Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Method for process-based modeling of combat scenarios using interaction analysis weapon systems*

Dongsu JEONG, Dohyun KIM, Yoonho SEO‡

Department of Industrial and Management Engineering, Korea University, Seoul KS013, Korea

E-mail: jdsvs2979@korea.ac.kr; davydo@korea.ac.kr; yoonhoseo@korea.ac.kr

Received Nov. 26, 2019; Revision accepted June 21, 2020; Crosschecked Feb. 11, 2021

Abstract: With technological advancements, weapon system development has become increasingly complex and costly. Using modeling and simulation (M&S) technology in the conceptual design stage is effective in reducing the development time and cost of weapons. One way to reduce the complexity and trial-and-error associated with weapon development using M&S technology is to develop combat scenarios to review the functions assigned to new weapons. Although the M&S technology is applicable, it is difficult to analyze how effectively the weapons are functioning, because of the dynamic features inherent in combat scenario modeling, which considers interrelations among different weapon entities. To support review of weapon functions including these characteristics, this study develops a process-based modeling (PBM) method to model the interactions between weapons in the combat scenario. This method includes the following three steps: (1) construct virtual models by converting the weapons and the weapon functions into their corresponding components; (2) generate the combat process from the combat scenario, which is derived from the interrelations among weapons under consideration using reasoning rules; (3) develop a process-based model that describes weapon functions by combining the combat process with virtual models. Then, a PBM system based on this method is implemented. The case study executed on this system shows that it is useful in deriving process-based models from various combat scenarios, analyzing weapon functions using the derived models, and reducing weapon development issues in the conceptual design stage.

Key words: Weapon systems; Process-based modeling (PBM); Combat scenario; Interaction analysis; Metamodel; Petri net
https://doi.org/10.1631/FITEE.1900649 **CLC number:** TP391.9

1 Introduction

The changes in combat systems caused by technological advancements have made the task of developing new weapon systems for national survival increasingly complex and difficult. Weapon systems consist of the strategies, tactics, training, materials, and resources (such as tools, supplies, and facilities) that are needed to achieve the objectives of using weapons in combat (Jeong DY et al., 2014; Schwenn et al., 2015). Developing such new weapon systems typically involves four steps: concept development, design verification, prototyping, and evaluation and testing (Park SC et al., 2010). Following these steps requires much time and money, because in each step, individual methods and independent techniques are used. Countries with strong national defenses use modeling and simulation (M&S) technology which begins in the conceptual design stage to reduce the time and cost of weapon development. They also require a combat scenario to reduce the complexity and trial-and-error associated with weapon development. A combat scenario can be used to review the functions of weapons by determining the interactions among various kinds of weapons in advance (Eisner et al., 1991; No and Son, 2005; Tolk, 2012; Li et al., 2013). This approach allows one to preview the functions that weapon systems perform and determine

‡ Corresponding author
* Project supported by the Defense Acquisition Program Administration and Agency for Defense Development of the Republic of Korea (Nos. UD110006MD and UD140022PD)
ORCID: Dongsu JEONG, https://orcid.org/0000-0002-8210-9389

whether they can accomplish their assigned missions against a counterforce before they are constructed (Park SC et al., 2010; USDoD, 2011; Lee PJ and Lee, 2011; Kim TG et al., 2013). Therefore, M&S of weapon systems used in combat scenarios and the modeling of combat scenario itself have been studied intensively so that combat scenarios can be developed efficiently (Park SC et al., 2010; Rainey and Tolk, 2015). An advantage of the M&S of weapon systems is its ability to effectively describe the weapon functions in combat scenarios by assembling virtual weapons. However, modeling new weapon systems for combat scenarios requires new simulation models that consider design methods of each weapon whenever these weapons are used; this is a time-consuming and cost-intensive task. According to the U. S. Department of Defense (USDoD), the time and cost in evaluating weapon systems and correcting their defects using M&S is estimated to represent about 30% of the steadily increasing total development cost (Park SK and Lee, 2003). In addition, modeling research reflecting the unique characteristics of weapons is limited because weapon systems are often developed by reusing the previously developed M&S (Claxton et al., 2005; Kewley et al., 2008; Tolk et al., 2013). A further advantage of combat scenario modeling is the convenience and efficiency brought by reviewing how weapons used in the combat are functioning, taking various combat elements into account. However, such modeling involves difficulties with dynamic features, such as different actions despite the same operational tactics, classifying specific functions (e.g., attack of an aircraft, movement of a tank, and detection of a radar) and different components (e.g., aircraft engine, tank propulsion, and radar detector) of weapons, and analyzing complex interrelations with other weapon entities (e.g., aircraft, tank, and missile) (USDoD, 2011; Lee PJ and Lee, 2011; Kim D et al., 2015).

We have studied the weapon system modeling based on M&S technology for many years. Our research presents a system and methods that enable reconstruction and simulation of new weapon systems by standardizing weapon systems into reusable components (Lee B and Seo, 2014; Kim D et al., 2015; Park HR and Seo, 2015; Jeong D et al., 2016; Seo DJ and Seo, 2016). In the initial phase of our research, weapon system modeling separates the system into its schematic components, builds core assets by selecting components and architectures from the same combat domain, and designs the modeling system to reuse weapons. However, this approach has shortcomings, particularly the previously mentioned problems related to the combat scenario: (1) The development of simulation models for new weapons requires trial-and-error time and cost because each weapon uses the simulation model with a limited structure. (2) It is complicated to design weapons and weapon functions consisting of many other entities. (3) Comprehensive analyses of weapons used in the combat are difficult because the interoperability between heterogeneous simulation models is limited. To solve these problems, a new method is needed to support an effective review of the weapon functions by quickly identifying the complex interrelations among the weapons prior to the development of the weapon system. To meet this need, in this study, we develop a process-based modeling (PBM) method to generate a combat scenario in which the weapon functions and the mission completion are reviewed by modeling the interactions between weapons. This method includes the following three steps: the first step is to construct virtual models by converting the weapons and the weapon functions into their corresponding components; the second step is to quickly generate the combat process from the combat scenario, which is derived from the interrelations among weapons under consideration using reasoning rules; the final step is to develop a process-based model describing the weapon functions by combining the combat process with virtual models.

We focus mostly on combat scenario modeling at the engagement level, and contribute to the M&S technology in three ways: first, we design a method to assign the specific functions corresponding to the weapon structure; second, we provide a new modeling method that depicts the combat scenario as a series of processes by analyzing the interactions between weapons and weapon functions; third, we present the metamodeling in which different concepts and methods for development of weapons are interconnected to provide an integrated model.

2 Related work

In this section, we investigate two key issues of the PBM method: combat entity modeling and combat scenario modeling. These issues describe how to model the structural and behavioral elements of weapons and how to describe combat scenarios using modeled weapons to address the shortcomings presented in Section 1. These issues are currently a focus of the field of combat M&S.

2.1 Combat entity modeling

Most of DoDs studied the combat models that involve elements of physical and behavioral aspects by defining the work breakdown structure (WBS) of entities used in combat (DAPA, 2012; Tolk, 2012; Tolk et al., 2013). The one semi-automatic force (OneSAF), a part of the best-known combat model, describes combat situations to simulate the command and control of units and members through the component-based development (CBD) technology, which is a software method for creating new systems and applications by combining components (Wittman and Harrison, 2001). In developed countries, studies have been conducted on componentization and standardization using the CBD technology to reduce weapon development costs and reflect users' needs (Lee JO et al., 2010). This approach of combining components allows the CBD technology to virtually organize the varied and differentiated combat entities needed by users. Implementing a virtual combat entity requires the entity to be structured into one system that includes not only the physical properties but also the behavioral properties of that weapon (Seo KM et al., 2017), and CBD is appropriate for these studies.

However, it has been difficult for component-based combat model studies to obtain behavioral information from various functions (Tolk et al., 2013; Seo KM et al., 2017). As a solution to this issue, Kim D et al. (2015) proposed a systematic design model that describes the dynamic relationships of weapons using the function–behavior–structure (FBS) method, which designs models based on the function, behavior, and structure correlation. In addition, using ontology techniques which create semantic aggregations centered on unique functions and interrelations between knowledge structures (Kim D et al., 2005), Park HR and Seo (2015) and Seo DJ and Seo (2016) conducted research into recognizing and reconfiguring similar weapon functions based on physical and behavioral components. These studies have the advantage of being able to quickly construct virtual combat entities, taking user requirements into account.

In contrast, as a way of expressing the mission of a weapon rather than the nature of that combat entity, Lee B and Seo (2014) conducted a study that visually modeled the mission procedures that weapons should perform. This approach has the advantage of being able to investigate the operational performance of combat entities with a simple combination of only resources and mission procedures. However, these studies alone are inadequate for modeling a combat entity with a unique mission (such as a flight of exploration with steering), and defining the interrelationships among pieces of the wide-range information possessed by combat entities is challenging. Therefore, in Section 3.1.2, we will present a design method that creates a combat entity composed of much diverse information and the method to express its mission.

2.2 Combat scenario modeling

To counter new enemy weapons, researchers focus on the effect and impact that weapons have on the battlefield, taking into account the interrelationships among weapons with unique capabilities, such as flight with evasive steering. Therefore, research has been conducted actively on combat scenarios that can simulate combat situations including various factors such as combat environments (i.e., air, land, and sea), combat rules (i.e., reconnaissance, attack, and defense), and weapon performance (No and Son, 2005; Tolk, 2012; Kim TG et al., 2013; Choi J et al., 2015).

The most representative way of expressing a combat scenario is using military scenario definition language (MSDL) and coalition battle management language (C-BML) (Tolk, 2012). MSDL is designed to provide the M&S community in military scenario development with a common mechanism for verifying and loading military scenarios (SISO, 2008). C-BML is an emerging international standard for the unambiguous expression and exchange of plans, orders, and reports across command-and-control systems, simulation systems, and robotic systems. The benefit of MSDL and C-BML is that they provide an expressive formal basis and a structure for the combat scenario. Various studies have been conducted using these languages to create a combat scenario model (Zhao et al., 2012; Jafer and Durak, 2017). However, MSDL and C-BML have difficulty in expressing system engineering artifacts that further clarify the behavioral simulation process of combat entities.

Also, they suffer from the limitation that it is difficult to verify the performance of missions according to the state changes of combat models with physical and behavioral information through simulation. To solve these problems, using event-based modeling methods that can track the status changes of models, methods of expressing the artifacts of the combat scenario model have been studied, such as the unified modeling language (UML), Petri net, business process management notation (BPMN), and discrete event system specification (DEVS) (Kim TG and Moon, 2012; Choi BK and Kang, 2013). Qi et al. (2006) visually modeled the attack and defense courses of ballistic missiles by describing the structural and action models of the entities using UML. Özhan et al. (2008) presented a formalized conceptual model and its metamodeling to describe military field artillery tasks, including their physical and behavioral parts, using live sequence charts. These studies can quickly describe simple combat models; however, they are limited in their ability to meaningfully express complex combat situations. To deal with these issues, Li XB et al. (2013) presented a method to easily describe engagement situations using Petri net. It is a set of graphical and mathematical tools for modeling complex and dynamic systems. Seo KM et al. (2014) presented a modeling method to demonstrate military simulation including combat objects or entities at the engagement level using DEVS. Luo YL et al. (2016) studied the modeling method to describe the air command and security (ACS) process, presenting the complex operation process as encapsulated flows in coarser granularity using BPMN. The biggest difference between the event-based modeling and other modeling is that the event-based modeling can describe the changes in the state of a combat model with behavioral information according to the event flow. Combat scenarios should be able to express the missions of various weapons, and event-based models are applicable to the characteristics of such combat scenarios.

