This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2907566, IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Model-based Design of Defense Cyber-Physical Systems to Analyze Mission Effectiveness and Network Performance

BONG GU KANG[1], KYUNG-MIN SEO[2], (Member, IEEE), and TAG GON KIM[3], (Senior Member, IEEE)

[1] Institute of Convergence Technology, KT, Seoul 06763, South Korea
[2] Naval and Energy System R&D Institute, Daewoo Shipbuilding and Marine Engineering (DSME), Seoul 04521, South Korea
[3] Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea

Corresponding author: Kyung-Min Seo (e-mail: kmseo.kumsung@gmail.com).

ABSTRACT This paper discusses two modeling and simulation (M&S) issues for defense cyber-physical systems (CPSs): 1) model-driven development via model interoperation and 2) simulation analysis between interoperated models. To clearly describe the network capabilities between the CPSs within network-centric warfare (NCW), we first modeled communication factors separately from the overall CPSs. The CPS models and the network model were considered discrete event systems, which were realized with DEVSim++ and ns-3, respectively. For simulation of the two types of models, we interoperated them with high-level architecture-based middleware to preserve the independence of each one. The goal for NCW analysis is to find a balance between the computational, physical, and communication factors of the defense CPSs to achieve their missions within a networked environment. In the case study of network-centric ground warfare, the CPS models generate mission effectiveness to be analyzed according to realistic network conditions; at the same time, the network model enables the analysis of the network performance given validated combat scenarios. We expect this study will help simulation experts interactively analyze computational and physical capabilities of the CPSs as well as communication effects between the CPSs.

INDEX TERMS Network-centric warfare; system of systems; discrete event system; model interoperation; linear regression analysis; measure of effectiveness; measure of performance

I. INTRODUCTION

A cyber-physical system (CPS) is an engineered system to be built from the synergy of computational and physical subsystems [1]. An autonomous unmanned vehicle, for a typical example of the CPS, decides with proper computations and it dynamically behaves according to the decision [2, 3]. For large-scale CPSs, communications between the CPSs as well as their inner subsystems should be evaluated, which results in networked CPSs [4] [5]. A wireless sensor network or mobile ad-hoc network (MANET) is used for information acquisition between the networked CPSs [6, 7].

In the military field, combat entities such as weapons, vehicles, or vessels are regarded as CPSs because they are physical objects with computational capabilities [8, 9]. Network-centric warfare (NCW) links many combat objects so that the corresponding forces receive information superiority and demonstrate a better situational awareness of the battlespace [10-12]. Hence, the NCW is seen as networked CPSs containing large-scale CPSs and a communication system for them [13]. For example, the military CPSs in the NCW are linked via a network architecture and carry out tactical operations regarding understanding situational awareness, making tactical decisions, and developing a course of action. For this reason, the NCW has been studied in terms of networked CPSs to evaluate a timely and robust information-sharing mission achievement [14, 15].

NCW analysis lies in the organic integration of multiple domains, i.e., the CPSs in the cyber-physical area and their communications in the information domain [16]. To this end, this study uses modeling and simulation (M&S) techniques, which have been widely utilized for designing and analyzing complex CPSs [17, 18]. Studying CPS models gives us insight into how they will behave in the real world [19]. We can experimentally analyze the operational and functional capabilities of the CPSs and experience how they are vulnerable to cyber attacks [20].

During the past decades, several defense CPS models have been proposed for NCW analysis. It is observed that most of them need two improvements regarding modeling and analysis aspects. For example, some studies have developed their models with an integrated approach rather than a system of system (SoS) approach. Such an enormous performance penalty prevents their usage on simulating CPSs of practical scale. Also, others had shortcomings for their methods or simulation results. They did not explain how operational scenarios and communication effects are affected interactively based on empirical simulation results.

This study presents model-driven development and analysis for the NCW with two motivated aspects: model interoperation and complementary analysis. As illustrated in Figure 1, we partition an overall NCW system into two independent models: a computer-generated force (CGF) model and a network model. Then the two models are interoperated by sustaining high modularity and localization [21, 22]. In specific, the CGF model focuses on constructive force-on-force simulation, which describes the outward appearances of military CPSs such as physical and computational activities [23]. The network model accounts for the deployment of the CPSs. It takes charge of transferring information between them considering communication effects such as transmission delay and data loss [24].

Another advantage of model interoperation resides in its complementary analysis. In other words, model interoperation allows a simulation analyst to design and execute overall simulation independently, which makes it possible to analyze individual simulations separately and interactively [8]. For example, the CGF model would compute mission effectiveness (y_{CGF} in Figure 1) with realistic communication effects (v_N) accounted for by the network model. Similarly, the network model would measure network performance (y_N) according to a validated military scenario (v_{CGF}) generated by the CGF model.

Because the developed models have different system taxonomies and unique characteristics, suitable simulation tools are required for model implementation or simulator development. In this study, the CGF model is realized with DEVSim++ [25]. It is an M&S tool based on the semantics of the discrete event systems specification (DEVS) formalism [26]. The network model is implemented using ns-3, which is an object-oriented simulator for communication networks [27]. For interoperable simulation, high-level architecture (HLA) has been widely used to interact with distributed heterogeneous simulators [28]; thus, we use HLA-based middleware for interoperable simulation between the CGF and the network simulators.

As a case study, we applied the proposed M&S development to network-centric ground warfare. In this application, mission effectiveness for the CGF simulator and network performance for the network simulator were analyzed on two levels. First, we macroscopically analyzed simulation results by changing all variables and parameters of the two simulators. Then we focused on specific experimental points of each simulator as minimizing the influences of the other simulator. Linear regression models were built to analyze computational and physical capabilities from the CGF simulator as well as communication effects from the network simulator. We expect that this study will provide support for M&S activities for networked CPSs.

This study is organized as follows. Section 2 presents our motivation, and Section 3 introduces related works and compares them. Section 4 proposes our M&S development for the networked CPSs. Section 5 explains and discusses an application for NCW analysis. Finally, Section 6 provides concluding remarks.

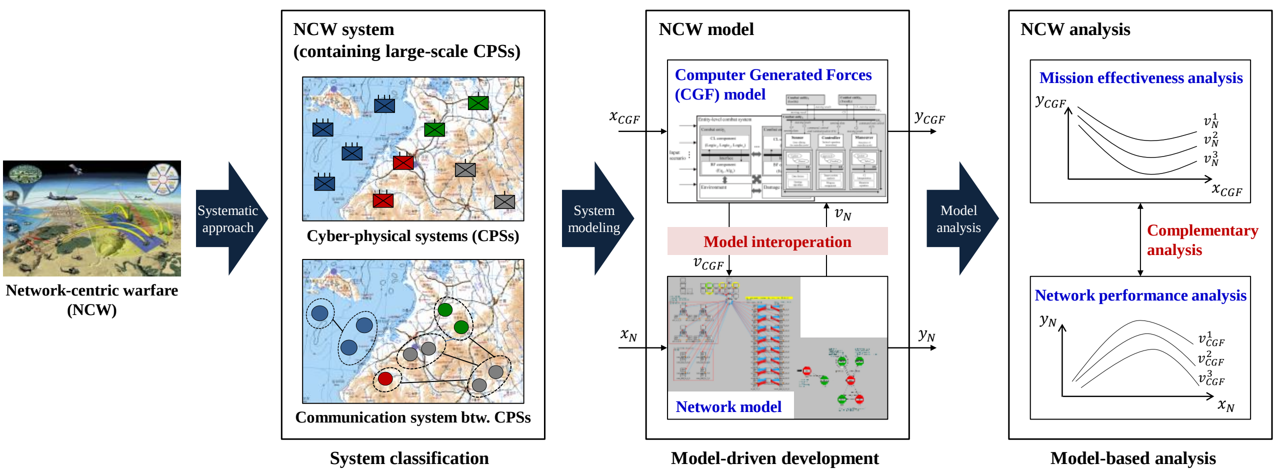

FIGURE 1. Overall proposed modeling and analysis of defense cyber-physical systems (CPSs) for network-centric warfare (NCW) analysis.

II. MOTIVATION

This section clarifies our motivation regarding two perspectives: modeling and analysis aspects.

A. MODELING ASPECT: SYSTEM VERSUS SYSTEM OF SYSTEMS

As explained in the introduction, combat objects in NCW are considered CPSs because they perform physical activities based on computational procedures. This study classifies the NCW system, or the networked CPSs, into a CGF system including the defense CPSs and their communication system. In the defense field, it is a type of command, control, and communication (C3) system.

Figure 2 illustrates two modeling methods for merging the two systems. The first approach is a conventional method in which a modeler views the system as a whole singular system. M_I in Figure 2 is the integrated model containing CGF and network models, which jointly share common information in an integrated form. On the other hand, in the case of M_{II} as the second approach, the modeler views the system as an SoS that is operationally and manageably independent between subsystems [29, 30].

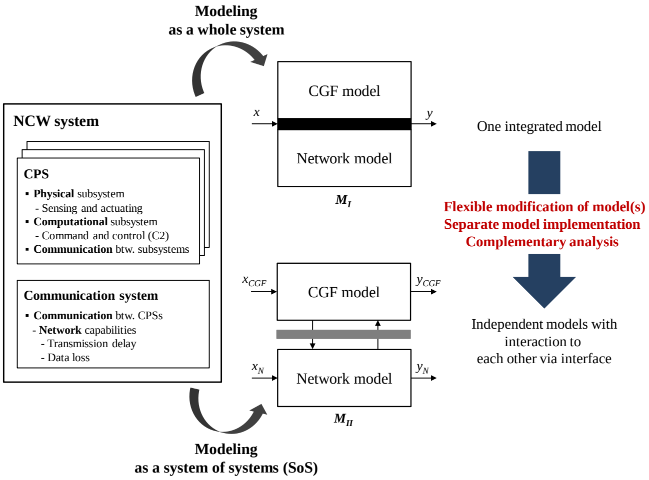

FIGURE 2. Modeling views for networked CPSs: Single system versus system of systems.

The main difference between the two modeling methods is the number of independent models: The integrated system features only one model, whereas the SoS features two or more models. The interoperation method facilitates each model in carrying out its tasks autonomously and independently [21]. Such independence allows the modeler to modify one model locally with all the other models left unchanged. Flexible modification enables evolutionary system development, which is a distinguishing characteristic of SoS development [31].

Also, independent models could be realized with a separate implementation framework to support their specifications. Because each model has a different system taxonomy or characteristic, the modeler should develop it with an appropriate simulation tool [21, 32]. This local modification and separate implementation would be a necessary condition for complementary analysis, which is explained in the following subsection.

B. ANALYSIS ASPECT: SINGLE ANALYSIS VERSUS COMPLEMENTARY ANALYSIS

The goal for NCW analysis is to find a balance between military factors to maximize the mission success rate of a friendly force within a networked environment [8]. In specific, the NCW analysis needs to analyze two indices of interest: mission effectiveness for the CGF model and network performance for the network model.

Figure 3 shows two analytic views for the NCW system: single analysis via integrated simulation and complementary analysis based on interoperable simulation. In the single analysis, indices of interest, which are either mission effectiveness or network performance, are influenced by a combination of CPSs' operations and network conditions all at once. Because all parameters and variables, i.e., p^{CPS}, v^{CPS}, p^N, and v^N, are dependable and mixed in the integrated model, it is difficult to classify, design, and find the factors that strongly influence the index.

