoperation of Heterogeneous Levels of Models in Battle Experimentation," *SIMULATION : Trans. Soc. Modeling Simulation Int.*, vol. 88, no. 6, June 2012, pp. 649–667.

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

[44] T.I. Fossen, "*Guidance and Control of Ocean Vehicles*," New York, NY, USA: Wiley, 1994.

[45] T.G. Kim et al., "DEVSim++ Toolset for Defense Modeling and Simulation and Interoperation," *J. Defense Modeling Simulation: Appl.*, *Methodology*, *Technol.*, vol. 8, no. 3, 2011, pp. 129–142.

Kyou Ho Lee received his BS and MS degrees in electronics engineering from Kyungpook National University, Daegu, Rep. of Korea, in 1980 and 1982, respectively and his PhD degree in information and computer engineering from the University of Gent, Belgium, in 1998. From 1983 to 2005, he was a team leader and principal member of the research staff at ETRI, Daejeon, Rep. of Korea. He also worked as a researcher with AIT Inc., San Jose, CA, USA, from 1986 to 1988 and was a visiting scholar at the Department of Computer Science and Systems, University of Washington, Tacoma, USA, from 2011 to 2012. Since 2005, he joined Inje University, Gimhae, Rep. of Korea, as a full professor with the Department of Information and Communications Engineering. He is also a member of the High Safety Vehicle Core Technology Research Center and Ubiquitous Healthcare Research Center. His current research interests include cyber physical systems, variable structure systems, digital embedded systems, and ubiquitous and healthcare systems.

Jeong Hee Hong received her BS degree in electrical engineering from Pusan National University, Busan, Rep. of Korea, in 2005. She received her MS and PhD degrees in electrical engineering from KAIST, Daejeon, Rep. of Korea, in 2007 and 2013, respectively. She is currently a post-doctoral researcher with the Department of Industrial and Systems Engineering, KAIST. Her theoretic research focuses on systems engineering. In particular, she is interested in discrete event systems modeling and simulation; simulation-based optimization; and distributed systems development. Her practical research includes high-level architecture and large-scale system simulation, such as defense systems.

Tag Gon Kim received his PhD degree in computer engineering with specialization in systems M&S from the University of Arizona, Tucson, USA, in 1988. He was an assistant professor in electrical and computer engineering at the University of Kansas, Lawrence, USA, from 1989 to 1991. He joined the Department of Electrical Engineering, KAIST, Daejeon, Rep. of Korea, in the autumn of 1991 and has been a full professor with the EECS department since 1998. He was the president of the Korea Society for Simulation. He was the editor-in-chief for Simulation: Transactions for the Society for Computer Modeling and Simulation International. He is a co-author of the book Theory of Modeling and Simulation, Academic Press, 2000; the author of the edited book Artificial Intelligence and Simulation, Springer, 2004; and a co-author of the book Modeling and Simulating Command and Control, Springer, 2013. He has published about 200 papers in M&S theory and practice in international journals and conference proceedings. He is very active in defense M&S in the Rep. of Korea. He was, and still is, a consultant for defense M&S technology at various Korean government organizations, including the Ministry of Defense; Defense Agency for Technology and Quality; Korea Institute for Defense Analysis; and Agency for Defense Development. He is a fellow of the SCS and a senior member of the IEEE.