However, this type of combat scenario modeling has difficulty in describing the combat model's physical and behavioral information, and it is difficult to define the interoperability between simulated combat models. This situation limits comprehensive review of the objects used in combat. To overcome these issues, in Section 3.3, we will present a new way to describe the combat scenario model as a process with a series of event flows by analyzing the interactions between weapons.

3 Process-based modeling method

The PBM method can describe and analyze combat scenarios as a series of processes through interactions between weapons and their functions. To address the problems caused by characteristics of combat scenario modeling presented in Section 1, we studied the PBM methods based on the related research in Section 2 to determine the following four requirements: (1) Combat scenario modeling should systemically comprise many weapon elements. (2) Combat scenario modeling should logically describe the interrelations among the dynamic features inherent in the combat scenario model. (3) Combat scenario modeling should analyze how effectively the weapons are functioning. (4) Combat scenario modeling should review how properly the weapons and the weapon functions are used in the combat. This concept is illustrated in Fig. 1. The core of this method is a set of PBM steps. In the middle of Fig. 1, PBM shows the key modeling steps that enable the development of the process-based model; i.e., resource modeling designs weapons in physical and behavioral components using CBD, and generates resource models that depict combat actions by combining these components using FBS.

Functional module modeling, to create the functions of the resource model, generates the functional modules (FMs) that describe weapon functions using common functional elements (CFEs) derived using ontology technology. Then, FM is combined with the resource model to generate an encapsulated model (EM). To analyze EM, combat scenarios are needed.

Process modeling generates combat scenarios of allies and enemies from standard norms, and generates combat processes to review weapon functions according to the sequence of weapon utilization in combat scenarios. This modeling is used to design combat situations by assigning EMs to the combat process and then generating a process-based model of a metamodel with the ability to analyze the functions of the models.

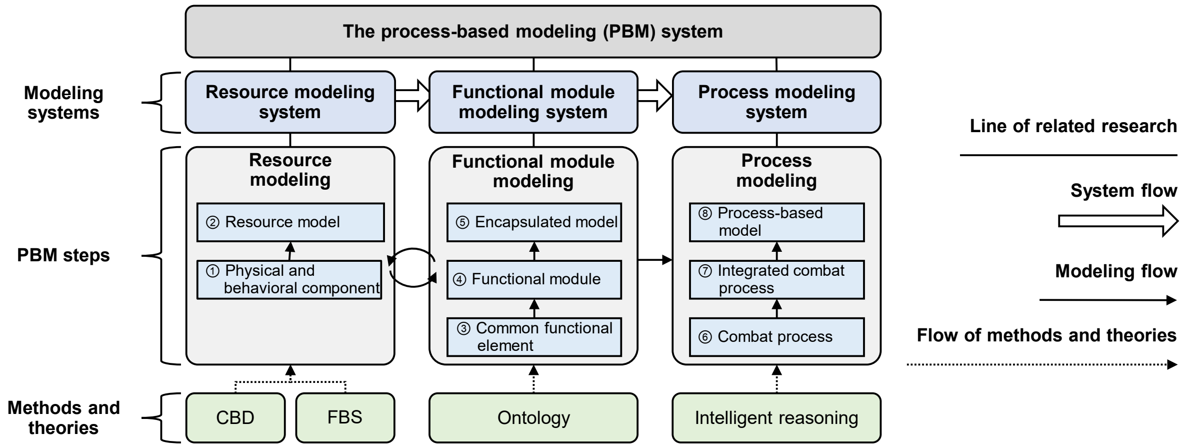

Fig. 1 Overall flow of the process-based modeling method (CBD: component-based development; FBS: function–behavior–structure)

The methods and theories provide the necessary technologies and methods to implement process-based models. Modeling systems can be implemented as separate systems at each modeling phase, and integrated ultimately into a complete system, the PBM system. The PBM method has been studied using several methods presented in Section 2. Informed particularly by the technical ideas related to evaluating weapon functions to virtually model weapons by Lee B and Seo (2014) and Luo et al. (2016) in the event-based modeling method, we introduce a new modeling simulation method that can design weapons and their functions systematically, by modeling combat scenarios with processes as a series of event flows using interactions between weapons. The PBM method can model and review combat scenarios to represent entities, interrelations, and model challenges.

3.1 Resource modeling

Resource modeling is a step in which a combat entity is created. It takes into account the physical and behavioral elements of the weapons' information that constitutes the weapon systems. The combat entities of weapons at this step are based on the USDoD's WBS (USDoD, 2011; DAPA, 2012). The combat entity comprises physical and behavioral aspects. The modeling of the physical aspect relates to the internal physical mechanisms in the physical and information domains, and the behavioral modeling (i.e., the intelligent aspect of domain modeling) relates to the tactics of how to use combat models under particular circumstances. Physical modeling is the main concern of modelers, and behavioral modeling is usually affiliated technically to physical modeling (Li et al., 2013; Seo KM et al., 2017). During realization, the virtual combat entities should be structured as one system incorporating not only the physical but also the behavioral functions; for this, CBD is appropriate. Because CBD by itself has limited ability to express the unique behavioral functions of weapons caused by changes in combat environments and tactics, an additional FBS technology is used to describe these behavioral functions (Kim D et al., 2015).

3.1.1 Physical and behavioral component

To generate weapon functions, a target model describing weapons in great detail is necessary. The model, which includes physical and behavioral information, is generated through combined information. The physical components present hardware characteristics, showing physical information such as engines and wings. The behavioral components present software characteristics, showing behavioral information such as actions and movements. Fig. 2 shows the physical and behavioral components based on CBD (① in Fig. 1). These components were created referring to the WBS of USDoD, and we present six representative weapons (Lee JO et al., 2010; DAPA, 2012; Jeong DY et al., 2014; Kim D et al., 2015).

3.1.2 Resource model

The resource model (② in Fig. 1) is to represent both the physical and behavioral information by

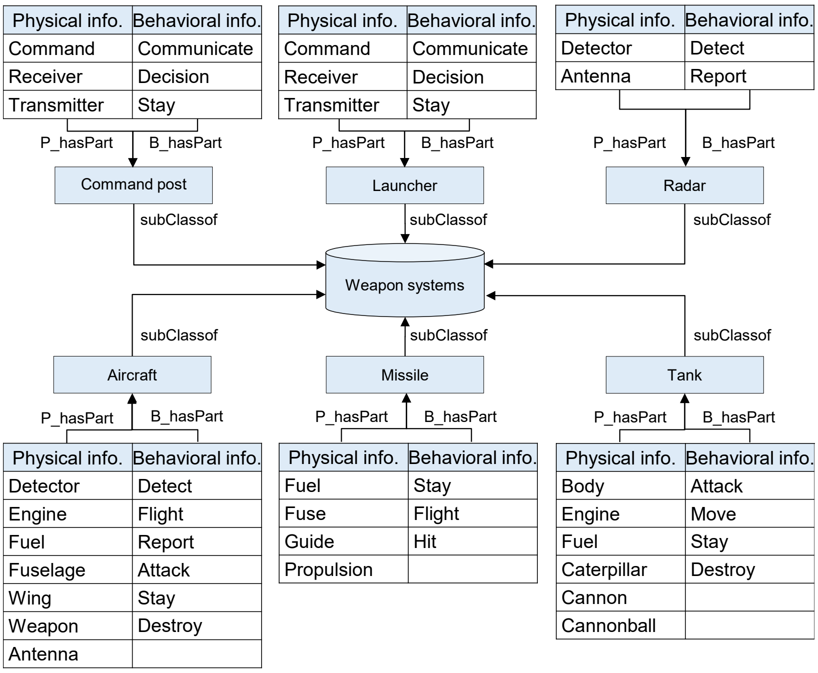

Fig. 2 Physical and behavioral components

combining the physical and behavioral components. The physical information of the resource model indicates the physical characteristics and static form of the hardware function of weapon entities. The behavioral information indicates the behavioral characteristics and the dynamic form, along with the software function of the weapon entities; its accuracy is limited when describing behavioral characteristics using only physical components. Therefore, the resource model uses a design method based on the FBS technology, which can derive the behavioral characteristics of weapons from their physical characteristics (Lee JO et al., 2010; Kim D et al., 2015). To design the FBS, the FBS interrelations are defined using the attributes of the weapons' physical and behavioral components. For instance, the steps presented in Fig. 3 explain the FBS procedures of generating a resource model for aircraft as follows:

①→②: First, after setting the purpose of "moving and attacking the enemy using aircraft," select these aircraft functions, such as attack activity, detect activity, and flight activity.

②→③: Select the function, and then create the behaviors of the aircraft, such as attack activity, detect activity, and flight activity.

③→④: The created behavior can deduce the structure, such as Wing, Fuel, Engine, and others. If the air-to-air missile (AAM) is selected as the attack activity, "1. High-wing" which is capable of high-altitude flight should also be selected.

①–⑤: The aircraft's behavioral information can be represented by creating the resource model through FBS procedures for a specific mission of the weapons.

3.2 Functional module modeling

Functional module modeling is a step in combining the behavioral functions to express the missions of a weapon. The behavioral functions represented in this step are elements that various weapons can use in common. These elements can be structured as one module using the ontology technology. Currently, ontology-based combat modeling has been used in various studies (Park HR and Seo, 2015; Seo DJ and Seo, 2016; Jafer and Durak, 2017). We have studied ways to express the mission of weapons as a flow of behavioral elements (Luo et al., 2016).