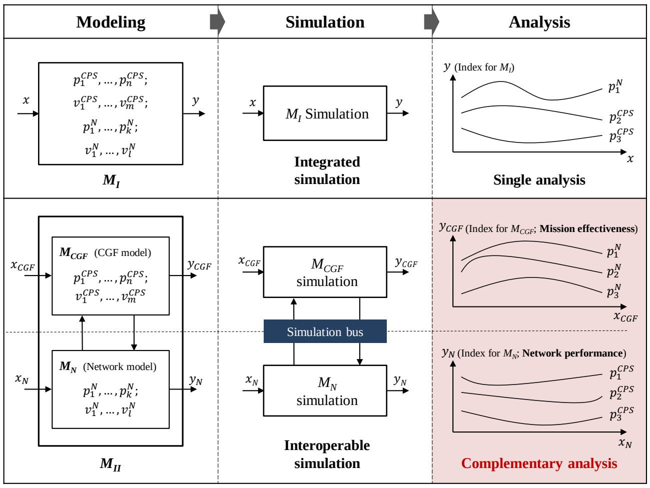

FIGURE 3. Analysis views for networked CPSs: Single analysis versus complementary analysis.

On the contrary, complementary analysis enables the analysis of indices of interest in an independent manner based on two or more simulations. In Figure 3, indices for the complementary analysis are partitioned into two simulations, i.e., y_{CGF} of M_{CGF} simulation and y_N of M_N simulation. Note that the independent analysis of y_{CGF} and y_N does not imply that their results are entirely isolated. Instead, these two results are affected by both interoperated models in an interactive manner. For example, y_{CGF} is influenced by not only the results of the M_{CGF} but also the variables and parameters in the M_N; and y_N is also affected by both variables and parameters in the

M_{CGF} and results of the M_N. Thus, the experimental design of the complementary analysis is much more flexible when changing variables and parameters belonging to each independent model.

III. LITERATURE REVIEW

For the past decades, several defense CPS models have been researched for NCW analysis. Table 1 summarizes their characteristics concerning the two motivations in our study, i.e., modeling and analysis aspects. The collective contribution of these studies is that they developed meaningful simulation systems in the military domain. Moreover, all the studies concentrated on constructive simulation for network-centric analysis. Because the network model is the most critical part of NCW simulation, the studies prominently featured physics-based network modeling at an engineering-level simulation.

Despite these contributions, these studies still warrant some improvements. From the modeling perspective, some studies developed their CPS models with an integrated approach rather than an SoS-based approach [33-36]. In other words, authors in these studies proposed a unified M&S framework to support network abilities and mission operations. Although their models are suitable for a centralized simulation environment, some weaknesses are inevitable, e.g., flexible modification of component models, separate model implementation, independent experimental design, and complementary analysis, those of which are described in the previous section.

Several researchers have studied SoS-based modeling, which is similar to our approach. For example, in Cayirci and Ersoy's work [37], military operational data were unilaterally delivered to a communication model for performance analysis of the communication system. Because of one-way transmission from operational to communication simulations, they could not analyze the mission effectiveness of the defense CPSs given the communication effects. On the other hand, Paz and Baer [38] combined a CGF and a communication simulator, which are HLA compliant for interoperable simulation. The roles of independent models in this study are nearly identical to our modeling roles.

Nevertheless, from an analytical point of view, most studies including Paz and Baer's research did not provide sufficient methods or empirical results for complementary analysis. One of their goals was NCW analysis; however, they mostly focused on how to model network systems and implement them. For example, Miner *et al.* [36] only provided a single analysis for tactical communication simulation in the integrated environment. Furthermore, although Paz and Baer [38] dealt with the NCW simulation from the perspective of SoS, they only expressed the interface between two simulations without experimental analysis. Most studies did not explain how operational scenarios and communication effects are affected interactively based on empirical simulation results.

To summarize, previous studies suffer from either inefficient M&S methods or insufficient simulation analysis despite satisfactory software development. They generally focused on specific parts between physical, computational, and communication systems. Because the NCW requires a balanced analysis between the three elements, the CPS concept will help the researchers to consider all the elements in an interactive and integrated manner. In this situation, the focus of this study is to make the best use of M&S for networked CPSs, to overcome these disadvantages, and to produce empirical simulation results.

TABLE 1. Comparison of previous studies on modeling and simulation (M&S) development for NCW system.

Previous work	Modeling perspective	Analysis perspective	Implication
Walsh *et al.* [33]	They proposed a net-centric end-to-end model in an integrated form.	They did not provide analysis methods or empirical simulation results.	They referred extensibility to interoperate with other campaign-level models.
Murphy and Flournoy [34]	They developed NCW software as a unified M&S framework.	They focused on introducing analysis tools and methods without providing empirical simulation results.	They referred interoperability to independent models as the future work.
Nam and Lee [35]	They proposed an integrated model containing CPSs and their communication.	They conducted analysis only from the perspective of the C2 related parameters.	They mentioned the importance of time-related factors such as communication.
Miner *et al.* [36]	They mentioned the necessity of SoS-based modeling for NCW analysis.	They focused on communication analysis.	They did not provide analysis of each system from the SoS perspective.
Cayirci and Ersoy [37]	They separated the NCW model into operational and communication models.	They centered on an analysis of communication effects by utilizing a database recorded by operational simulation.	They were not interested in an analysis of operational simulations.
Paz and Baer [38]	They used HLA-based interoperation between a CGF model and a communications model.	They could not provide analysis methods or empirical simulation results.	They referred extensibility to interoperation with other simulations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2907566, IEEE Access

IV. PROPOSED WORK

In this section, we describe the proposed M&S for networked CPSs. Overall M&S development, the design of two independent models, and model interoperation are explained here.

A. OVERALL M&S DEVELOPMENT

In conventional combat modeling, military CPSs have assumed sufficient communication between them. For example, some studies have considered the perfect delivery of information between combat models [39, 40]; others have used a simple connection probability to determine whether a transmission is successful [41, 42]. However, in reality, communication is based on radio transmission along line-of-sight paths or multipath propagation effects [43]. If the communication fails, detection information or orders are not shared normally, which may lead to severe mission failure. Therefore, M&S for NCW should consider realistic communication effects between the military CPSs as well as their operations and behaviors.

Figure 4 shows the overall M&S development for NCW analysis, which is classified into three phases. First, a CGF model (*I* in Figure 4) mainly contains CPS models to describe CPSs that participate in NCW. The CPS models fundamentally forward external messages to a communication agent (CA) model, and the CA model routes them to destination models if possible. For the communication system, the network model (*I*) covers network nodes according to the network topology. The nodes can be dynamically removed, added, or substituted by interacting with the relevant CGF model. As explained in the introduction, the CGF model is specified by the DEVS formalism, and the network model is described as the MANET.

Next, the decomposed models are realized with suitable M&S tools (*II*). For example, DEVS models can be implemented by utilizing an implementation framework supporting the DEVS formalism, such as DEVSimHLA [25], DEVSML [44], or CD++ [45]. The network model also has several commercial simulation tools to reflect realistic communication effects between the nodes. This separation of model design and implementation is a practical advantage of our SoS approach.

Finally, two simulators to be developed are interoperated based on a standardized ambassador or architecture (*III*). HLA, test and training enabling architecture (TENA), or service-oriented architecture (SOA) is a solution for model interoperation. In this study, we use a runtime infrastructure (RTI), which is software that supports the HLA [46].

B. CGF MODEL DESIGN AND IMPLEMENTATION

The DEVS formalism, which is a set-theoretic specification of discrete event systems, consists of atomic and coupled models [26]. The formalism is highly compatible with military systems organized hierarchically in a one-to-one manner. For example, a coupled model structurally guarantees modularity between component models through input/output (I/O) relations. An atomic model provides behavioral semantics, which is useful for expressing computational and physical activities. For these reasons, the formalism has been widely used for combat modeling [11, 47, 48]. Because DEVS models can be transformed into graph diagrams, we provide DEVS diagrams in Figures 5 and 7 [49].

Figure 5 illustrates a DEVS-coupled diagram for CGF modeling, whose DEVS notations are represented at the end of the figure. Fundamentally, the CGF model consists of several types of components models, that are either atomic or coupled models, and their connections. Inside the model, the CA model is connected with every CPS model according to two pairs of *IC (Internal Coupling)* relations: (*to_CA* of the CPS models, *from_CPS* of the CA model) and (*to_CPS* of the

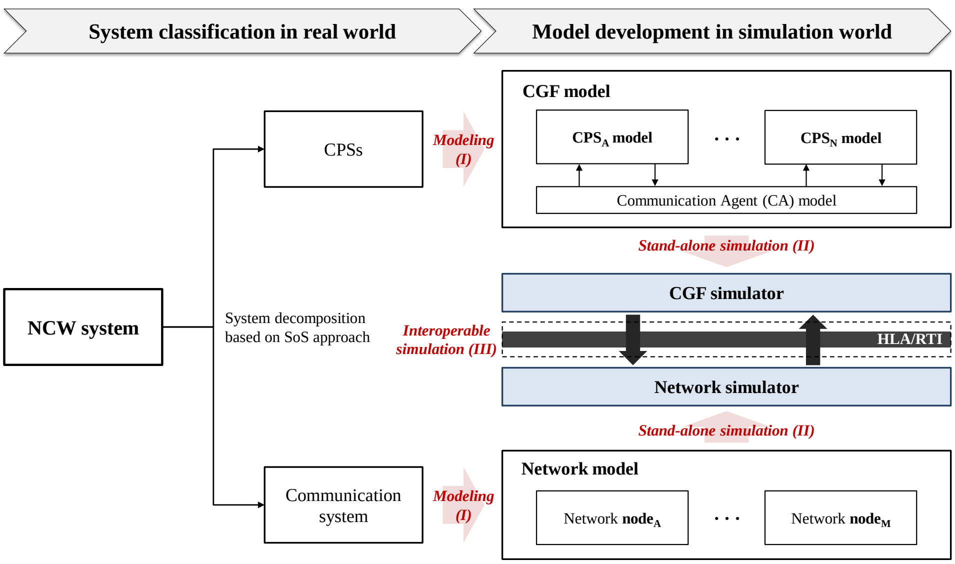

FIGURE 4. **Overall M&S development of NCW system.**

CA model, *from_CA* of the CPS models). Externally, two types of relations, i.e., the *EIC (External Internal Coupling)* and *EOC (External Output Coupling)*, are used to link between the outside and the inner CA model.