3.2.1 Common functional elements

Resource models contain complex and varied behavioral information for performing some inherent weapon missions, such as the reconnaissance information of the aircraft and the detection information of the detector. This behavioral information includes

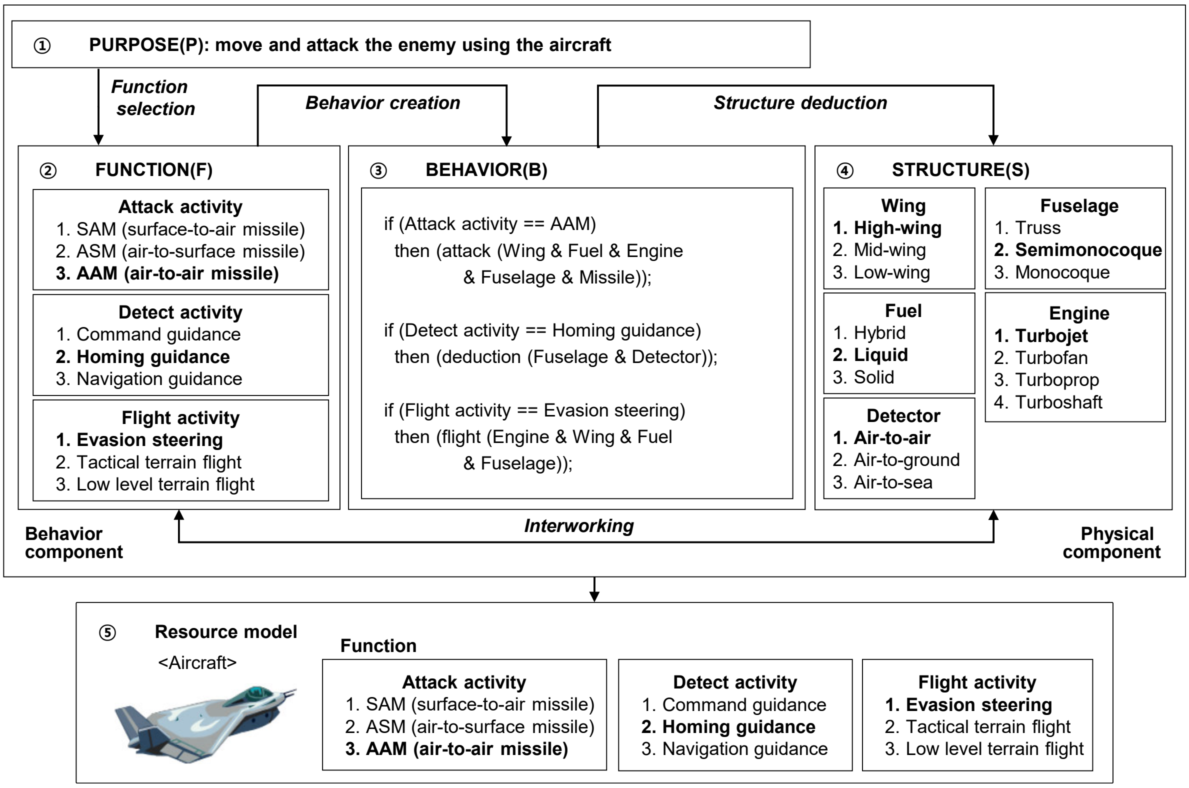

Fig. 3 An aircraft creation example based on the function–behavior–structure method

separate behaviors and appellations of the weapons. For instance, although the movement of a tank and the flight of an aircraft have physically different behaviors, they share the common function of changing position. In weapon systems, the common function is referred to as a common element and is used in integration, systems engineering, systems testing and evaluation, and training (USDoD, 2011). Although common elements can hold the behavioral information of different weapons, it is difficult to classify varied functions and similar behaviors. Therefore, we categorize the behavioral information in resource models using ontology technology and defined CFEs, which replace the granular form of common components by classifying components from all resource models (③ in Fig. 1) (Kim CO et al., 2005; Park HR and Seo, 2015; Luo et al., 2016). Fig. 4 explains the generation process of CFEs and the relationship among them using ontology. First, resource models are classified into physical and behavioral components. Then, nine CFEs are derived through a combination of the common functions of the components. CFEs present the specific functions that the resource models perform based on forward and reverse interactions. These functions are the most basic and essential information for weapons analysis.

3.2.2 Functional module

The weapon functions are presented as a combination of various behaviors to achieve the weapons' goals, such as accomplishing an aircraft reconnaissance mission. This is replaced with the combination of CFEs that define an FM (④ in Fig. 1). An FM specifies the roles of the elements involved in performing specific missions such as detection, attack, and evasion. FM also presents CFEs listed in a series of flows to describe the functions. Within the same module, FM can be generated by combining different CFEs, which means using six FMs (reconnaissance, detection, decision, attack, destruction, and launch). Fig. 5 explains the procedures of creating other FMs out of CFEs. Using FMs for the detection function, reconnaissance module 1 uses a dynamic resource model (such as an aircraft or a tank) to process the BPFs' Move, Detect, Attack, and Stay, while reconnaissance module 2 uses a static resource model (such as a command post or a radar) to process the BPFs' Stay, Transmit, Detect, and Report.

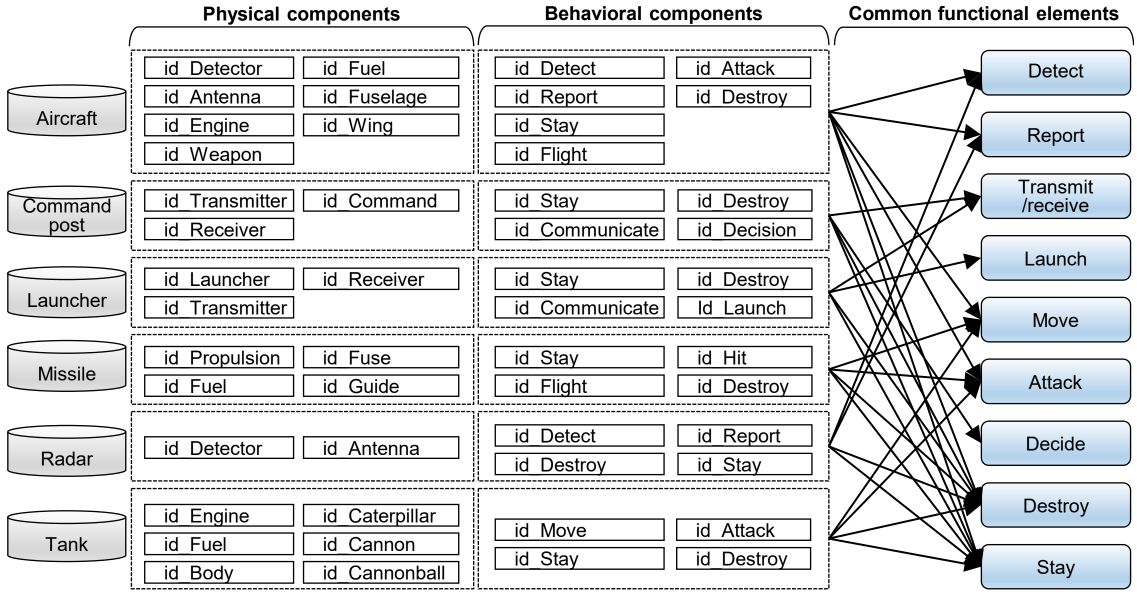

Fig. 4 The generation of common functional elements from resource models

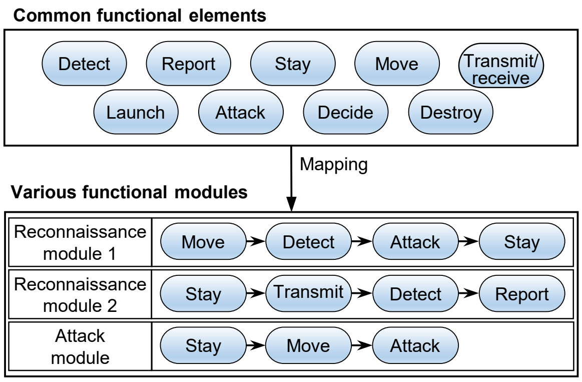

Fig. 5 An example of the creation of functional modules

3.2.3 Common functional elements

To generate the resource model reflecting inherent weapon missions, the main object must present both the resource model and FM. Thus, we create a combined object EM (⑤ in Fig. 1), for analyzing the weapon function by linking the resource model and FM. An EM is an encapsulation that interconnects the physical and behavioral components of a resource model and the BPFs of an FM. The created EM involves the specific functions. It is used as the key model for analyzing the weapon functions used in combat scenarios. Fig. 6 depicts the procedures of creating EM by combining the aircraft shown in Fig. 3 with reconnaissance module 1 shown in Fig. 5. First, we classify the physical and behavioral components and CFEs to link an aircraft and a reconnaissance module. Then, we connect the physical components (id_Wing, id_Fuel, id_Engine, and id_Fuselage) and a behavioral component (id_Flight) of an aircraft to connect a BPF (move). Other physical and behavioral components connect other CFEs.

3.3 Process modeling

Process modeling is a step in creating a combat scenario model that reviews the interrelationships between the functions of the weapons used in the combat. The combat scenario model uses a variety of information (Tolk, 2012; Zhao et al., 2012; Seo KM et al., 2017), including the weapons, tactics, and environments of friend and enemy forces. Only when the interactions between the weapons are reflected, can the semantic scenario model be designed. In this subsection, the process-based models that depict the functions of weapons as a series of processes are studied (Lee B and Seo, 2014).

3.3.1 Combat process

To accurately determine the combat situation, the various weapons and their functions require a combat scenario that can confirm what their mission is and how they perform it (Tolk, 2012; Zhao et al., 2012). Because combat scenarios are characterized generally by numerous wide-range variables, it is difficult to confirm the mission and weapons used in

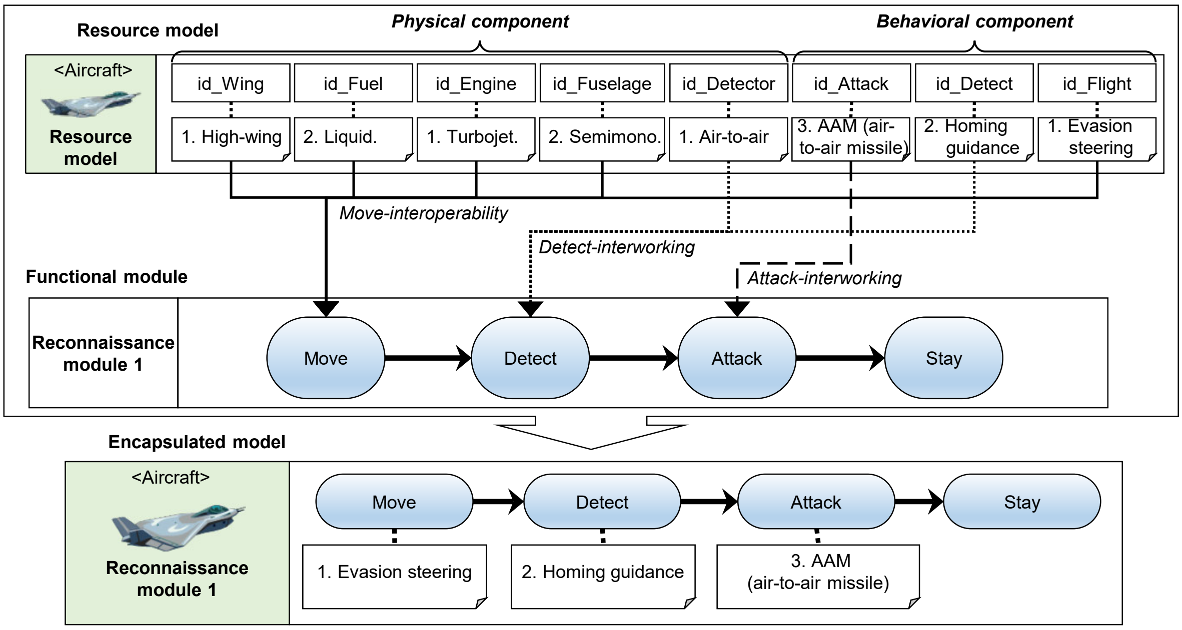

Fig. 6 An example of an encapsulated model combining a resource model and a functional module

the combat unless they are exactly the ones involved. To solve this problem, we study a standardized battle scenario model by referring to an ontology-based battlefield scenario study describing the mission of weapons with specific functions in combat using mission, enemy, terrain and weather, troops, the time available, and civil considerations factors (METT-TC) representing the canonical, militarily significant information for military evaluation and decisions (USDA, 2007; Jeong D et al., 2016; Oh et al., 2016). We propose a combat process that describes a series of weapon sequences considering weapon functions to achieve the mission goals of combat scenarios (⑥ in Fig. 1). However, further research using MSDL is needed to implement formal descriptions of combat scenarios. Therefore, the combat process is studied in consideration of integration with MSDL that will be promoted in the future.

Fig. 7 presents three steps for generating a combat process. The first step is to convert written field manuals in PDF, Word, or Hangul Word Processor (HWP) into a text format to extract the combat scenarios. The second step is to extract the combat scenario from the converted text file according to data transformation based on the when, who, when, where, what, and how (5W1H) procedures. The second step generates the missions of a combat scenario, and the generated missions are used to extract a combat process through ontology-based scenario generation (Jeong D et al., 2016). This combat process has only the basic names of weapons and their functions. Finally, the third step is to save the combat process in a database. This subsection presents the combat process that expresses the mission of the combat entity in a series of flows; however, further research using MSDL is required to implement the formal description of the combat scenario.

3.3.2 Integrating combat processes

Checking the weapons functions used in combat requires scenarios of not only the friendly side but also the enemy that counters them. For instance, a complex mission (such as “destroy the enemy tank using aircraft”) is described more realistically when the combat scenarios for both the friendly aircraft and the enemy tank are interconnected. An integrated combat process (ICP) is generated to connect the combat processes representing both the friendly and enemy combat scenarios (⑦ in Fig. 1). ICP indicates a new combat scenario in which weapons with specific functions are used and new combat processes are rendered by renewing the interaction of EMs used in different combat processes. The integration of combat processes can be realized through intelligent reasoning and deducting meaningful information through applying intelligent reasoning to the data available.

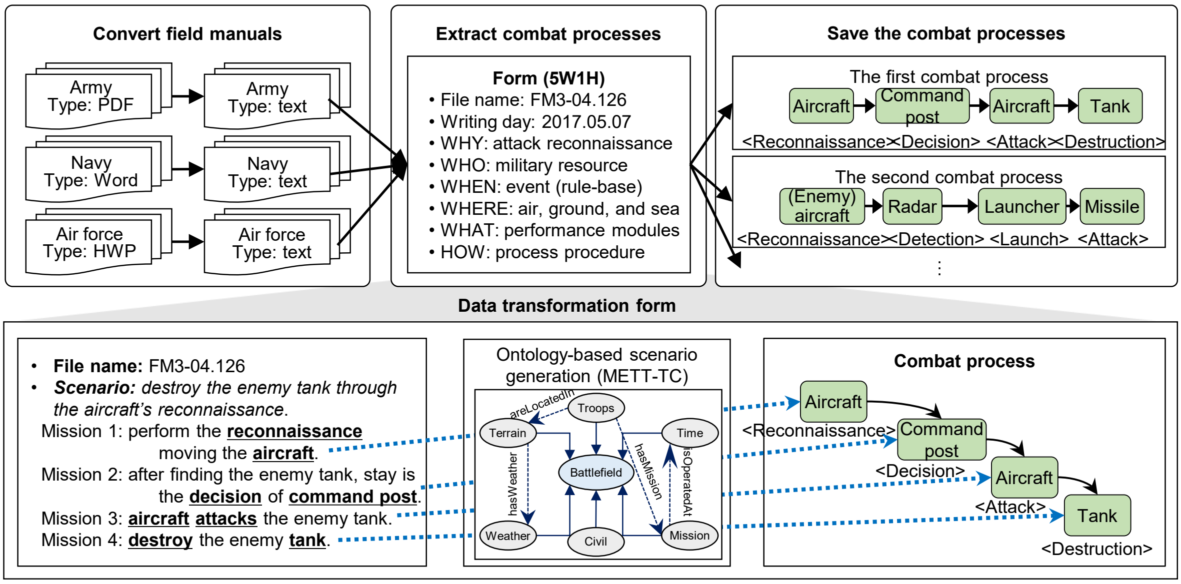

Fig. 7 An example of the creation of a combat process

To generate ICP, we deduce the interrelation between EMs and the interaction between BPFs to apply two reasoning rules based on intelligent reasoning technology. The intelligent reasoning based cognitive solutions (i.e., decision that models the intelligent aspect) comprise four categories: built-in, table-based, rule-library-based, and script-based modeling (Li et al., 2013). To define the interaction analysis, we use the table-based and rule library based modeling. The table-based modeling is used with text files that describe the tactics that define the rules by referring to the decision table to select between the alternative possible tactics used under specific conditions (Son et al., 2010). The rule-library-based modeling is used to build a rule library, dynamically questioning that library, and finding the matching rules. These rules can be revised at runtime and the revisions take effect immediately; thus, this approach is efficient when modeling alternative decisions (Li et al., 2013). Rules of interaction analysis between EMs and CFEs and the algorithm for ICP are shown as follows.

1. Rule of interaction analysis between EMs

The rule of interaction analysis between EMs is for determining the overall EM order by examining the relations between EMs that make up different combat processes using table-based modeling. This rule can be instantiated to infer the order of the EMs in Table 1 using the first and second combat processes shown in Fig. 7. First, to check the existence of overlapping EMs in the combat processes, "O" is entered in the columns of combat process data in Table 1 whenever EMs are used. Second, based on combat processes, for all subsequent EMs (Table 1) that can follow the prevenient EMs (Table 1), "O" is entered in the columns of precedence between EMs. For example, a preceding EM_0 in combat scenario #1 deduces the following EM_1, EM_2, and EM_3, while a preceding EM_0 in combat scenario #2 deduces the following EM_4, EM_5, and EM_6. Finally, the forward and reverse interactions between EMs are replaced in a from–to sequence to represent columns of a series of processes. This sequence is a way to preferentially find the fewest prevenient EMs after selecting the last subsequent EM. For example, if the last subsequent EM is a tank, an aircraft (with "attack") is selected to reach the minimum condition among the prevenient EMs (an aircraft and a command post). For this rule, it helps to have a series of processes for EMs, such as an aircraft (with reconnaissance), a command post (with decision), an aircraft (with attack), and a tank (with destruction).

2. Rule of interaction analysis between CFEs

The rule of interaction analysis between CFEs using rule library based modeling is an if–then rule for providing interconnectivity between the CFEs that compose the EMs provided in the column of the series of processes in Table 1. This rule creates connectivity between CFEs, represented by three conditions for Definitions 1 and 2.

Table 1 An example of interaction analysis between encapsulated models

Data of combat processes (Fig. 7)					Precedence of EMs								Series of processes	
Combat scenario		Encapsulated model: $EM_n^{r,f}$			Prevenient EM_n	Subsequent EM_n								
#1	#2	Resource model (r)	Functional module (f)	EM_n		EM_0	EM_1	EM_2	EM_3	EM_4	EM_5	EM_6	From	To
O	O	Aircraft	Reconnaissance	0	EM_0	–	O	O	O	O	O	O	(EM_6)	$EM_{1,6}$
O		Command	Decision	1	EM_1		–			O		O	EM_0	EM_2
	O	Radar	Detection	2	EM_2			–	O		O		EM_1	EM_3
	O	Launcher	Launch	3	EM_3				–		O		EM_2	(EM_3)
O		Aircraft	Attack	4	EM_4					–		O	EM_0	EM_5
	O	Missile	Attack	5	EM_5	(O)					–		EM_4	EM_6
O		Tank	Destruction	6	EM_6							(O)	EM_5	(EM_0)

"O" refers to the possible entering data; "(O)" means the precedence of EM_n may be the possible entering data or not

Definition 1 (1:1 rule) The 1:1 rule refers to the relation in which one element interacts with only one specific element. It is determined by locations and types of input and output elements through the following three conditions:

(1) The start condition of elements includes the Transmit node.

(2) The second condition of elements includes the Receive and Transmit nodes, prioritizing with the Receive node and followed by the Transmit node.

(3) The end condition of elements includes the Receive node.

For instance, an output CFE Transmit node and an input CFE Receive node have a predefined 1:1 relation, connected through three conditions. If there is no CFE reception to be received after the CFE Transmit node, either the rule of the predefined CFEs or the N:1 rule should be followed.

Definition 2 (N:1 rule) The N:1 rule refers to the relation in which elements do not apply the 1:1 rule when interacting with a particular element. It is also determined by the locations and types of input and output elements through the following three conditions:

(1) The start condition of elements includes the output nodes: Report, Detect, Attack, Move, Launch, Decide, and Transmit.

(2) The second condition of elements includes various nodes, prioritizing with the Stay node.

(3) The end condition of elements includes the Stay node.

For instance, if there is no input CFE Receive node that should follow an output CFE Transmit node, there is an N:1 relation using the CFE Stay node, and the relation is connected through three conditions. If the 1:1 rule and N:1 rule are applicable at the same time, the 1:1 rule is followed preferentially.

3. Algorithm for ICP

Based on the two interaction analysis rules, Algorithm 1 is built to automatically generate an ICP. For ICP generation, Algorithm 1 implements recursive tasks to connect two combat processes as an ICP. The first task is to find two combat processes that correspond to the friendly and the enemy combat scenarios. The second task is to select EMs that correspond to each combat process. It is assumed that the order of the FMs composing the EMs is predetermined by users. The third task is to define the mutual interaction between the EMs of two combat processes using rules for interrelation analysis between EMs. To interconnect the CFEs that compose the EMs, the fourth task is to select the CFEs that compose the FMs of the EMs. Next, the fifth task is to define the interconnection between the CFEs using the rules for interrelation analysis between CFEs. Finally, the sixth task is to generate the ICP. Algorithm 1 is used to automatically generate a process-based model.

Algorithm 1 Creation of an integrated combat process based on interaction analysis

Initialize: resource model r, CFE_i, functional module f, $EM_n^{r,f}$, CS_i, CP_i, and ICP_i.

while CP_i is selected for realizing the purpose of CS_j **do** // The first task
  if (CP_i==CS_j) **then**
    Add conditions of CP_i corresponding to CS_j;
  end if
  Select CP_i with the condition of CS_j;
end while
while CP_i is selected **do** // The second task
  Add the conditions of r corresponding to CP_i;

Add the conditions of f corresponding to r;
if ($EM_n^{r,f}$ ==r and f) **then**
Add the conditions of $EM_n^{r,f}$ corresponding to r and f;
end if
end while
Add $k\cdot CP_i$; // k is the prevenient $EM_n^{r,f}$ (denoted as a) or // the subsequent $EM_{n-1}^{r,f}$ (denoted as b)
while $EM_n^{r,f}$ is selected **do** // The third task
Load $EM_n^{r,f}$ to meet the conditions of CP_i;
for each $EM_n^{r,f} \in CP_i$ **do**
Set b ($EM_{n-1}^{r,f}$) to accomplish the final mission of CP_i;
Select all of a ($EM_n^{r,f}$) to carry out the next mission except b which is set to perform the final mission;
Apply the precedence conditions corresponding to a and b;
Create a table for a series of processes according to the precedence conditions;
end for
end while
while $EM_n^{r,f}$ is selected **do** // The fourth task
for the $EM_n^{r,f}$ loaded following the series of processes in the regular sequence
Load CFE_i corresponding to $EM_n^{r,f}$;
Select all CFE_i's corresponding to a and b;
if (the 1:1 If–Then rule==the relation between a and b) **then** // The fifth task
Apply the 1:1 If–Then rule;
else
Apply the N:1 If–Then rule;
end if
end for
Create ICP;
Create an ICP_i which has all $EM_n^{r,f}$'s following the series of processes; // The sixth task
end while
Return the result.