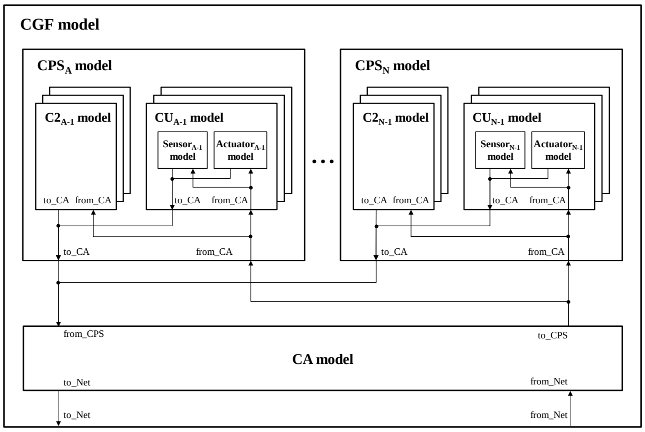

FIGURE 5. DEVS-coupled diagram of computer-generated force (CGF) model.

$$CGF = < X, Y, M, EIC, EOC, IC, Sel >$$

- $X = \{from\_Net\}$;
- $Y = \{to\_Net\}$;
- $M = \{CA\} \cup \bigcup_{i=1}^{n} \{CPS_i\}$;
- $EIC = \{(CGF.from\_Net,\ CA.from\_Net)\}$;
- $EOC = \{(CA.to\_Net,\ CGF.to\_Net)\}$;
- $IC = \{(\bigcup_{i=1}^{n} CPS_i.to\_CA,\ CA.from\_CPS),$
  $(CA.to\_CPS,\ \bigcup_{i=1}^{n} CPS_i.from\_CA)\}$;
- $Sel(\{CA, CPS_i\}) = CA$.

Similar to the CGF model, the CPS model is hierarchically structured with two sub-models, i.e., command and control (C2) atomic models and combat unit (CU) coupled models. The following notations represent the CPS and CU coupled models. The C2 model takes charge of military computational tasks, e.g., understanding of situational awareness, making tactical decisions, and developing courses of actions. The CU model, which includes actuator and sensor sub-models hierarchically, represents an outward form with physical effects. Note that indecomposable sub-models in a CPS model, i.e., C2, Actuator, and Sensor, have no direct connections between them, which means that the CPS and CU coupled models have no *IC* relations. The indirect connections are related to behaviors of the CA model, which will be explained with the CA model.

$$CPS = < X, Y, M, EIC, EOC, IC, sel >$$

- $X = \{from\_CA\}$;
- $Y = \{to\_CA\}$;
- $M = \bigcup_{i=1}^{n} \{C2_i\} \cup \bigcup_{k=1}^{m} \{CU_k\}$;
- $EIC = \{(CPS.from\_CA,\ \bigcup_{i=1}^{n} C2_i.from\_CA),$
  $(CPS.from\_CA,\ \bigcup_{k=1}^{m} CU_k.from\_CA)\}$;
- $EOC = \{(\bigcup_{i=1}^{n} C2_i.to\_CA, CPS.to\_CA),$
  $(\bigcup_{k=1}^{m} CU_k.to\_CA, CPS.to\_CA)\}$;
- $IC = \emptyset$;
- $Sel(\{C2_i, CU_k\}) = CU_k$.

$$CU = < X, Y, M, EIC, EOC, IC, Sel >$$

- $X = \{from\_CA\}$;
- $Y = \{to\_CA\}$;
- $M = \{Sensor, Actuator\}$;
- $EIC = \{(CU.from\_CA,\ Sensor.from\_CA),$
  $(CU.from\_CA,\ Actuator.from\_CA)\}$;
- $EOC = \{(Sensor.to\_CA, CU.to\_CA),$
  $(Actuator.to\_CA, CU.to\_CA)\}$;
- $IC = \emptyset$;
- $Sel(\{Sensor, Actuator\}) = Sensor$.

According to the type and resolution of the CPS model, it can contain multiple C2 or CU models. For example, if a CPS model expresses an aircraft as a single unit with low resolution, it is enough to design single C2 and CU models in the CPS model. On the other hand, if a CPS model represents a higher object to aggregate the units such as a company with ten tanks, a modeler can build several CU models for each tank within the CPS model. Finally, although a CPS model is unit level, if it has a high resolution, several CU or C2 models could be required to express the complexity of the model.

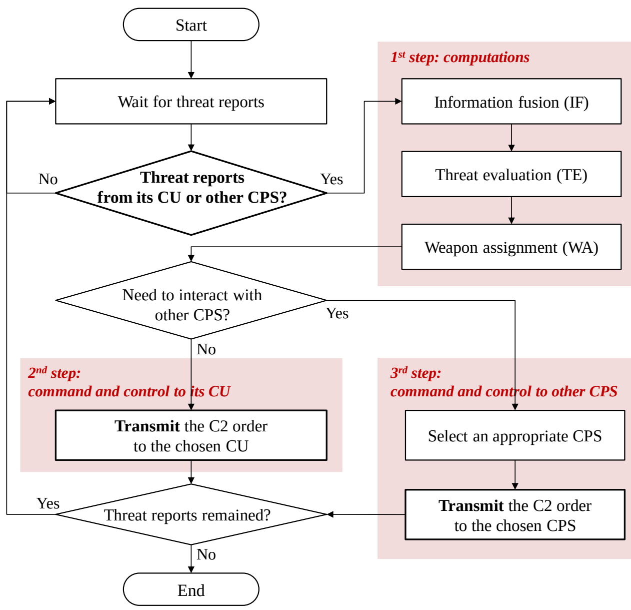

FIGURE 6. Example of computational procedures of top-level command and control (C2) model.

As well as the structure, the type and resolution of the CPS model also influence behaviors of the C2 and CU models. Figure 6 illustrates the computational process of the C2 model that is included in the top-layer CPS model. Based on threat reports from CU models in the same CPS model or other lower-layer CPS models, it fundamentally conducts the following C2 actions sequentially: 1) information fusion (IF),

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2907566, IEEE Access

2) threat evaluation (TE), and 3) weapon assignment (WA). After accomplishing these actions, the model determines whether to handle it internally or externally, i.e., whether to send the decision (the C2 order) to neighbor CU models or other CPS models. This computational process is recursively conducted until no targets are reported. Blocks with bold lines and fonts in Figure 6 denote I/O relations for the C2 model.

Figure 7 shows a DEVS atomic diagram to formalize the top-layer C2 model. The atomic model has three states for C2 activities, i.e., *IF*, *TE*, and *WA*, and one standby state, *WAIT*. When the model receives a report at the *WAIT* state, it changes the current state to the *IF* state and starts to conduct the C2 activities sequentially. Detailed actions or algorithms for the activities are described in *integrateInformation*(), *evaluateThreat*(), and *assignWeapon*(). In this model, mission accomplishment is influenced by 1) accurate decisions based on the C2 actions and 2) elapsed times for the decisions, i.e., elapsed times for state transitions from the *IF* to the *WA* state. Therefore, to analyze the mission effectiveness against various combat scenarios, a modeler can diversify time advance values of the *IF*, *TE*, and *WA* states and modify the algorithms. When reports are received at the *IF*, *TE*, and *WA* states, the C2 model stacks them and continues its current activities. If the model has no report, it stands by in the *WAIT* state until receiving a new report. The overall DEVS specifications are as follows:

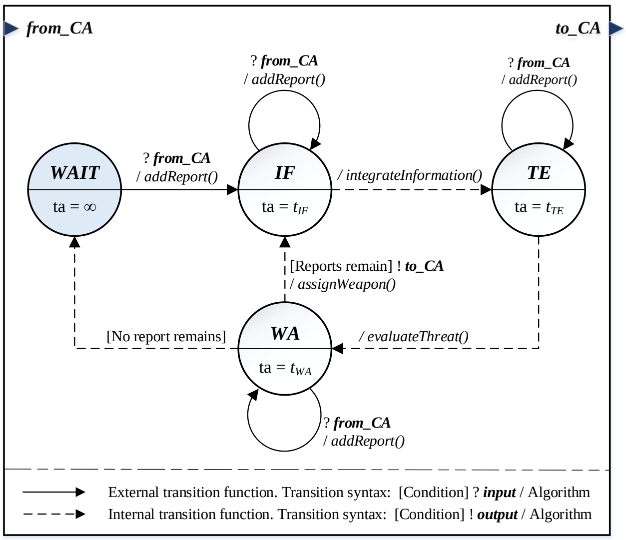

FIGURE 7. Discrete event systems specification (DEVS) atomic diagram of top-level C2 model.

$$C2_{Top} = < X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta >$$

- $X = \{from\_CA\}$;
- $Y = \{to\_CA\}$;
- $S = \{WAIT, IF, TE, WA\}$;
- $\delta_{ext}: (WAIT) \times (from\_CA) \rightarrow (IF)$ *and execute addReport(),*
  $(IF) \times (from\_CA) \rightarrow (IF)$ *and execute addReport(), refreshIFTime() ,*
  $(TE) \times (from\_CA) \rightarrow (TE)$ *and execute addReport(), refreshTETime(),*
  $WA \times (from\_CA) \rightarrow (WA)$ *and execute addReport(), refreshWATime();*
- $\delta_{int}: (IF) \rightarrow (TE)$ *and execute integrateInformation(),*
  $(TE) \rightarrow (WA)$ *and execute evaluateThreat(),*
  $(WA) \rightarrow (IF)$ *and execute assignWeapon() if the model has reports,*
  $(WA) \rightarrow (WAIT)$ *and execute assignWeapon() if the model has no reports;*
- $\lambda: (WA) \rightarrow (to\_CA)$;
- $ta: (WAIT) \rightarrow \infty$,
  $(IF) \rightarrow t_{IF}$,
  $(TE) \rightarrow t_{TE}$,
  $(WA) \rightarrow t_{WA}$.

Next, the CU-coupled model has two types of DEVS atomic models: a sensor and an actuator model. $\text{Sensor}_{\text{Radar}}$, as specified below, is an example of the search radar model. First, it has one operational state, i.e., *DETECT*, and one standby state, i.e., *WAIT*. The model is fundamentally activated and deactivated via an order received from a relevant C2 model. After receiving the order at the *WAIT* state initially, it changes to the *DETECT* state and detects threats during every t_{DETECT}. If threats are identified, the model sends the detection information as an output. Among detailed algorithms of $Sensor_{Top}$, *addThreat()* stores inputs regarding threat information, *refreshDetectTime()* updates t_{DETECT} when the model receives an input in the *DETECT* state, and *detectThreat()* calculates the actual threats among the stored ones. According to the sensor type, these algorithms are diversely developed.

$$Sensor_{Radar} = < X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta >$$

- $X = \{from\_CA\}$;
- $Y = \{to\_CA\}$;
- $S = \{WAIT, DETECT\}$;
- $\delta_{ext}: (WAIT) \times (from\_CA) \rightarrow (DETECT)$ *if from_CA is for C2 order,*
  $(DETECT) \times (from\_CA) \rightarrow (DETECT)$ *and execute addThreat(), refreshDetectTime() if from_CA is for threats,*
  $(DETECT) \times (from\_CA) \rightarrow (WAIT)$ *if from_CA is for C2 order;*
- $\delta_{int}: (DETECT) \rightarrow (DETECT)$ *and execute detectThreat();*
- $\lambda: (SEND) \rightarrow (to\_CA)$ *if threats are detected;*
- $ta: (WAIT) \rightarrow \infty$,
  $(DETECT) \rightarrow t_{DETECT}$.