3.3.3 Process-based model

The combat scenario should express the intimate relationships of weapons corresponding to the friendly and enemy combat scenarios as logical procedures (⑧ in Fig. 1). A process-based model is an abstract model that closely combines an ICP (as determined above) with the EOs of the enemy and friendly forces (Fig. 8). It interconnects all the CFEs for all the EOs, and includes the purposes of the combat scenarios for different enemies and friendly forces (shown as #1 and #2 in Table 1, respectively). Fig. 8 depicts a process-based conceptual model that interconnects EMs to achieve the goals of the friendly forces (destroying the enemy tank using the stealth aircraft's reconnaissance function) and enemy forces (detecting enemy aircraft using a radar, and then using a missile attack) corresponding to ICP.

4 Process-based modeling system through metamodeling

Based on the PBM method, we propose a process-based model, which is a conceptual model to effectively describe combat scenarios. However, because the process-based model does not provide a formal description, the model needs the formal systems model for users and developers to understand it. In this section we design a metamodel so that others can use the PBM method to analyze the metamodel in public simulations. This procedure is carried out as mathematical formalization for easier analysis of the metamodel.

4.1 Unified modeling language based metamodeling for process-based modeling

The reason that the PBM method is designed as metamodeling is to express a method for combat modeling and the different syntaxes for weapon system development steps in a formal language. A metamodel is generally a model that defines the structure of a modeling language. Metamodeling is responsible mainly for providing a modeling language to specify the metamodels and the abstract syntax of combat modeling (Li et al., 2013, 2017). This study focuses on the design of a PBM that captures the combat scenario and relationships of weapons and their functions. Based on the aforementioned PBM abstraction, we build a metamodel (Fig. 9) that transforms the PBM method using the metamodeling language of UML class diagram. UML is a set of graphical notations based on a metamodel for specifying, visualizing, and documenting object-oriented software systems and is formally defined using a metamodel (OMG, 2003; Alshayeb et al., 2016).

The core concepts of the metamodel are EM, ICP, and rule interaction analysis, i.e., classes of class

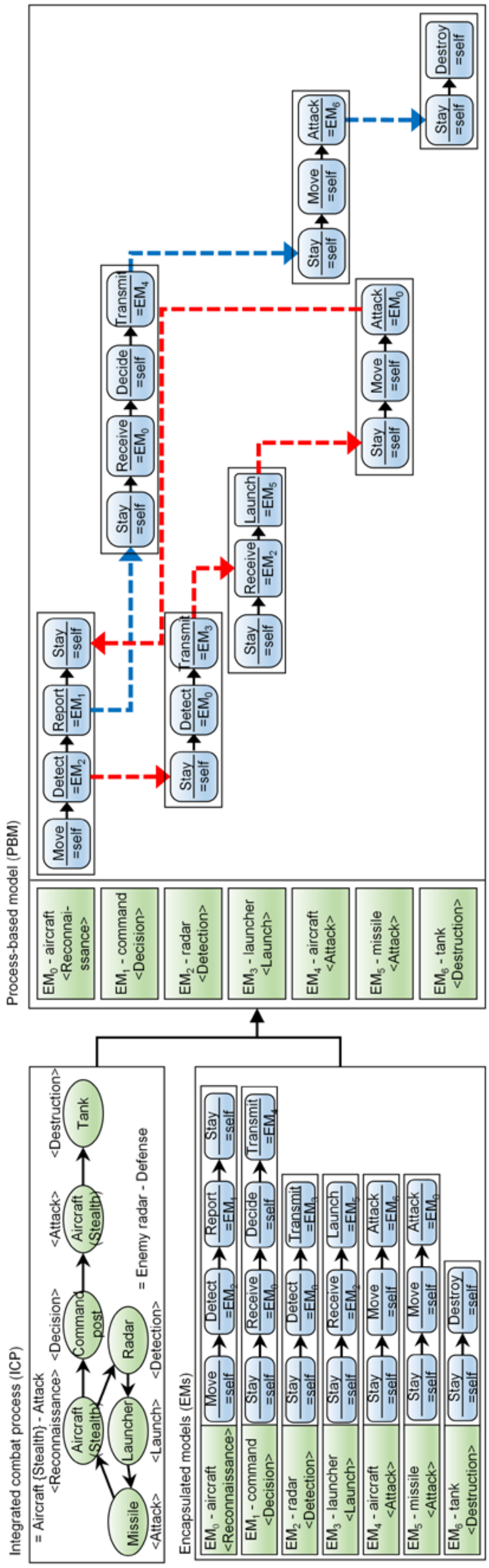

Fig. 8 An example of a process-based model combined with an integrated combat process with encapsulated models

diagrams such as models, processes, and rules. The model is the combat model of a weapon, and attempts to achieve the mission as a series of processes corresponding to ICP. Each model has physical and behavioral information and performs the specific function of the weapon. To build the model, it should have classes designed by the resource model and the FM described in Section 3.1.2. The process is the combat process which represents a sequence of tactical actions based on a series of combat processes by transforming combat scenarios derived from the standard norms of weapon systems. Implementing this process requires not only the combat process structure but also the relation between the detailed elements, which means that it should have classes designed according to combat processes and the rules of interaction analysis presented in Section 3.3.4. These rules are essential for designing PBMs, which are based on the rules of interaction between EMs and between CFEs. All of the model elements that occur in the conceptual model must be derived from the metamodel elements shown in the diagram. The metamodel elements are shown as classes and the relationships among them are shown as UML associations, aggregations, or generalized relations.

4.2 Process-based modeling system

We propose three modeling parts for the PBM method (Fig. 1) and design the UML-based metamodel in Fig. 9. As the metamodel provides only the abstract syntax, we build a system using the concrete syntax of the metamodel. Originally, systems in each part should be introduced for each part; however, since this study focuses on how to describe and analyze the weapons used in combat scenarios, it introduces only a system that generates PBMs. For simplicity, resource modeling and FM modeling systems are not explicitly introduced, and previously studied systems are referenced (Lee B and Seo, 2014; Kim D et al., 2015; Oh et al., 2016; Seo DJ and Seo, 2016; Jeong DS and Seo, 2018).

Fig. 10 illustrates the purpose of the combat scenario; EMs define the forward and reverse order, and CFEs define the interactions and data structure of a PBM for model generation. "Set up the combat objective" involves sequentially entering the sequence of resource models and the type of the resource models and functional modules according to

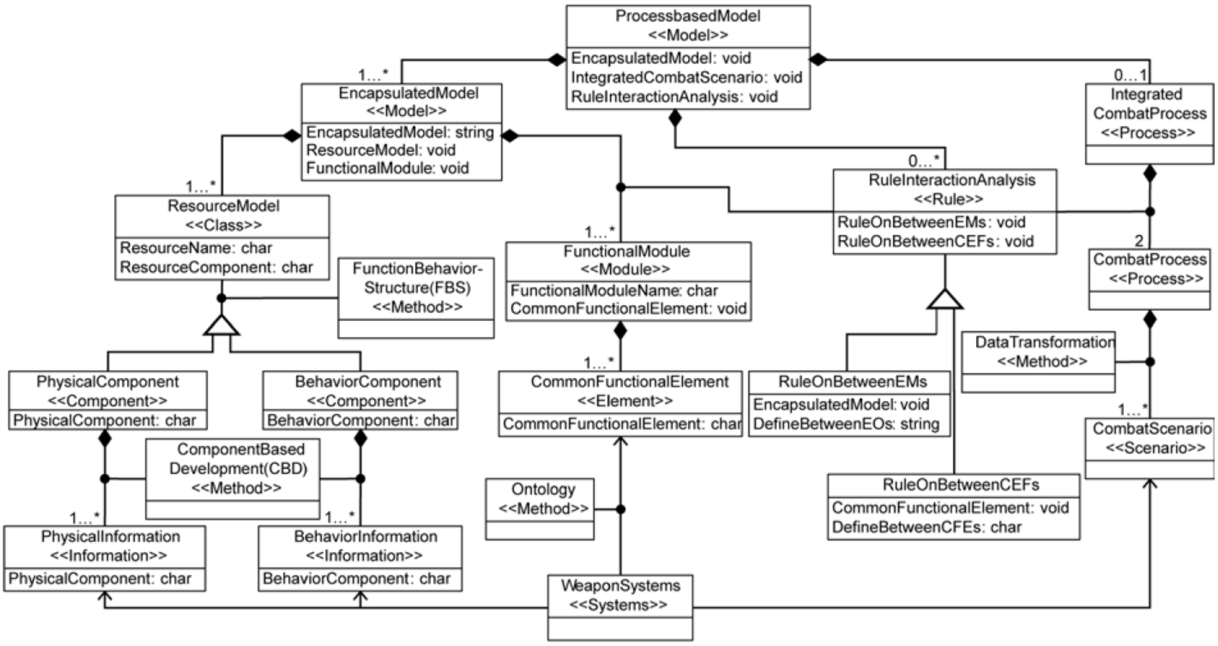

Fig. 9 UML class diagram based metamodel of the process-based modeling method

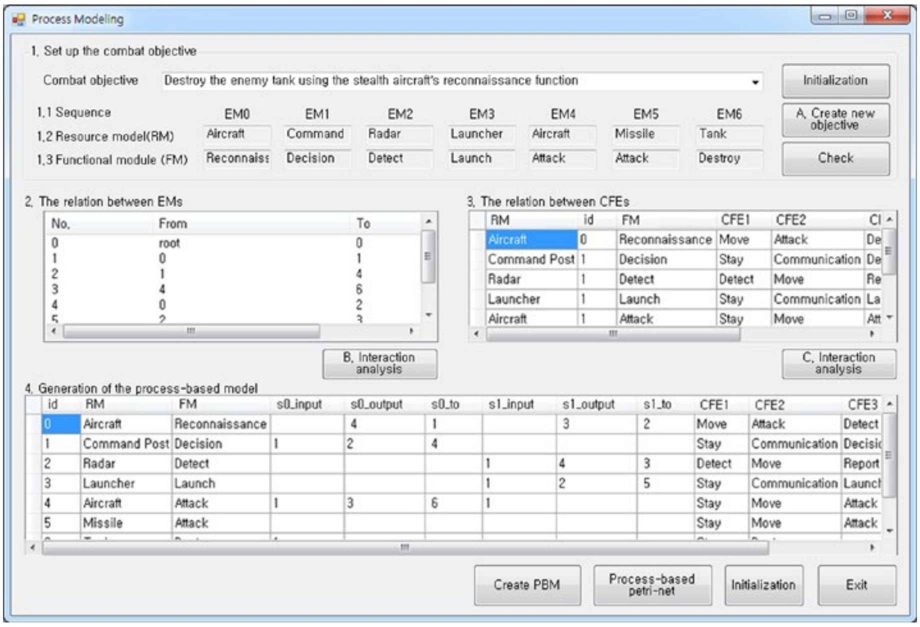

Fig. 10 The process-based modeling system for the integrated combat process

the predefined purpose of the integrated combat scenario. "The relation between EMs" involves listing the EMs corresponding to the resource models and function modules of "set up the combat objective" and entering the prevenient EMs in the "From" column and the subsequent EMs in the "To" column. These columns show EM sequences reasoned according to the rules of interaction analysis between the EMs presented in Section 3.3.2. "The relation between CFEs" involves defining interactions between the CFEs of EMs and entering the information corresponding to each column. These columns show the interaction between CFEs according to the rules of interaction analysis between CFEs in Section 3.3.2. "Generation of the process-based model" describes the data structures of PBM combined with an ICP with EMs. First, information in the columns, such as the resource model, FM, and elements, is automatically fed through the three steps entered previously. Then, the columns sn_input, sn_output, and sn_to show the position of the CFE receiving the execution message when an EM is executed, the position of the CFE delivering the message after an EM is executed, and the number of EMs running as a result of EMs, respectively. This step generates a PBM by entering information using Algorithm 1.