The actuator model carries out specific physical behaviors such as launching an attack or maneuvering tactically. Similar to $\text{Sensor}_{\text{Radar}}$, $\text{Actuator}_{\text{Maneuver}}$, as an example of a propulsion system, has one operational state and one initial state. This model also maneuvers according to a received order and sends

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2907566, IEEE Access

the maneuvering result outside. The output, i.e., *to_CA*, has two data: 1) updated position whenever it moves every t_{Move} and 2) accomplishment of the C2 order. The detailed maneuvering action is described in *updatePosition()*. It is noted that the $C2_{Top}$, $Sensor_{Radar}$, and $Actuator_{Maneuver}$ explained here are examples of the C2 and CU models. Likewise, DEVS-based modeling provides a complete and clear semantics of an object to be modeled, thus reducing the effort required to read and understand it [47].

$$Actuator_{Maneuver} = < X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta >$$

- $X = \{from\_CA\}$;
- $Y = \{to\_CA\}$;
- $S = \{WAIT, MOVE\}$;
- δ_{ext}: $(WAIT) \times (from\_CA) \rightarrow (MOVE)$,
  $(MOVE) \times (from\_CA) \rightarrow (MOVE)$ *if from_CA is for a new C2 order*,
  $(MOVE) \times (from\_CA) \rightarrow (WAIT)$ *if from_CA is for stopping the current C2 order*;
- δ_{int}: $(MOVE) \rightarrow (MOVE)$ *and execute updatePosition()*;
- λ: $(MOVE) \rightarrow (to\_CA)$;
- ta: $(WAIT) \rightarrow 0$,
  $(MOVE) \rightarrow t_{MOVE}$.

In the introduction, we explained that CPSs are distant. It is true that most components within the same CPS are also apart from one another. For example, an antiballistic missile system is generally comprised of a radar system and interceptor launchers, which are physically distributed to retain a safe separation distance. From a modeling view, this means that CPS models and their component models are not directly connected.

In this case, the CA model is in charge of the following roles. It provides an indirect path for transferring messages between all CU and C2 models, e.g., orders from the C2 models and reports from the CU models. To exchange the messages through a communication network, the CA model interacts with the network model. In the network model, the messages are processed by being divided into two types: positional and traffic information. Thus, the CA model utilizes two mapping functions, i.e., *getNodeInformation()* and *getTrafficInformation()*, to convert the messages from the CPS models into interpreted forms in the network model.

The following specifications represent DEVS modeling of the CA model. It contains five states, which are classified into three groups: 1) *WAIT* for the model's standby, 2) *SEND_MOB_NET* for sending positional, i.e., mobility information from the CPS to the network model, and 3) *SEND_TRF_NET*, *SEND_TRF_CPS*, *RESEND*, and *LOSS* for exchanging traffic information between the two models. The positional information is transmitted in only one direction. When the CA model receives an updated location of the CPS model, the model immediately converts and forwards it to the network model and does not wait for any response.

On the contrary, traffic information is a two-way transmission. For example, we assume that the CA model in the *WAIT* state receives an order message from a C2 model, i.e., a source model. In this case, the CA model transforms the message into a traffic and delivers it to the network model at the *SEND_TRF_NET* state. Then it waits for the traffic result from the network model. Because a damage or a loss for the traffic can occur in the communication network, the two states, i.e., *RESEND* and *LOSS*, are necessary for deciding its retransmission or loss. If the CA model receives the result from the network model normally, it converts the input into the original order and forwards it to the destination model in the CPS model through the *SEND_TRF_CPS* state.

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

- $X = \{from\_CPS, from\_Net\}$;
- $Y = \{to\_CPS, to\_Net\}$;
- $S = \{WAIT, SEND\_MOB\_NET, SEND\_TRF\_NET, SEND\_TRF\_CPS, RESEND, LOSS\}$;
- δ_{ext}: $(WAIT) \times (from\_CPS) \rightarrow (SEND\_MOB\_NET)$ *if from_CPS is for mobility*,
  $(RESEND) \times (from\_CPS) \rightarrow (SEND\_MOB\_NET)$ *if from_CPS is for mobility*,
  $(WAIT) \times (from\_CPS) \rightarrow (SEND\_TRF\_NET)$ *and execute addTraffic() if from_CPS is for traffic*,
  $(RESEND) \times (from\_CPS) \rightarrow (RESEND)$ *and execute addTraffic(), refreshResendTime() if from_CPS is for traffic*,
  $(RESEND) \times (from\_Net) \rightarrow (SEND\_TRF\_CPS)$ *and execute removeTraffic()*;
- δ_{int}: $(SEND\_MOB\_NET) \rightarrow (WAIT)$ *and execute if the previous state is WAIT*,
  $(SEND\_MOB\_NET) \rightarrow (RESEND)$ *and execute refreshResendTime() if the previous state is RESEND*,
  $(SEND\_TRF\_NET) \rightarrow (RESEND)$,
  $(RESEND) \rightarrow (SEND\_TRF\_NET)$ *if retransmission is necessary*,
  $(RESEND) \rightarrow (LOSS)$ *and execute removeTraffic() if data are missed in the network model*,
  $(LOSS) \rightarrow (SEND\_TRF\_NET)$ *if the CA model has traffic*,
  $(LOSS) \rightarrow (WAT)$ *if the CA model has no traffic*,
  $(SEND\_TRF\_CPS) \rightarrow (SEND\_TRF\_NET)$ *if the CA model has traffic*,
  $(SEND\_TRF\_CPS) \rightarrow (WAIT)$ *if the CA model has no traffic*;
- λ: $(SEND\_MOB\_NET) \rightarrow (to\_Net)$ *and execute getNodeInformation()*,
  $(SEND\_TRF\_NET) \rightarrow (to\_Net)$ *and execute getNodeInformation() and getTrafficInformation()*,
  $(SEND\_TRF\_CPS) \rightarrow (to\_CPS)$ *and execute getNodeInformation() and getTrafficInformation()*;
- ta: $(WAIT) \rightarrow \infty$,
  $(SEND\_MOB\_NET) \rightarrow 0$,
  $(SEND\_TRF\_NET) \rightarrow 0$,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2907566, IEEE Access

$(SEND\_TRF\_CPS) \rightarrow 0,$
$(RESEND) \rightarrow t_{resend},$
$(LOSS) \rightarrow 0,$
$(DECISION) \rightarrow 0.$

Among the specific actions in this CA modeling, *getNodeInformation()* manages how the component models in the CPS model are mapped into distinct nodes in the network model. By performing this action, the CA model only sends the node identification (ID) instead of the component model itself for updating its position. Similarly, *getTrafficInformation()* maps the messages regarding orders and reports in the CPS model into network traffic with distinguishable IDs. These two actions are executed in the output functions.

One of the main advantages of developing DEVS models is the separation of modeling and its implementation. Among various DEVS-based M&S tools, we chose DEVSimHLA, which is a toolset that supports the development of HLA-compliant simulators based on the DEVS semantics [40]. The toolset consists mainly of two groups: one for standalone model realization and the other for model interoperation via HLA/RTI. The former group is represented by DEVSim++, which is a simulation engine for DEVS models. The next group for model interoperation includes the smart adaptor referred to as the KHLAAdaptor, which enables M&S developers to join any HLA-compliant simulator easily. In this study, we implemented the CGF model using DEVSim++; we used the KHLAAdaptor for interoperable simulation, which will be explained in the following subsection.

C. NETWORK MODEL DESIGN AND IMPLEMENTATION

The objective of network modeling is to reflect realistic communication effects for the CGF model. In this regard, the network model receives two types of messages from the CA model in the CGF model.

First, the network model receives positional messages of component models in the CPS models. As the component models move, according to the messages, the network model also updates the locations of communication nodes and dynamically revises the configuration of the nodes. Next, it receives operational messages, e.g., orders and reports described in Section 4.B, which are interacted between the CPS component models. When the network model receives the messages, it finds available routing paths, computes the communication effects of the paths such as communication delay and data loss, and sends the messages concerning computation to the CGF model. The second type of messages is regarded as communication traffic in the network model.

For network M&S, we used ns-3, which is a discrete-event network simulator based on object-oriented modeling. It contains a set of libraries to support various models and is compatible with other external libraries; thus, a modeler can easily modify his/her models in the open-source environment [50]. Because the MANET, our targeted network, is a collection of mobile nodes, it dynamically forms temporary links in the ns-3 simulator. When the state of a link changes from reliable to unreliable, one of the end nodes in this link will be newly selected as an active node. Thus, the MANET suffers from frequent link changes and failures, which in turn influence measures of mission effectiveness in military applications.

Figure 8 shows network M&S using the ns-3. Fundamentally, a node model in the network model is mapped into a component model with communication equipment in the CGF model. All the node models are interconnected to a channel model for the network packet flow. The node model has three hierarchical sub-models, i.e., net device, protocol stack, and application. The net device model transfers the packets over a channel model, and the protocol stack model describes a routing protocol for the MANET. Examples of reactive MANET routing protocols include ad hoc on demand distance vector (AODV), dynamic source routing (DSR), and destination sequenced distance vector (DSDV). Among them, in this study, we used the DSDV protocol due to its suitability for the military network systems [51, 52]. Finally, The application model expresses behaviors of the node [50].

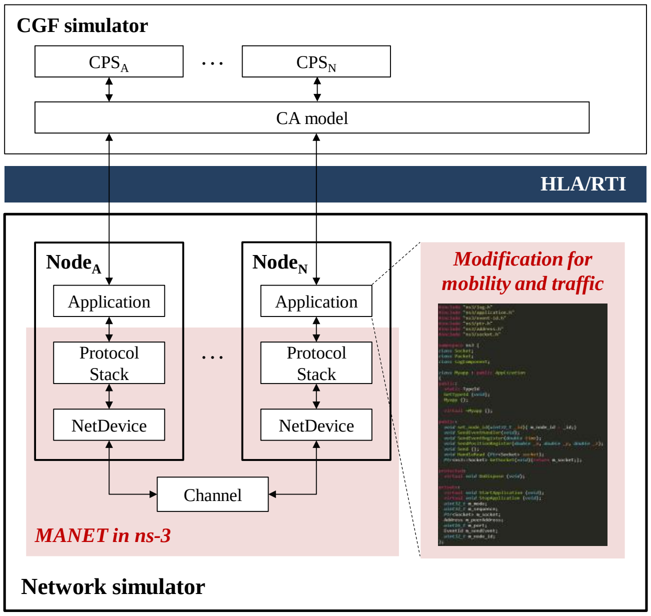

FIGURE 8. Network M&S using ns-3.

Specifically, the application model provides several application program interfaces (APIs) for mobility and traffic, and a modeler uses them to describe network configuration [53] [54]. Because the CGF model in our study provides these two factors, the application model needs to be modified. First, we extended the application model to receive mobility and traffic from the CGF model. Next, the ns-3 simulator does not support the HLA-compliant function; thus, we also modified the application model to participate in the HLA-based interoperable simulation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2907566, IEEE Access

D. MODEL INTEROPERATION WITH CGF AND NETWORK SIMULATORS

For sharing information between the two simulators, we designed a federation object model, as shown in Figure 9. It contains one object for the nodes' positions and one interaction for the traffic between them. The object, i.e., *NodeMobililty* in Figure 9, has two attributes for the corresponding node: *NodePosition* to update the position of the node and *NodeID* for its assigned ID. In the interaction, i.e., *MessageTraffic*, three parameters were defined: *SoruceNodeID* and *DestinationNodeID* to assign IDs of the both-sided nodes, and *MessageID* for an individual ID of the message. It is noted that *MessageTraffic* does not cover the detailed contents of the message.

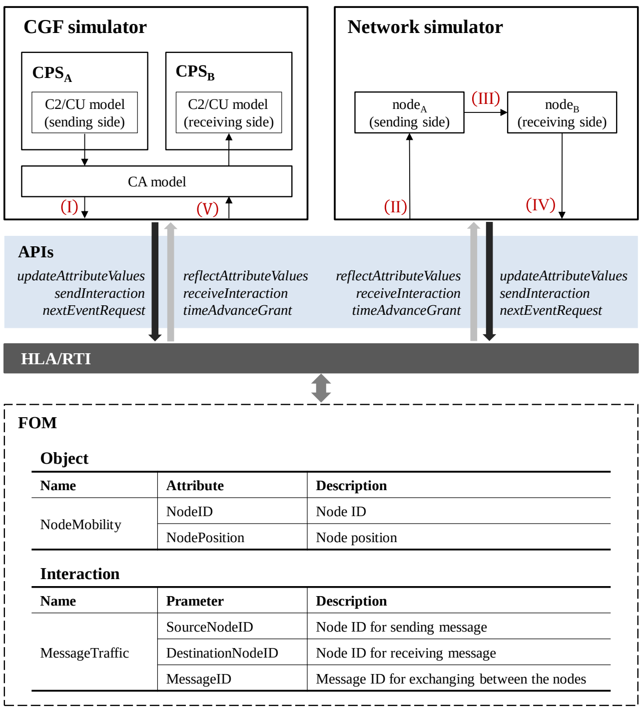

FIGURE 9. Interoperation between CGF and network simulators via high -level architecture.

The object and interaction are transferred between the two simulators with proper APIs in RTI. Specific APIs for this study are as follows. First, we used two sets of patterns to publish and subscribe: *sendInteraction* and *receiveInteraction* for the interaction and *updateAttributeValues* and *reflectAttributeValues* for the object. *nextEventRequest*, which is an event-driven pattern, is utilized for discrete event simulation; and *timeAdvanceGrant* is used to indicate the completion of the event request.

The Roman numerals in Figure 9 represent the five steps through which the simulators interacted with each other. In the first and second steps, as a source, CPS_A sends a C2 order to the network simulator via the CA model. In the third step, the network model computes communication traffic from $node_A$ to $node_B$, which influences the model's outputs, e.g., end-to-end delay or packet delivery ratio. After that, the network simulator sends the C2 order to the receiving side, i.e., CPS_B, via the CA model. All the steps excluding the third one can be achieved from HLA/RTI.

V. APPLICATION

This section presents a practical application of networked CPSs using the proposed M&S development. In the following subsections, we describe the simulation scenario, experimental design, model-based analysis, and discussion.

A. SIMULATION SCENARIO

The application focuses on network-centric ground warfare. Figure 10 simplifies a simulation scenario including blue (friendly) and red (hostile) CPSs. Arrows represent their expected paths and interacted communications during the warfare. The blue forces include a total of 131 military entities: specifically, 108 infantry-squad CUs, 12 platoon-level C2s, four company-level C2s, one battalion-level C2, and six mortar CUs. Infantry-squad CUs are designed as unit-level modeling to aggregate soldiers into a squad. Except for the 108 squads, all the entities are modeled as individual objects. In this scenario, the red forces are three times larger than the blue forces.

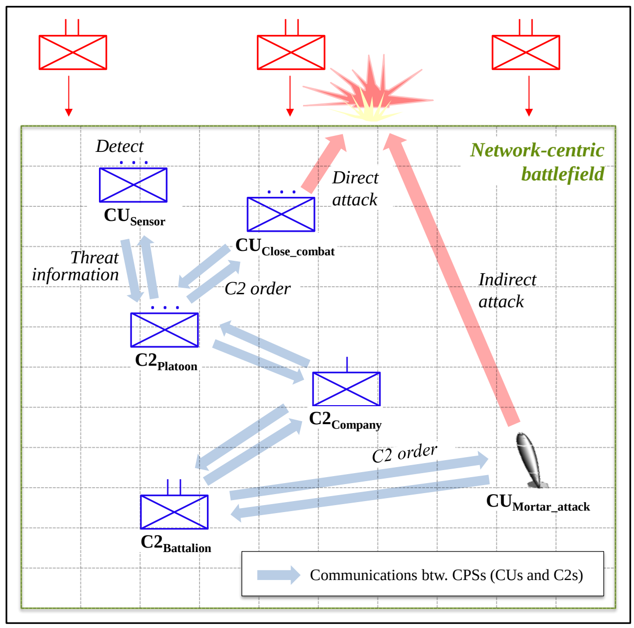

FIGURE 10. Illustration of abstracted simulation scenario.

The blue CPSs carry out close and standoff attacks against approaching enemies. When the red CPSs approach the detected range of the friendly one, the blue CUs on the front line detect them and report the threat information to the $C2_{Platoon}$. The report is hierarchically conveyed from the $C2_{Platoon}$ to the $C2_{Battalion}$ [55]. After the $C2_{Battalion}$ receives the report, it conducts a threat evaluation and weapon assignment (TEWA) process and commands the appropriate attacks, e.g.,

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

close combats or mortar attacks. The CU_{Close_combat} or the CU_{Mortar_attack} that receives the C2 order attacks the enemies.

The significant communications between the CUs and the C2s are expressed as blue arrows in Figure 10. The communications within the network may influence the mission success of the blue forces. For instance, indirect and direct attacks from the CU_{Mortar_attack} and the CU_{Close_combat} will be smooth if the communications are perfect without packet delay and data loss.

Based on the model configuration explained in Figure 5, we built hundreds of DEVS models, which is shown in Table 2. The friendly forces for this application contain 246 coupled and 246 atomic models. The number of CPS coupled models is equal to the number of friendly combat entities. All the DEVS models except for the CGF and CA models were reused with two different levels. First, behavior and stages of atomic models were reused to develop new models that are slightly different from the old ones. Next, a model itself was also composed to construct higher composite models [11]. These scalability, composability, and reusability are the main advantages of DEVS modeling [47].

TABLE 2. **Numbers of DEVS models to be developed for friendly forces.**

Model type	Description	No. of DEVS models: Coupled model	Atomic model
CGF	Outmost coupled model	1	
CPS	Military entities	131	
C2	Computational models with three hierarchies		17
CU	Physical coupled models	114	
Sensor	Sensor atomic models for infantry and mortar units		114
Actuator	Actuator atomic models for infantry and mortar units		114
CA	Communication agent		1
Total		246	246

From a military M&S view, an abstract level of CGF simulation would be above the mission level, and that of the network simulation would be the engineering level [11]. Because the scenario has been used for our previous studies [56-58], we assume that it was already validated.

B. EXPERIMENTAL DESIGN

Table 3 shows the primary input parameters of the two developed simulators. The CGF simulator has three computational parameters regarding decision-making and one physical parameter for weapon capability. The decision-making parameters are times related to C2, and the physical parameter is an engineering performance of a lethal radius of indirect weapons. We modeled 60-mm mortar systems for indirect weapons [59]. The lethal radius as the physical parameter is the bursting area the mortar bomb throws fragments in an irregular pattern so that the enemies are damaged. In this study, the lethal radius for the 60-mm mortar systems varies 20 to 30 meters [60].

The network simulator contains parameters for specifications of communication equipment. We first determined the main uncontrollable parameters of the ns-3 simulator as follows: the periodic time interval between exchange of full routing tables among nodes (*PeriodicUpdateInterval*) = 100 seconds, the minimum time an update is to be stored in a table (*SettlingTime*) = 5 seconds, the maximum number of packets that we allow a routing protocol to buffer (*MaxQueueLen*) = 500, the maximum number of packets that we allow a destination to buffer (*MaxQueuedPacketsPerDst*) = 5, and the maximum time packets can be queued (*MaxQueueTime*) = 30 seconds. For controllable parameters, we identified five ones of Table 3 affecting the simulation results after consulting with network system experts.

Because simulation models are characterized statistically, not deterministically, repeated simulation runs at the same experimental point allow for mitigation of the effects of statistical and random noise in experimental outputs. Nonetheless, there are no definitive guidelines for selecting the appropriate number of replications for Monte-Carlo simulation [61]. One simulation in this application lasted approximately 1.18 hours, which may be a little different

TABLE 3. **Simulation parameters of CGF and network simulators.**

Simulator type	Parameter name	Parameter level	No. of levels	Description
CGF simulator	Lethal radius (P_{CR})	20, 21, …, 30 (m)	11	The causality radius of CU model for indirect weapon
	C2 intelligence fusion time (P_{IF})	20, 24, …, 60 (sec)	11	The time required for intelligence fusion in C2 model
	C2 threat evaluation time (P_{TE})	20, 24, …, 60 (sec)	11	The time required for threat evaluation in C2 model
	C2 weapon assignment time (P_{WA})	20, 24, …, 60 (sec)	11	The time required for weapon assignment in C2 model
Network simulator	Packet size (P_{PS})	100, 200, 400, …, 6400 (byte)	11	The size of packet
	Transmission power (P_{TP})	-10, -5, 0, 5, …, 40 (dBm)	11	The transmission power of node
	Transmission gain (P_{TG})	0, 2, 4, …, 20 (dB)	11	The transmission gain of node
	Reception gain (P_{RG})	0, 2, 4, …, 20 (dB)	11	The reception gain of node
	PhyMode (P_{PM})	1, 2, 5.5, 11 (Mbps)	4	The 802.11 phy layer mode of DsssRate

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

depending on experiment points. Due to the constraint of such a time-consuming problem, we need to choose the lower bound for the replications to ensure the output accuracy statistically. Consequently, we repeated 30 times every experimental point, which is generally derived from the probabilistic model based on the central limit theorem. This number of repetitions has been widely used for Monte-Carlo simulation that should consider simulation cost and accuracy in a balanced way [62].

Table 4 indicates the measurement indices for evaluating mission success and network performance. In the military analysis, the former is relevant to the measure of effectiveness (MoE) to accomplish mission objectives, and the latter corresponds to the measure of performance (MoP) that represents system-particular quantifiable features. This application evaluates two MoEs for the CGF simulator and two MoPs for the network simulator [63].

TABLE 4. Simulation outputs of CGF and network simulators: mission effectiveness and network performance.

Simulator type	Measurement index	Description
CGF simulator	Enemy-survivability rate	The ratio of the number of red survivors over the number of initial red forces
	Loss-exchange ratio	The ratio of the number of red losses over the number of blue losses
Network simulator	Packet delivery ratio	The ratio of the number of successfully delivered packets from the source to the destination node
	End-to-end delay	The average time taken by packets to arrive at the destination from the source node

Finally, the development environment for this application is as follows: DEVSim++ (ver. 3.1), ns-3 (ver. 3.18), and RTI 1.3-NG were used for the CGF simulator, the network simulator, and their interoperation, respectively. All the software was executed in desktops with I5-3550 3.3 GHz CPU and 8 GB RAM.