4.3 Process-based Petri-net model for mathematical formalism

A PBM conceptually describes weapon interactions; however, it should reflect mathematical formalism to execute actual simulations. Mathematical formalism enables dynamic execution of the combat elements for combat M&S based on various tools, which support the formal analysis and verification of combat models (Seo Y et al., 2006). To support the validation and formal analysis of PBM, we propose the following steps to execute the simulation by applying Petri net, which is a typical classical formalism. The first step is to analyze the characteristics of the conceptual model. PBM has various combat processes and characteristics. Although these processes and characteristics can be depicted with the combat situation, they cannot be simulated dynamically or analyzed formally. For instance, the attack function of an aircraft depends on the sequence of CFEs and the components of the aircraft, which can be used with completely different CFEs than the attack function of a tank. Therefore, the main concern of PBM is how to describe the task concurrency because the weapon needs to concurrently cope with different kinds of targets and threats such as aircrafts, missiles, and tanks. The second step is to select a formalism that is adequate for PBM. To study the characteristics of PBM and the combat process, we investigate various classical formalisms such as BPMN, DEVS, and Petri net (Kim TG et al., 2013; Li et al., 2013; Luo et al., 2016). Of them, Petri net is a suitable choice for analyzing the concurrency, boundedness, and reachability of multiple tasks. It can be used to effectively validate various state changes of combat processes and complex interrelations between combat models. The third step is to transform the process-based model to the Petri-net model. If Petri net is chosen, then PBM needs to be transformed to the process-based Petri-net model. The following three steps are used for the transformation: (1) Load the process-based model generated by the PBM system. Information on PBM is saved in a table form in a MySQL database, and is set up to interface with other programs. (2) Make rules for converting PBM information into an XML form, which Petri-net simulators can read (Li et al., 2013). The Petri-net simulators carry out the semantics of formalism through the interrelation between PBM and Petri net in Table 2. (3) Transform PBM into the XML-format model using the mapping relationship in Table 2. PBM is saved first as XML, then the models are transformed as XML, and the XML models are finally imported into the Petri-net tool.

Table 2 Mapping relations between the process-based model and the Petri-net model

Process-based model	Petri-net model
Common functional element (initial element of each encapsulated model)	Place
Common functional element (final element of each encapsulated model)	Place
Phase	Place
Condition (partly)	Transition
PhaseIn	Input
PhaseOut	Output
CombatProcessPhase	Token

5 Case study: an aircraft with a stealth function and a radar with a detection function

To demonstrate the usability and applicability of the proposed PBM system, this section presents the process-based models of an aircraft with a stealth function and a radar with a detection function. How PBM is modeled from combat scenarios is presented first, followed by a study of how formalism can be used to support the formal analysis and the verification of PBMs.

5.1 Process-based modeling of the combat scenario

We design PBMs that describe an aircraft with a stealth function and a radar with a detection function using the four combat scenarios from field manuals (USDA, 2007; Jeong D et al., 2016; Oh et al., 2016). An aircraft's stealth function allows missions to be performed without being detected by enemy, and the radar detection function provides enemy information to other weapons after detecting the enemy. To analyze these two weapons' functions, we generate two ICPs by combining the following four combat scenarios, according to Section 3.3:

The first combat scenario: destroy the enemy's tank using the stealth aircraft's reconnaissance module.

The second combat scenario: detect the enemy's aircraft using the radar, and then attack the enemy's tank with a missile.

The third combat scenario: attack the enemy using the stealth aircraft's attack module.

The fourth combat scenario: detect the enemy's aircraft using a radar to detect stealth, and attack it with the tank.

The two ICPs are generated by mutually combining the first and second combat scenarios and the third and fourth combat scenarios. The first ICP has friendly weapons (two aircraft and a command post) and enemy weapons (a radar, a launcher, a missile, and a tank), and the second ICP has a different set of friendly weapons (a radar, a command post, and two aircraft) and enemy weapons (a radar and a tank). To design the combat models from ICPs, we model PBMs by deploying EMs according to ICPs through Algorithm 1. The first PBM is shown in Fig. 8, and the second one is shown in Fig. 11. The process-based conceptual models are fundamentally of the same form, except that the combat scenarios are specific according to the tactical goal. The first PBM has the EM sequences for achieving both the friendly and enemy purposes. It is designed to analyze how effectively an aircraft with the stealth function performs its missions when carrying out the detection function. The other PBM also has the EM sequences for achieving the purposes of the first combat scenario (Radar→Command-post→Aircraft (Attack)→Aircraft (Move)) and the second scenario (Aircraft→Radar→Tank). It is designed to analyze the stealth function by deploying an aircraft with the stealth function and a radar to detect it. We add information on the resource model and the FM required for EM generation, and build the PBM data using the C#-based PBM system presented in Section 4. Then we run the simulation using that system and generate the XML-based Petri-net model to analyze the process-based models.

5.2 Petri net based analysis of the process-based model

Following the transformation steps specified in Section 4.3, we transform the process-based conceptual model (Fig. 8) into the process-based Petri-net model in Fig. 12. This process-based Petri-net model contains 22 places replacing all the CFEs of EM_0–EM_6 (Table 3). We set the initial token number of places 1, 5, 9, 12, 15, 18, and 21 (initial elements of EM_0–EM_6) to 1 to start the state transition. The tokens can move to the next phase if two tokens are located

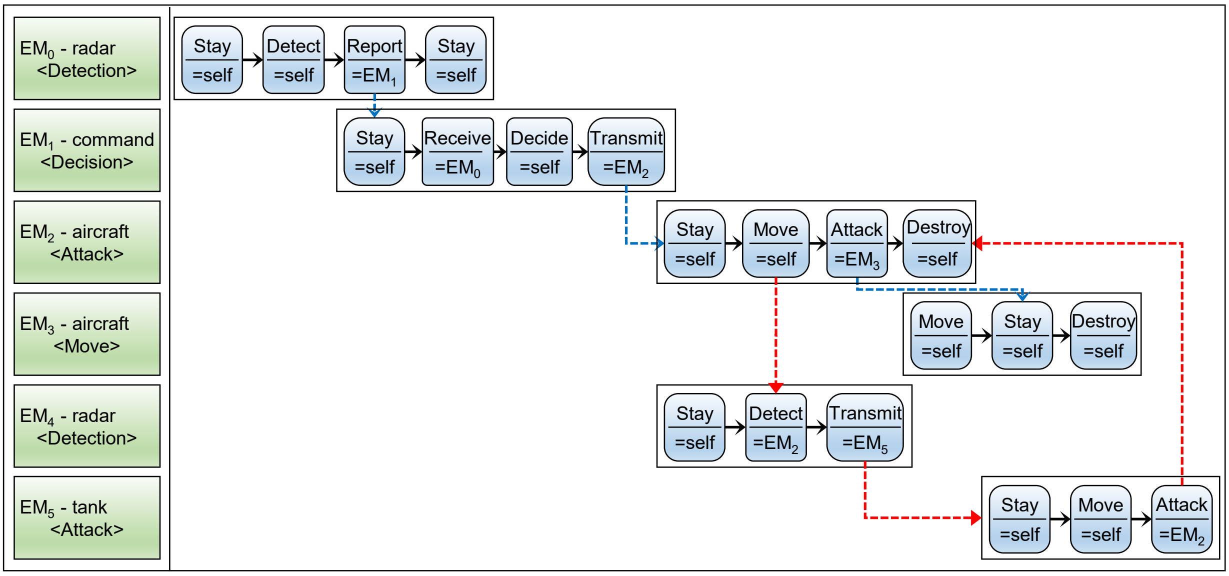

Fig. 11 Process-based modeling showing the radar's function of dectecting the aircraft with the stealth function