C. MODEL-BASED ANALYSIS: COMPLEMENTARY ANALYSIS

The ideal experimentation for simulation analysis is to use all design spaces. In our case, the total experimental points are $11(P_{CR}) \times 11(P_{IF}) \times 11(P_{TE}) \times 11(P_{WA}) \times 11(P_{PS}) \times 11(P_{TP}) \times 11(P_{TG}) \times 11(P_{RG}) \times 4(P_{PM}) = 857{,}435{,}524$. These numerous experiments via interoperable simulation cause a time-consuming problem due to the computational complexity of the simulators and the overhead of interoperation middleware [56].

To overcome this weakness, we carried out two statistical analyses from macroscopic and microscopic views. The macro-analysis analyzes simulation results by changing all the parameters of the two simulators. For example, it facilitates the CGF simulator to compute its outputs via dynamical interaction with the network simulator. On the other hand, the micro-analysis concentrates on interesting points of one simulator by fixing the parameters of the other simulator. Thus, the macro-analysis is more effective when an analyst needs to know the general tendencies between the inputs and outputs of the two simulators at the beginning of the experiments. Afterward, he or she focuses on specific experimental points of one simulator to keep the influences of the other simulator constant. These two-stage analyses are the best way to carry out the complementary analysis proposed in this study.

1) EXPERIMENT 1: MACRO-ANALYSIS

Of the entire design spaces in Table 2, we selected a total of 257 experimental points for the macro-analysis: 147 points using face-centered central composite (FCC), 55 points using Latin hypercube (LH) design, and 55 random points.

As a representative simulation result, Figure 11 shows the change of the enemy-survivability rate according to the two different communication situations. The blue-line group indicates the simulation results with the poor network environment, i.e., P_{PS} = 100, P_{TP} = -10, P_{TG} = 0, P_{RG} = 20, and P_{PM} = 1. On the contrary, the red-line group shows the results based on a better network environment, i.e., P_{PS} = 100, P_{TP} = 40, P_{TG} = 20, P_{RG} = 20, and P_{PM} = 1. Each group has multiple light lines for repeated simulations and one clear line for an average of the repetitions. Therefore, the gaps between the light line and the clear one mean statistical and random noise.

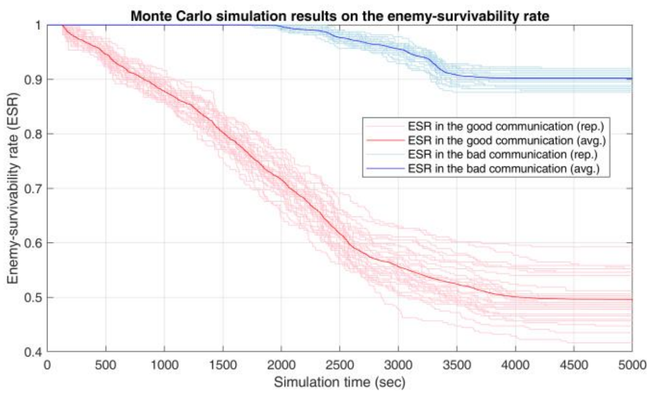

FIGURE 11. Simulation results on enemy-survivability rate over simulation time.

The two groups similarly show a tendency to decrease the enemy-survivability rate over simulation time. In particular, the red group shows a more rapid decrease than the blue one, which means that better communication brings more mission success. This trend is especially true at the early stage of the simulation when the indirect attacks are carried out.

In order to interpret the sensitivity and robustness of the simulation results, we built first-order linear regression models. Tables 5 and 6 show the models of Experiment 1 ($y \sim 1 + P_{CR} + P_{IF} + P_{TE} + P_{WA} + P_{PS} + P_{TP} + P_{TG} + P_{RG} + P_{PM}$). We measured the standardized errors of the estimates (*SE*), t-statistic values of hypothesis tests for the corresponding coefficients (t_{Stat}), and significant probabilities

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

TABLE 5. **Linear regression models on two types of mission effectiveness at macro level.**

	Enemy-survivability rate: *coefficient*	Enemy-survivability rate: *SE*	Enemy-survivability rate: *tStat*	Enemy-survivability rate: *p-value*
C	1.0035	0.0355	28.2303	2.8460e-79
P_{CR}	-0.0058	0.0011	-5.3663	1.8490e-07
P_{IF}	-1.5187e-04	2.7522e-04	-0.5518	0.5816
P_{TE}	7.5907e-05	2.7195e-04	0.2791	0.7804
P_{WA}	2.2544e-04	2.7624e-04	0.8161	0.4152
P_{PS}	6.2323e-06	1.6351e-06	3.8116	1.7438e-04
P_{TP}	-0.0036	2.1993e-04	-16.4951	1.0226e-41
P_{TG}	-0.0030	5.4991e-04	-5.4304	1.3440e-07
P_{RG}	-0.0030	5.5136e-04	-5.4893	9.9963e-08
P_{PM}	0.0047	0.0010	4.5530	8.3084e-06

adj. R^2 = 0.610, *p-value* = 1.26e-47

	Loss-exchange ratio: *coefficient*	Loss-exchange ratio: *SE*	Loss-exchange ratio: *tStat*	Loss-exchange ratio: *p-value*
C	-0.0610	0.1190	-0.5128	0.6086
P_{CR}	0.0210	0.0036	5.7862	2.17e-08
P_{IF}	2.8290e-04	9.2111e-04	0.3071	0.7590
P_{TE}	-4.6770e-04	9.1016e-04	-0.5139	0.6078
P_{WA}	-0.0011	9.2451e-04	-1.1462	0.2528
P_{PS}	-2.15e-05	5.47e-06	-3.9332	1.0898e-04
P_{TP}	0.0116	7.3606e-04	15.7413	3.90e-39
P_{TG}	0.0095	0.0018	5.17534	4.71e-07
P_{RG}	0.0097	0.0018	5.25101	3.26e-07
P_{PM}	-0.0149	0.0034	-4.3256	2.21e-05

adj. R^2 = 0.594, *p-value* = 1.65e-45

TABLE 6. **Linear regression models on two types of network performance at macro level.**

	End-to-end delay: *coefficient*	End-to-end delay: *SE*	End-to-end delay: *tStat*	End-to-end delay: *p-value*
C	0.0230	0.1029	0.2231	0.8236
P_{CR}	3.0360e-04	0.0031	0.0967	0.9230
P_{IF}	3.0692e-04	7.9546e-04	0.3858	0.6999
P_{TE}	2.6683e-04	7.8654e-04	0.3393	0.7347
P_{WA}	2.7188e-04	7.9788e-04	0.3408	0.7336
P_{PS}	4.56e-05	4.72e-06	9.6651	6.12e-19
P_{TP}	0.0035	6.3514e-04	5.5316	8.10e-08
P_{TG}	7.1376e-04	0.0016	0.4487	0.6541
P_{RG}	0.0017	0.0016	1.0704	0.2855
P_{PM}	-0.0225	0.0030	-7.5529	8.34e-13

adj. R^2 = 0.398, *p-value* = 9.41e-25

	Packet delivery ratio: *coefficient*	Packet delivery ratio: *SE*	Packet delivery ratio: *tStat*	Packet delivery ratio: *p-value*
C	1.0035	0.0355	28.2303	2.8460e-79
P_{CR}	-0.0058	0.0011	-5.3663	1.8490e-07
P_{IF}	-1.5187e-04	2.7522e-04	-0.5518	0.5816
P_{TE}	7.5907e-05	2.7195e-04	0.2791	0.7804
P_{WA}	2.2544e-04	2.7624e-04	0.8161	0.4152
P_{PS}	6.2323e-06	1.6351e-06	3.8116	1.7438e-04
P_{TP}	-0.0036	2.1993e-04	-16.4951	1.0226e-41
P_{TG}	-0.0030	5.4991e-04	-5.4304	1.3440e-07
P_{RG}	-0.0030	5.5136e-04	-5.4893	9.9963e-08
P_{PM}	0.0047	0.0010	4.5530	8.3084e-06

adj. R^2 = 0.610, *p-value* = 1.26e-47

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2907566, IEEE Access

(*p-value*) [64] [65]. The *SE* is relevant to the estimation noise, and the t_{Stat} (=*coeff./SE*) means the actual signal against the noise. When a parameter has a *p-value* less than 0.05, it can be significant for a confidence interval of 95%.

In Table 5, because the *p-values* of all parameters except P_{IF}, P_{TE}, and P_{WA} are less than 0.05, they are statistically significant for the outputs of the CGF simulator. Note that P_{IF}, P_{TE}, and P_{WA} seem to have little effects, although they are the parameters of the CGF simulator. The parameters with higher *coefficients* have more significant influences on the outputs than the parameters with lower values. In this regard, P_{CR} has the most significant impact on mission effectiveness.

The results of Table 6 are slightly different from those of Table 5. Only three parameters, i.e., P_{PS}, P_{TP}, and P_{PM}, are statistically significant for the end-to-end delay; on the other hand, all the parameters of the network simulator along with P_{CR} are statistically significant for the packet delivery ratio. Similar to Table 5, P_{IF}, P_{TE}, and P_{WA} also have little effects on network performance. Finally, the *adj.* R^2 values of Tables 5 and 6 are less than or equal to 0.61. The small *adj.* R^2 values do not guarantee that the regression models match with the original data, which makes it necessary to conduct microscopic analyses.

2) EXPERIMENT 2: MICRO-ANALYSIS FROM CGF SIMULATOR SIDE

As explained in Section V.C, the micro-analysis of one simulator is based on the fixed input parameters of the other simulator. Note that fixing the parameters does not mean that the opposite simulator generates constant outputs all the time. For example, although x_N of the network simulator in Figure 1 is fixed, v_N and y_N can be changeable according to the received v_{CGF}. In this study, for the micro-analysis, we fixed the parameters when the opposite simulator generates ideal outputs.

In Experiment 2, the ideal point of the network simulator is where it generates a high packet delivery ratio and a low end-to-end delay. As shown in Figure 12, the cross marks in two graphs indicate the overall 257 points depending on the two outputs of the network simulator. Among them, we found a pair of marks to satisfy the above conditions, which are indicated in red. This experimental point has P_{PS} = 800, P_{TP} = 10, P_{TG} = 15, P_{RG} = 15, and P_{PM} = 1.