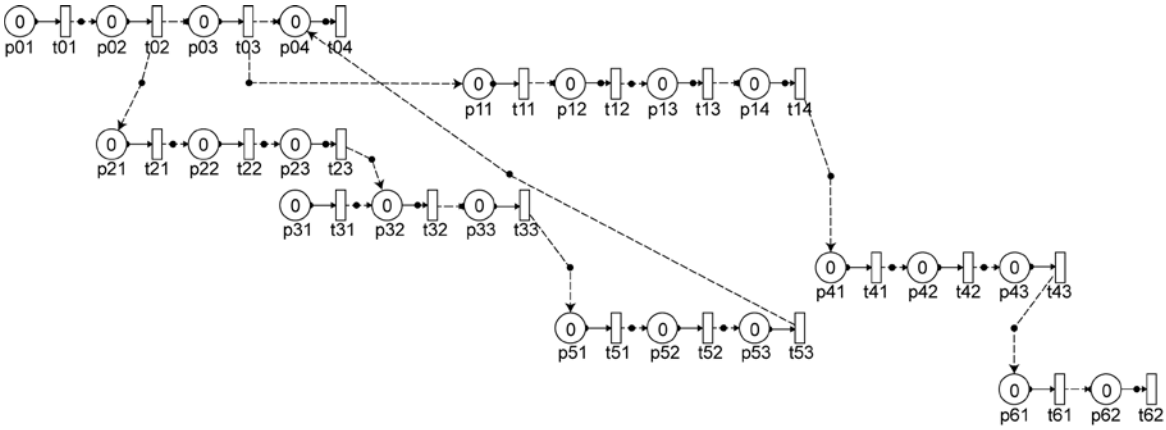

Fig. 12 A process-based Petri-net model showing aircraft with a stealth function

Table 3 An analysis of the process-based Petri-net model

Model	Action	Place	Number of tokens that moved through the 12 markings											
			01	02	03	04	05	06	07	08	09	10	11	12
EM_0 (aircraft)	Move	01	1	0	0	0	0	0	0	0	0	0	0	0
	Detect	02	0	1	0	0	0	0	0	0	0	0	0	0
	Report	03	0	0	1	0	0	0	0	0	0	0	0	0
	Stay	04	0	0	0	1	1	1	1	1	1	1	2	0
EM_1 (command post)	Stay	05	0	0	0	1	0	0	0	0	0	0	0	0
	Receive	06	0	0	0	0	1	0	0	0	0	0	0	0
	Decide	07	0	0	0	0	0	1	0	0	0	0	0	0
	Transmit	08	0	0	0	0	0	0	1	0	0	0	0	0
EM_2 (radar)	Stay	09	0	0	1	0	0	0	0	0	0	0	0	0
	Detect	10	0	0	0	1	0	0	0	0	0	0	0	0
	Transmit	11	0	0	0	0	1	0	0	0	0	0	0	0
EM_3 (launcher)	Stay	12	0	0	0	0	1	0	0	0	0	0	0	0
	Receive	13	0	0	0	0	0	2	0	0	0	0	0	0
	Launch	14	0	0	0	0	0	0	1	0	0	0	0	0
EM_4 (aircraft)	Stay	15	0	0	0	0	0	0	0	1	0	0	0	0
	Move	16	0	0	0	0	0	0	0	0	1	0	0	0
	Attack	17	0	0	0	0	0	0	0	0	0	1	0	0
EM_5 (missile)	Stay	18	0	0	0	0	0	0	0	1	0	0	0	0
	Move	19	0	0	0	0	0	0	0	0	1	0	0	0
	Attack	20	0	0	0	0	0	0	0	0	0	1	0	0
EM_6 (tank)	Stay	21	0	0	0	0	0	0	0	0	0	0	1	0
	Destroy	22	0	0	0	0	0	0	0	0	0	0	0	1

on an interrelated place, and they wait until the required number of tokens at the place is met. When analyzing the process-based Petri-net model, different methods are available, such as those based on state boundedness, reachability, and liveness. Because the core focus of this study is how to analyze the interactions between weapons of PBM based on the combat processes described in Table 3, in this subsection, we do not study format-based formal analysis methods. Therefore, PBM is analyzed by presenting a series of markings. The 22 places represent the combat actions of the friendly and enemy forces, and the 12 markings represent the flows of the combat processes of the combat situation: the actions of the friendly forces first travel along a set of specified places (EM_0 actions p01–p04, EN_1 actions p05–08, EM_4 actions p15–p17, and EM_6 actions 21–22). The actions of enemy forces travel along with a set of specified places (EM_2 actions p09–p11, EM_3 actions p12–p14, and EM_5 actions p18–p20). The markings reflect the friendly and enemy combat processes. First, for the friendly forces, the reconnaissance function of the initial aircraft is marked 01–08, and the decision function of the command post receiving the reconnaissance information is marked 04–07. Then, the attack function of the aircraft for destroying the enemy's tank is marked 08–10, and the destruction function of the tank is marked 11–12. For the enemy forces, the detection function of the radar is marked 02–04, and the launch function of the launcher for attacking the detected enemies is marked 03–05. Finally, the attack function of the missile is marked 04–07. To analyze the concurrency of the process-based Petri-net model and interactions between weapon entities, we analyze this model in terms of boundedness, concurrency, and conflict, which are core characteristics of Petri net, as follows.

1. Boundedness analysis

The two places (p04 and p32 in Fig. 12) that reflect the interaction between two CFEs can have two tokens, and other places can generally have only one token. This Petri net can either produce new tokens or lessen tokens in the process of interactions between weapons, but it represents changes of only the limited boundary. Therefore, considering the concurrency of this Petri-net model, the boundedness value of this Petri net is 4 (markings 05–06) including the places with interaction.

2. Concurrency analysis

The concurrency of Petri net exists when the number of tokens is larger than two, meaning that there are more than two weapons being processed currently (markings 03–11). Markings 03–11 have processed the weapons corresponding to the friendly and enemy forces simultaneously, and marking 05 has used up to two weapons. Of the friendly weapons, the stealth aircraft of EM_0 has one token of the stay action until receiving a certain action from the enemy (markings 04–10), and then this aircraft is attacked before the destruction of the enemy tank (markings 11–12). Therefore, we need to take new actions and use other weapons to defend the friendly aircraft from the enemy one. Because the EM_2 radar, which detects the stealth aircraft from the enemy weapons, quickly performs the process from the detection of the stealth aircraft to the attack of the missile (EM_2 in markings 04–05, EM_3 in markings 05–07, and EM_5 in markings 08–10), one can conclude that this radar is effective. Therefore, it is crucial to simulate various tactics through concurrent task coordination to reflect the effective tactical orders of complex combat situations.

3. Conflict analysis

Conflicts are necessary for modeling special combat situations, such as concurrent tasks and mutual exclusion. Conflicts usually require special treatment, i.e., user intervention. Place 04 (stay) represents the aircraft's stay action waiting for the next command after the mission. In the actual combat, weapons do not stay in one position for a long time, because each weapon performs its assigned missions for tactical purposes. However, to describe combat scenarios, place 04 in marking 04–10 takes one token because the aircraft should receive an uncertain action, i.e., being attacked by an enemy missile through a stay action. Likewise, the weapons with action "stay" in Figs. 8 and 12 should stay to receive an uncertain action of performing their next mission according to the predetermined combat scenarios. Place 13 (receive) represents that the launcher has an action of launching a missile after receiving the information from the radar. Marking 06 has two tokens because "transit" and "receive" have a 1:1 relationship according to rule interaction analysis between the CFEs in Section 3.3.2. If the conditions are met, the number of following tokens decreases to one. According to the execution semantics of Petri net, the transition for places 04 and 13 cannot fire unless two tokens are prepared.

Overall, although the Petri net model cannot analyze all aspects of PBM, it can depict the combat scenario exactly and formally in an exact and formal way. The Petri-net model can observe the task concurrency and conflict, and support a formal analysis. Based on this analysis information, we can effectively review the functions inherent in the weapons at the conceptual design stage.

5.3 Comparative analysis and interoperation analysis

5.3.1 Comparative analysis with related studies

We compare the aforementioned research and analysis with the operational test and evaluation system proposed by Lee B and Seo (2014) and with the modeling method of integrating the ACS process proposed by Luo et al. (2016), as discussed in Section 2. We find the following significant differences satisfying the four requirements proposed in Section 3:

1. The operational test and evaluation system proposes the performance evaluation model through constituents of weapon systems. Our process-based model not only systematically models the unique function of a weapon to classify and reconstruct physical and behavioral elements through the CBD and FBS technology, but also establishes interrelations between weapons based on the combat scenarios.

2. The operational test and evaluation system focuses mainly on visual simulation, whereas our PBM system focuses on the analytic model (i.e., the process-based model), which enables formal analysis by analyzing the interrelations between weapons that are inherent in dynamic features.

3. The modeling method of integrating the ACS process concentrates on conceptual designs that integrate the operation systems used in ACS. Our PBM method defines interaction rules and the algorithm in detail to integrate combat processes, and presents a method to review how various weapons are functioning when using Petri net in the conceptual design stage.

4. The modeling method of integrating the ACS process mostly designs a system that is limited to aircraft and the command post. Our PBM method designs the PBM system to generate the combat scenario model using various combat weapons. The PBM system shows that it can review how properly the weapons and the weapon functions are used in combat through the process-based Petri-net model.

These differences actively solve the problems discussed in Section 1 and can help deploy the weapon systems required in wide-range combat situations by reviewing the weapons and their functions.

5.3.2 Interoperation analysis through the conceptual interoperability model

To analyze transparency and interoperation to ensure the conceptual alignment of the process-based model, we compare the interoperability of the process-based model and the models in related papers. We use the levels of the conceptual interoperability model (LCIM) to evaluate how well different interoperability approaches satisfy the various levels of the developed model (Wang et al., 2009; Tolk, 2012). The results are shown in Table 4.

Lee B and Seo (2014) showed the exchange of data between systems by conceptually modeling the interrelationships between weapons. However, systematic expression of the compositional information of the many weapons used in combat is limited, and the approach fails to meaningfully express the interrelationships between weapons. Luo et al. (2016) provided a process integration M&S framework for an air command scenario. Therefore, the approach supports the exchange of contexts between processes, which can be analyzed as semantic syntax. Further research using more detailed methodologies is needed because these methodologies are conceptual models for the reconstruction of processes and system structures resulting from state changes.

In contrast, PBM presents a standardized method to identify information, processes, and contexts

Table 4 Interoperability analysis of the process-based model proposed in this paper and the models proposed in related papers

Level	Interoperability at this level	Lee B and Seo (2014)	Luo et al. (2016)	This paper
L6 (conceptual)	Grasp processes, contexts, information, and modeling assumptions for a conceptual model	X	X	Δ
L5 (dynamic)	Realign information protection and consumption to understand the context and meaning according to the state changes of the context and meaning	X	Δ	O
L4 (pragmatic)	Be aware of the context and meaning of the information being exchanged	Δ	O	O
L3 (semantic)	Understand and exchange semantic phrases	Δ	O	O
L2 (syntactic)	Exchange an agreed protocol and the right forms of data	O	O	O
L1 (technical)	Detect technical connection and data change between systems	Δ	O	O

X: least interoperability; Δ: medium interoperability; O: best interoperability through the segmentalized modeling steps presented in Section 3 and provides a conceptual model that logically expresses the interactions between the weapons and their functions to form processes dynamically. These differences reflect the fact that PBMs present a new way of encompassing information from the technical level to the conceptual level in combat scenario modeling, and that data (i.e., entity and appropriate concepts), data usage methods (i.e., process concepts and methods), and constraints can be described as system engineering artifacts. Therefore, using the arguments proposed by Wang et al. (2009), the PBM method allows transparency to ensure conceptual alignment while protecting the intellectual property of the detailed implementation. This enhances the practical usability of the method.

6 Conclusions and future work

This research is to present a PBM method to automatically generate combat scenario models that can verify the functions that new weapons will be able to perform at the conceptual design stage. The PBM method generates the EM comprising physical and behavioral components using the CBD and FBS technology and the combat process of the combat scenario in which interactions between combat entities representing particular missions are considered. Then, the process-based model of the combat scenario is built to verify the combat entities and their functions by matching EMs to missions corresponding to the combat process. To verify the feasibility of the proposed method, we design the PBM system through metamodeling based on UML, and the system provides the process-based Petri-net model for mathematical analysis of the weapons and their functions. The case study illustrates that the proposed method facilitates the automatic transformation from various weapons and combat scenarios to the process-based models and can quickly verify weapon functions and mission completion using the characteristics of Petri net to solve the issues presented in Section 1.