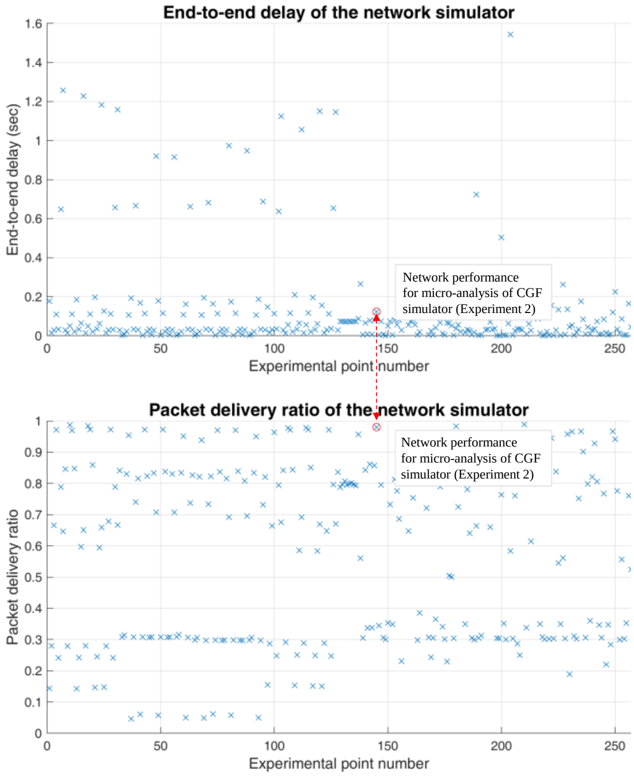

FIGURE 12. **Selection of experimental point of network simulator for CGF simulator's micro-analysis.**

TABLE 7. **Linear regression models on two types of mission effectiveness of CGF simulator parameters at microscopic level.**

	Enemy-survivability rate: *coefficient*	Enemy-survivability rate: *SE*	Enemy-survivability rate: *tStat*	Enemy-survivability rate: *p-value*
C	0.8833	0.0124	70.975	1.1798e-48
P_{CR}	-0.0116	0.0004	-29.517	1.5322e-31
P_{IF}	0.0008	9.5324e-05	8.026	2.7077e-10
P_{TE}	0.0006	9.6496e-05	6.1857	1.5246e-07
P_{WA}	0.0007	0.0001	7.0184	8.5508e-09

adj. R^2 = 0.962, *p-value* = 4.01e-32

	Loss-exchange ratio: *coefficient*	Loss-exchange ratio: *SE*	Loss-exchange ratio: *tStat*	Loss-exchange ratio: *p-value*
C	0.2493	0.0650	3.8346	0.0004
P_{CR}	0.0434	0.0020	21.1212	2.5977e-25
P_{IF}	-0.0029	0.0004	-5.9186	3.8346e-07
P_{TE}	-0.0023	0.0005	-4.6184	3.1239e-05
P_{WA}	-0.0028	0.0005	-5.3160	3.0224e-06

adj. R^2 = 0.93, *p-value* = 6e-26

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2907566, IEEE Access

Although the parameters of the network simulator are fixed, the full design space from the CGF simulator side is $11(P_{CR}) \times 11(P_{IF}) \times 11(P_{TE}) \times 11(P_{WA}) = 14{,}641$. This still requires a long execution time, with approximately 518,000 hours when 30 repetitions per each point are conducted (1.18 hours per one execution). For this reason, we selected 51 experimental points from the entire spaces: 25 points using the FCC design, 9 points using the LH design, and 17 random points.

Table 7 shows first-order linear regression models for Experiment 2 $(y{\sim}1 + P_{CR} + P_{IF} + P_{TE} + P_{WA})$. First, through the *p-values*, all the parameters are statistically significant. Next, the signs of *coefficients* inform whether they are directly or inversely proportional to the outputs. Because the two outputs of the CGF simulator have completely conflicting trends, the *coefficient* of one parameter has different signs according to the outputs. In addition, P_{CR} has a larger *coefficient* than P_{IF}, P_{TE}, and P_{WA}, which means that the physical parameter is more affected than the computational parameters in this experiment. Finally, the *adj.* R^2 values are high enough to judge that the model fully matches the original data. This is a clear difference from the macro-analysis.

3) EXPERIMENT 3: MICRO-ANALYSIS FROM NETWORK SIMULATOR SIDE

In Experiment 3, the micro-analysis from the network simulator side was performed. Two graphs in Figure 13 show all 257 experimental points according to the two outputs of the CGF simulator, i.e., the enemy-survivability rate and the loss-exchange ratio. The ideal point in view of the CGF simulator is the one with not only a low enemy-survivability rate but also a high loss-exchange ratio. A pair of red marks in Figure 13 are the ideal point, which has $P_{CR} = 30$, $P_{IF} = 20$, $P_{TE} = 20$, $P_{WA} = 20$.

Similar to Experiment 2, the full design space from the network simulator side requires about 2,073, 000 hours, which is estimated from 30 repeated simulations for 58,564 points and 1.18 hours per 1 simulation. Thus, we selected 87 experimental points from the entire design space: 43 points using the FCC, 22 points using the LH design, and 22 random points.

FIGURE 13. Selection of experimental point of CGF simulator for network simulator's micro-analysis.

TABLE 8. Linear regression models on two types of network performance of network simulator at microscopic level (taken from [51]).

	End-to-end delay: *coefficient*	End-to-end delay: *SE*	End-to-end delay: *tStat*	End-to-end delay: *p-value*
C	0.0775	0.0660	1.1737	0.2452
P_{PS}	4.68e-05	9.28e-06	5.0439	4.64e-06
P_{TP}	0.0033	0.0013	2.6683	0.0098
P_{TG}	9.3956e-04	0.0031	0.2987	0.7662
P_{RG}	6.5681e-04	0.0031	0.2088	0.8354
P_{PM}	-0.0234	0.0057	-4.0851	1.34749e-04

adj. $R^2 = 0.39$, *p-value* = 1.66e-06

	Packet delivery ratio: *coefficient*	Packet delivery ratio: *SE*	Packet delivery ratio: *tStat*	Packet delivery ratio: *p-value*
C	0.3844	0.0498	7.7202	1.63e-10
P_{PS}	-1.59e-05	7.00e-06	-2.2756	0.0265
P_{TP}	0.0102	9.4723e-04	10.7808	1.43e-15
P_{TG}	0.0088	0.0024	3.7153	4.5282e-04
P_{RG}	0.0084	0.0024	3.5254	8.2480e-04
P_{PM}	-0.0160	0.0043	-3.6865e-04	4.9648e-04

adj. $R^2 = 0.712$, *p-value* = 8.14e-16

Table 8 shows the first-order regression models for Experiment 3($y \sim 1 + P_{PS} + P_{TP} + P_{TG} + P_{RG}$). As shown in the upper table, P_{TG} and P_{RG} have little effects on end-to-end delay. And the *adj.* R^2 value is 0.39, which does not guarantee that the regression model will match the original data. On the other hand, in the table below, all the network parameters influence the packet delivery ratio. This tendency is similar to the result of the macro-analysis. Next, in comparison to Table 6, the *adj.* R^2 values are higher, which guarantees that the micro-analysis is reliable enough to complement the results of the macro-analysis.

D. DISCUSSION

We discuss the statistical tendencies between the inputs and outputs of the CGF and the network simulators within the interoperable simulation. Experiment 1 macroscopically shows how the outputs of one simulator were influenced by the inputs or parameters of the other simulator. In the case of the CGF simulator, all communication parameters of the network simulator, i.e., P_{PS}, P_{TP}, P_{TG}, P_{RG}, and P_{PM}, had effects on both outputs of the CGF simulator. As shown in Figure 11, during the poor communications between friendly CPSs, the enemy-survival rate was unchanged at the beginning of the simulation and slightly decreased after 2,000 seconds. It implies that high-performance CPSs cannot always carry out successive missions according to network conditions.

Next, in the case of the network simulator, the computational parameters, i.e., P_{IF}, P_{TE}, and P_{WA}, have little effects on both outputs of the network simulator. On the contrary, the physical parameter, P_{CR}, about weapon performance is affected by the output of the network simulator. The interesting point is that P_{CR} has considerable influence on the packet delivery ratio in common with the communication parameters. Thus, in order to analyze network performance, it is important to consider network configuration and the simulation scenario. It is the main reason why model interoperation and complementary analysis are necessary.

In Experiments 2 and 3, we analyzed the inputs and outputs of one simulator by keeping the other simulator ideal for the micro-analysis. Specifically, Experiment 2 shows that P_{CR} has the most influence on the enemy-survivability rate and the loss-exchange ratio. In our experiments, physical factors more influence mission effectiveness than computational factors. In Experiment 3, although all the network parameters affect the packet delivery ratio, gain-related parameters, i.e., P_{TG} and P_{RG}, have little effects on the end-to-end delay compared to the other parameters.

The proposed M&S could be utilized for various what-if scenarios such as cyber attacks. For example, it can be used to simulate behaviors of CPSs under the network structure and analyze the proper alternatives for them to overcome the situation. To analyze various scenarios, we remain two additional techniques, i.e., simulation acceleration for reducing the overall simulation time and accurate statistical methods for describing the results. We have studied several modeling methods to reduce the simulation execution time by embracing a slight amount of error [57, 58]. Although these techniques are outside the scope of this study, it is helpful to use them for various complementary analyses of networked CPSs.

VI. CONCLUSION

Because communication technology is increasingly essential, networked CPSs have become a prerequisite. In this regard, NCW containing defense CPSs have been analyzed using M&S techniques; however, these studies suffer from shortcomings such as unified model development and localized simulation analysis.

This study proposed M&S development to address two issues, i.e., model-driven development via model interoperation and simulation analysis between interoperated models. We proposed modeling activities in a way that leads to the achievement of capabilities through an SoS approach rather than from just the performance of individual systems. As the main contribution, the proposed model and complementary analysis give rise to a practical insight that cannot be achieved with single-system simulation and single analysis.

In the case study, we analyzed the computational and physical capabilities of the CPSs as well as the communication effects between the CPSs with macroscopic and microscopic views. Specifically, we discussed several findings, i.e., balancing physical and computational abilities, the importance of information technology, and statistical trends between the inputs and outputs of the models. We hope this study provides better information and confidence to simulation experts and provides an interpretation to people who may not sufficiently comprehend the M&S results.

REFERENCES

[1] A. Laghari, Z. A. Memon, S. Ullah, and I. Hussain, "Cyber Physical System for Stroke Detection," *IEEE Access*, vol. 6, pp. 37444-37453, 2018.

[2] Y. Yang, X. Li, Z. Yang, Q. Wei, N. Wang, and L. Wang, "The Application of Cyber Physical System for Thermal Power Plants: Data-Driven Modeling," *Energies*, vol. 11, no. 4, p. 690, 2018.

[3] S. M. Rahman, "Cyber-physical-social system between a humanoid robot and a virtual human through a shared platform for adaptive agent ecology," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 190-203, 2018.

[4] X. Cao, P. Cheng, J. Chen, and Y. Sun, "An online optimization approach for control and communication codesign in networked cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 439-450, 2013.

[5] B. Satchidanandan and P. R. Kumar, "Dynamic watermarking: Active defense of networked cyber–

physical systems," *Proceedings of the IEEE*, vol. 105, no. 2, pp. 219-240, 2017.

[6] X. Guan, B. Yang, C. Chen, W. Dai, and Y. Wang, "A comprehensive overview of cyber-physical systems: From perspective of feedback system," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 1, pp. 1-14, 2016.

[7] Y. Guo, X. Hu, B. Hu, J. Cheng, M. Zhou, and R. Y. Kwok, "Mobile cyber physical systems: Current challenges and future networking applications," *IEEE Access*, vol. 6, pp. 12360-12368, 2018.

[8] K. H. Lee, J. H. Hong, and T. G. Kim, "System of Systems Approach to Formal Modeling of CPS for Simulation-Based Analysis," *Etri Journal*, vol. 37, no. 1, pp. 175-185, 2015.

[9] Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, "Review on cyber-physical systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 27-40, 2017.

[10] J. Moffat, *Complexity theory and network centric warfare*. DIANE Publishing, 2010.

[11] A. Tolk, *Engineering principles of combat modeling and distributed simulation*. John Wiley & Sons, 2012.

[12] B. Xiong, B. Li, R. Fan, Q. Zhou, and W. Li, "Modeling and Simulation for Effectiveness Evaluation of Dynamic Discrete Military Supply Chain Networks," *Complexity*, vol. 2017, pp. 6052037.1-6052037.9, 2017.

[13] K. Klingensmith and A. M. Madni, "Resilience Concepts for Architecting an Autonomous Military Vehicle System-of-Systems," in *Disciplinary Convergence in Systems Engineering Research*: Springer, 2018, pp. 65-81.

[14] S. Mukherjee and S. Chaudhury, "C-MAP: framework for multi-agent planning in Cyber physical systems," in *International Conference on Pattern Recognition and Machine Intelligence*, 2013, pp. 237-242: Springer.

[15] W.-T. Kim, I.-G. Chun, S.-H. Lee, H.-Y. Lee, and J. M. Kim, "Wip abstract: From design to operation of a large-scale cps," in *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, 2012, p. 215: IEEE Computer Society.

[16] Z. Liu, D.-s. Yang, D. Wen, W.-m. Zhang, and W. Mao, "Cyber-physical-social systems for command and control," *IEEE Intelligent Systems*, vol. 26, no. 4, pp. 92-96, 2011.

[17] T. A. Bapty, J. Scott, S. Neema, and R. Owens, "Integrated modeling and simulation for cyberphysical systems extending multi-domain M&S to the design community," in *Proceedings of the Symposium on Model-driven Approaches for Simulation Engineering*, 2017, p. 12.

[18] C. Shum *et al.*, "DecompositionJ: Parallel and Deterministic Simulation of Concurrent Java Executions in Cyber-Physical Systems," *IEEE Access*, vol. 6, pp. 21991-22010, 2018.

[19] P. Derler, E. A. Lee, and A. S. Vincentelli, "Modeling cyber–physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 13-28, 2012.

[20] D. Sullivan, E. Colbert, A. Kott, L. Osterritter, and G. Dobson, "Best Practices for Designing and Conducting Cyber-Physical System Wargames," in *International Conference on Cyber Warfare and Security*, 2018, pp. 651-XVI: Academic Conferences International Limited.

[21] K.-M. Seo, W. Hong, and T. G. Kim, "Enhancing model composability and reusability for entity-level combat simulation: A conceptual modeling approach," *Simulation*, vol. 93, no. 10, pp. 825-840, 2017.

[22] S. J. Taylor *et al.*, "Grand challenges for modeling and simulation: simulation everywhere—from cyberinfrastructure to clouds to citizens," *simulation*, vol. 91, no. 7, pp. 648-665, 2015.

[23] N. Li, W. Huai, and S. Wang, "The solution of target assignment problem in command and control decision-making behaviour simulation," *Enterprise Information Systems*, vol. 11, no. 7, pp. 1059-1077, 2017.

[24] J. Durham *et al.*, "Network-centric operations support: Lessons learned, status, and way-ahead," in *Proceeding of the 19th International Command and Control Research and Technology (ICCRTS) Symposium*, Alexandria Virhinia, 2014.

[25] T. G. Kim *et al.*, "DEVSim++ toolset for defense modeling and simulation and interoperation," *The Journal of Defense Modeling and Simulation*, vol. 8, no. 3, pp. 129-142, 2011.

[26] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press, 2000.

[27] I. Akkaya, P. Derler, S. Emoto, and E. A. Lee, "Systems engineering for industrial cyber–physical systems using aspects," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 997-1012, 2016.

[28] *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification (IEEE Std 1516-2000)*, 2001.

[29] J. Mohammad, "System of systems engineering: innovations for the twenty-first century," *JohnWiley & Sons*, 2011.

[30] K.-M. Seo, K.-P. Park, and B.-J. Lee, "Achieving Data Interoperability of Communication Interfaces for Combat System Engineering," *IEEE Access*, vol. 5, pp. 17938-17951, 2017.

[31] M. Jamshidi, *Systems of systems engineering: principles and applications*. CRC press, 2008.

[32] M. A. Hofmann, "Challenges of model interoperation in military simulations," *Simulation*, vol. 80, no. 12, pp. 659-667, 2004.

[33] J. Walsh, J. Roberts, and W. Thompson, "NCW End-to-end (NETE) Model for Future C2 Architecture Assessments," in *Proceedings of the 2005 International Command and Control Research and Technology Symposium The Future of Command and Control*, 2005.

[34] W. S. Murphy and M. A. Flournoy, "Simulating crisis communications," in *Proceedings of the 2002 Winter Simulation Conference* 2002, vol. 1, pp. 954-959: IEEE.

[35] H. Nam and T. Lee, "Modeling and simulation for evaluating the C3 structure in a NCW mission environment," in *Proceedings of the 2015 Winter Simulation Conference*, 2015, pp. 3204-3205: IEEE Press.

[36] N. E. Miner *et al.*, "Evaluating communications system performance effects at a system of systems level," in *Military Communications Conference, 2012-MILCOM*, 2012, pp. 1-6: IEEE.

[37] E. Cayirci and C. Ersoy, "Simulation of tactical communications systems by inferring detailed data from the joint theater-level computer-aided exercises," *Simulation,* vol. 78, no. 8, pp. 475-484, 2002.

[38] B. D. Paz and J. A. Baer, "Communication Effect Server Integration with OneSAF for Mission Level Simulation," in *Proceedings of the 2008 Fall Simulation Interoperability Workshop (SIW)*, 2008, pp. 271-276.

[39] C. Sung and T. G. Kim, "Collaborative modeling process for development of domain-specific discrete event simulation systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 42, no. 4, pp. 532-546, 2012.

[40] J. Kim, I.-C. Moon, and T. G. Kim, "New insight into doctrine via simulation interoperation of heterogeneous levels of models in battle experimentation," *Simulation,* vol. 88, no. 6, pp. 649-667, 2012.

[41] M. E. Páte-Cornell and P. S. Fischbeck, "Probabilistic interpretation of command and control signals: Bayesian updating of the probability of nuclear attack," *Reliability Engineering & System Safety,* vol. 47, no. 1, pp. 27-36, 1995.

[42] A. Yang, H. A. Abbass, and R. Sarker, "Land combat scenario planning: A multiobjective approach," in *Asia-Pacific Conference on Simulated Evolution and Learning*, 2006, pp. 837-844: Springer.

[43] T. Samad, J. S. Bay, and D. Godbole, "Network-centric systems for military operations in urban terrain: The role of UAVs," *Proceedings of the IEEE,* vol. 95, no. 1, pp. 92-107, 2007.

[44] S. Mittal and J. L. Risco-Martín, "DEVSML studio: a framework for integrating domain-specific languages for discrete and continuous hybrid systems into DEVS-based m&s environment," in *Proceedings of the Summer Computer Simulation Conference*, 2016, p. 41: Society for Computer Simulation International.

[45] G. A. Wainer, "Real-time simulation of DEVS models in CD++," *International Journal of Simulation and Process Modelling,* vol. 11, no. 2, pp. 138-153, 2016.

[46] *IEEE standard for modeling and simulation (M&S) High Level Architecture (HLA)-Framework and Rules*, 2000.

[47] K.-M. Seo, C. Choi, T. G. Kim, and J. H. Kim, "DEVS-based combat modeling for engagement-level simulation," *Simulation,* vol. 90, no. 7, pp. 759-781, 2014.

[48] J. H. Kim, K.-M. Seo, T.-E. Lee, and B.-W. Choi, "Achieving New Insights into Combat Engagement Analysis via Simulation-Based Sequential Experimentation," *Military Operations Research,* vol. 23, no. 4, pp. 51-80, 2018.

[49] H. S. Song and T. G. Kim, "DEVS diagram revised: a structured approach for DEVS modeling," in *Proc. European Simulation Conference (Eurosis, Belgium, 2010)*, 2010, pp. 94-101.

[50] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM demonstration,* vol. 14, no. 14, p. 527, 2008.

[51] A. Aggarwal, S. Gandhi, and N. Chaubey, "Performance analysis of AODV, DSDV and DSR in MANETS," *International Journal of Distributed and Parallel Systems,* vol. 2, no. 6, pp. 167-177, 2011.

[52] S. Katiyar, R. Gujral, and B. Mallick, "Comparative performance analysis of MANET routing protocols in military operation using NS2," in *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on*, 2015, pp. 603-609: IEEE.

[53] F. Bai and A. Helmy, "A survey of mobility models," *Wireless Adhoc Networks. University of Southern California, USA,* vol. 206, p. 147, 2004.

[54] F. Gebali, "Modeling Network Traffic," in *Analysis of Computer and Communication Networks*: Springer, 2008, pp. 1-47.

[55] A. H. Dekker, C4ISR architectures, social network analysis and the FINC methodology: an experiment in military organisational structure: Defence Science & Technology Organization (DTSO) 2002. [Online]. Available.

[56] B. G. Kang, K.-M. Seo, and T. G. Kim, "Communication analysis of network-centric warfare via transformation of system of systems model into integrated system model using neural network," *Complexity,* vol. 2018, pp. 6201356.1-6201356.16, 2018.

[57] B. G. Kang, S. H. Choi, S. J. Kwon, J. H. Lee, and T. G. Kim, "Simulation-Based Optimization on the System-of-Systems Model via Model Transformation and Genetic Algorithm: A Case Study of Network-

Centric Warfare," *Complexity*, vol. 2018, pp. 4521672.1-4521672.15, 2018.

[58] B. G. Kang, K.-M. Seo, and T. G. Kim, "Machine learning-based discrete event dynamic surrogate model of communication systems for simulating the command, control, and communication system of systems," *Simulation*, vol. 95, 2019, DOI: 10.1177/0037549718809890 [online published].

[59] W.-S. Yun, I.-C. Moon, and T.-E. Lee, "Agent-based simulation of time to decide: Military commands and time delays," *Journal of Artificial Societies and Social Simulation*, vol. 18, no. 4, pp. 1-32, 2015.

[60] D. Express. (2018, Feb.). *60-MM MORTAR IN THE CONTEXT OF THE CURRENT BATTLEFIELD CHALLENGES*. Available: https://defence-ua.com/index.php/en/publications/defense-express-publications/4930-60-mm-mortar-in-the-context-of-the-current-battlefield-challenges

[61] D. J. Mundform, J. Schaffer, M.-J. Kim, D. Shaw, A. Thongteeraparp, and P. Supawan, "Number of replications required in Monte Carlo simulation studies: A synthesis of four studies," *Journal of Modern Applied Statistical Methods*, vol. 10, no. 1, pp. 19-28, 2011.

[62] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky, "A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems," *Journal of global optimization*, vol. 31, no. 4, pp. 635-672, 2005.

[63] M. F. Khan, E. A. Felemban, S. Qaisar, and S. Ali, "Performance analysis on packet delivery ratio and end-to-end delay of different network topologies in Wireless Sensor Networks (WSNs)," in *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on*, 2013, pp. 324-329: IEEE.

[64] C. K. McCalley *et al.*, "Methane dynamics regulated by microbial community response to permafrost thaw," *Nature*, vol. 514, no. 7523, p. 478, 2014.

[65] J. H. Lee *et al.*, "Effect of Rheumatoid Factor on Vascular Stiffness in General Population without Joint Symptoms," *Kosin Medical Journal*, vol. 32, no. 1, pp. 25-35, 2017.