The main contribution of this paper is presenting a process-based model of the combat scenario that considers the interactions between combat entities with behavioral functions. It means that the variant forms of the combat scenario model can be generated easily through a combination of process modules. The PBM method additionally provides the following contributions: (1) reconfigurable combat scenario generation; (2) visual representation of combat entities and combat scenarios used in the engagement level; (3) integrated design of methods used to model combat entities and combat scenarios; (4) transparency of conceptual alignment while protecting the intellectual property contained in the combat scenario model information. These contributions address the complexity of weapon development in advance and can reduce the cost and time associated with the trial and error of development by reviewing the weapon functions through the process-based model. Application of this study will allow military strategists and researchers to perform various combat scenarios in simulation and to evaluate the effectiveness of new weapons without using physical assets. In addition, the PBM method can be applied to process-based M&S, and we can use the PBM method in M&S research of shipbuilding production systems (Jeong DS and Seo, 2018).

Because this work is a conceptual study of a new approach to combat scenario modeling, there is a lack of studies on the normalization of combat scenarios, standardization of the data used, and a complex system framework for researching integrated modeling systems. In future work, we plan to conduct the following studies to fill the aforementioned deficiencies: automated combat scenario model generation research based on MSDL, combat simulation and measures of effectiveness (MOE) research using DEVS, and PBM system framework research. First, to provide the initialization of the combat process, we plan to study formal descriptions that can provide normalization and standardization of combat scenario models by adding MSDL to the existing ontology-based scenario generation. Although weapons' functions have not been analyzed in detail because of the limited data available due to national security concerns, we plan to study formalism and simulation for transparency of the process-based models using the actual data used in national defense and DEVS to replace the existing Petri net. This will allow the MOE of combat entities to be studied. Finally, we will study the framework for implementing a complex system by integrating the systems of each modeling step.

Contributors

Dongsu JEONG designed the research. Dongsu JEONG and Dohyun KIM processed the data. Dongsu JEONG drafted the manuscript. Yoonho SEO helped organize the manuscript. Dongsu JEONG, Dohyun KIM, and Yonnho SEO revised and finalized the paper.

Compliance with ethics guidelines

Dongsu JEONG, Dohyun KIM, and Yoonho SEO declare that they have no conflict of interest.

References

Alshayeb M, Khashan N, Mahmood S, 2016. A framework for an integrated unified modeling language. *Front Inform Technol Electron Eng*, 17(2):143-159. https://doi.org/10.1631/FITEE.1500094

Choi BK, Kang D, 2013. Modeling and Simulation of Discrete-Event Systems. John Wiley & Sons, Inc., Hoboken, USA.

Choi J, Moon S, Kim T, et al., 2015. Study of development for distributed battlefield simulation environment: one-to-one single unit engagement model. *J Korea Soc Simul*, 24(4):69-76. https://doi.org/10.9709/jkss.2015.24.4.069

Claxton JD, Cavoli C, Johnson C, 2005. Test and Evaluation Management Guide (5th Ed.). Defense Acquisition University Press, Fort Belvoir, USA.

Defense Acquisition Program Administration (DAPA), 2012. Guide Book of Weapon System Test and Evaluation. Defense Acquisition Program Administration, Seoul, Korea.

Eisner H, Marciniak J, McMillan R, 1991. Computer-aided system of systems (S2) engineering. Proc IEEE Int Conf on Systems, Man, and Cybernetics, p.531-537. https://doi.org/10.1109/icsmc.1991.169739

Jafer S, Durak U, 2017. Tackling the complexity of simulation scenario development in aviation. Proc Symp on Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems, p.1-10. https://doi.org/10.22360/springsim.2017.msciaas.005

Jeong D, Kim D, Seo Y, 2016. Research of ontology based battlefield scenario generation systems. Proc 2nd Int Conf on Artificial Intelligence and Industrial Engineering, p.393-396. https://doi.org/10.2991/aiie-16.2016.90

Jeong DS, Seo YH, 2018. Process-based ship production system modeling for supporting decision processes. Proc 32nd Annual European Simulation and Modelling Conf, p.82-87.

Jeong DY, Kim GI, Cho SS, et al., 2014. The Latest Weapon Systematology. CHEONG MOON GAK Publishing Company, Seoul, Korea.

Kewley R, Cook J, Goerger N, et al., 2008. Federated simulations for systems of systems integration. Winter Simulation Conf, p.1121-1129. https://doi.org/10.1109/wsc.2008.4736181

Kim CO, Cho YH, Yoon JU, et al., 2005. Ontology based negotiation case search system for the resolution of exceptions in collaborative production planning. In: Meersman R, Tari Z, Herrero P (Eds.), On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops. Springer Berlin Heidelberg, p.24-25. https://doi.org/10.1007/11575863_12

Kim D, Seo Y, Sheen DM, 2015. Dynamic component reconfiguration system using case-based reasoning for weapons system in DM&S: guided weapon case. Proc IEEE European Modelling Symp, p.9-13. https://doi.org/10.1109/ems.2015.12

Kim TG, Moon IC, 2012. Combat modeling using the DEVS formalism. In: Tolk A (Ed.), Engineering Principles of Combat Modeling and Distributed Simulation. Wiley, Hoboken, USA, p.479-510. https://doi.org/10.1002/9781118180310.ch21

Kim TG, Kwon SJ, Kang B, 2013. Modeling and simulation methodology for defense systems based on concept of system of systems. *J Korean Inst Ind Eng*, 39(6): 450-460. https://doi.org/10.7232/JKIIE.2013.39.6.450

Lee B, Seo Y, 2014. A design of operational test & evaluation system for weapon systems thru process-based modeling. *J Korea Soc Simul*, 23(4):211-218. https://doi.org/10.9709/jkss.2014.23.4.211

Lee JO, Lee JJ, Suk JB, et al., 2010. A development of the dynamic reconfigurable components based on software product line: guided weapon system. *J Korea Soc Simul*, 19(4):179-188.

Lee PJ, Lee YU, 2011. A study on weapon systems acquisition for the use of Modeling & Simulation (M&S). *Conv Secur J*, 11(3):11-17.

Li XB, Lei YL, Vangheluwe H, et al., 2013. A multi-paradigm decision modeling framework for combat system effectiveness measurement based on domain-specific modeling. *J Zhejiang Univ-Sci C (Comput & Electron)*, 14(5): 311-331. https://doi.org/10.1631/jzus.C1200374

Li XB, Yang F, Lei YL, et al., 2017. A model framework-based domain-specific composable modeling method for combat system effectiveness simulation. *Soft Syst Model*, 16(4):1201-1222. https://doi.org/10.1007/s10270-015-0513-x

Luo YL, Wu YN, Qin YH, et al., 2016. Modeling method for integration of air command and security process. *Int J Model Simul Sci Comput*, 7(1):1641004. https://doi.org/10.1142/s179396231641004x

No H, Son T, 2005. NCW: trends of developed countries and challenges of ROK. *J Korean Def Iss Anal*, 1046:5-19 (in Korean).

Object Management Group (OMG), 2003. OMG Unified Modeling Language Specification (15th Ed.). Object Management Group, Inc., Framingham, MA, USA.

Oh SR, Jeong DS, Seo YH, 2016. Scenario-based evaluation object selection research for performance evaluation of weapon systems. Proc Korean Operations Research and Management Science Society Joint Conf, p.1239-1244 (in Korean).

Özhan G, Oğuztüzün H, Evrensel P, 2008. Modeling of field artillery tasks with live sequence charts. *J Def Mod Simul Appl Meth Techn*, 5(4):219-252. https://doi.org/10.1177/875647930800500402

Park HR, Seo YH, 2015. AHP-based ontology integration method for weapon function. *J Inform Techn Arch*, 12(1):99-112.

Park SC, Kwon Y, Seong K, et al., 2010. Simulation framework for small scale engagement. *Comput Ind Eng*, 59(3):463-472. https://doi.org/10.1016/j.cie.2010.06.003

Park SK, Lee JY, 2003. A study on the assessment of power improvement effectiveness of corps level C4I system applied to integrated fire operation. *J Mil Oper Res Soc Korea*, 29(1):26.

Qi ZH, Wang ZY, Zhang WH, et al., 2006. Design and implication of attack and defense simulation system for ballistic missile with UML. *J Syst Simul*, 18(3):602-603, 606 (in Chinese). https://doi.org/10.3969/j.issn.1004-731X.2006.03.020

Rainey LB, Tolk A, 2015. Modeling and Simulation Support for System of Systems Engineering Applications. Wiley, Hoboken, USA. https://doi.org/10.1002/9781118501757

Schwenn K, Colombi J, Wu T, et al., 2015. Toward agent-based modeling of the U.S. Department of Defense Acquisition System. *Proc Comput Sci*, 44:383-392. https://doi.org/10.1016/j.procs.2015.03.038

Seo DJ, Seo Y, 2016. A method for build an ontology-based component semantic search system for reconfiguration of weapon system. *J Korea Soc Simul*, 25(1):11-20. https://doi.org/10.9709/jkss.2016.25.1.011

Seo KM, Choi C, Kim TG, et al., 2014. DEVS-based combat modeling for engagement-level simulation. *Simulation*, 90(7):759-781. https://doi.org/10.1177/0037549714532960

Seo KM, Hong W, Kim TG, 2017. Enhancing model composability and reusability for entity-level combat simulation: a conceptual modeling approach. *Simulation*, 93(10):825-840. https://doi.org/10.1177/0037549717699872

Seo Y, Kim T, Kim B, et al., 2006. Representation and performance analysis of manufacturing cell based on generalized stochastic petri net. *Int J Ind Eng Theor Appl Pract*, 13(1):99-107.

Simulation Interoperability Standards Organization (SISO), 2008. Standard for Military Scenario Definition Language (MSDL), SISO-STD-007-2008. Simulation Interoperability Standards Organization, Orlando, USA.

Son MJ, Cho DY, Kim T, et al., 2010. Modeling and simulation of target motion analysis for a submarine using a script-based tactics manager. *Adv Eng Softw*, 41(3):506-516. https://doi.org/10.1016/j.advengsoft.2009.10.009

Tolk A, 2012. Engineering Principles of Combat Modeling and Distributed Simulation. Wiley, Hoboken, USA. https://doi.org/10.1002/9781118180310

Tolk A, Diallo SY, Padilla JJ, et al., 2013. Reference modelling in support of M&S—foundations and applications. *J Simul*, 7(2):69-82. https://doi.org/10.1057/jos.2013.3

United States Department of the Army (USDA), 2007. Attack Reconnaissance Helicopter Operations, FM 3-04.126. Headquarters, Department of the Army, Washington, USA.

U.S. Department of Defense (USDoD), 2011. Revision C, Work Breakdown Structures for Defense Materiel Items, New MIL-STD-881. Department of Defense Standard Practice, Washington, USA.

Wang WG, Tolk A, Wang WP, 2009. The levels of conceptual interoperability model: applying systems engineering principles to M&S. https://arxiv.org/abs/0908.0191

Wittman RLJr, Harrison CT, 2001. OneSAF: a Product Line Approach to Simulation Development. MITRE Corporation, Orlando, FL, USA.

Zhao XY, Zhou X, Mei Y, et al., 2012. Structure and content enhancement to military scenario definition language. Int IEEE Symp on Robotics and Applications, p.379-382. https://doi.org/10.1109/isra.2012.6219203