Simulation Modelling Practice and Theory 96 (2019) 101934

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Layered discrete event system specification for a ship production scheduling model

Bongeun Goo[a], Hyun Chung[a,b,*], Soonheung Han[a]

[a] Department Mechanical Engineering, KAIST, Daejeon, Republic of Korea
[b] Department Naval Architecture and Ocean Engineering, Chungnam National University, Daejeon, Republic of Korea

ARTICLE INFO

Keywords:
Modeling & Simulation (M&S)
Ship production scheduling
Layered discrete event system specification (LDEVS)

ABSTRACT

Shipbuilding is a representative engineering-to-order (ETO) industry that commences product design and manufacturing only after contract signing. It is difficult to accurately predict production schedule at the contract stage, and is this is extremely important because the design tends to be unique and the delivery date should be included in the contract given the paucity of information of the product. Although the intermediate products exhibit high similarity, design modification typically occurs even after the production has begun, and this increases the difficulty of accurate planning. Additionally, the production process incorporates labor-intensive assemblies and joining, and it inherently involves the uncertainties in process time and procedures. Thus, shipbuilding companies attempt to manage production plans via dividing them into hierarchical structures and tend to rely on empirical knowledge and data from production history. Recently, discrete event simulation (DES) is actively searched for the shipyard schedule management, which is successfully applied to mass production industries. However, it is not widely applied due to the inherent characteristics of shipbuilding industry. In order to solve the problem, we propose a layered discrete event system specification modeling method for a ship production scheduling system that provides a layer concept for mixed level of information. The aim involves integrating all production information. Each layer exhibits a level of usable information details. Furthermore, it is designed to facilitate cross-linked information between different layers. Thus, we define a mathematical formalism as an extended form of discrete event system specification and apply it to a production schedule model. The model is simpler and easier to implement because it reflects characteristics of shipbuilding production.

1. Introduction

Shipbuilding is a typical engineering-to-order (ETO) manufacturing system that produces a small number of products with a single design [1]. Given the expansion of global shipping volume and development of offshore oil fields, the shipbuilding industry is required to produce larger and more complex products, and this increases the difficulty of production management. The shipbuilding process undergoes detailed design after an order contract is signed. The design review is based on a ship-owner's requirements. However, the entire production schedule is fixed via the contract and a huge penalty is imposed when the production schedule is delayed. Thus, accurate prediction of cost and schedule is required from the early stages of design to improve the efficiency of ship production. However, existing scheduling is managed based on a supervisor's experience [2]. Thus, they experience difficulties in

* Corresponding author at: Department Mechanical Engineering, KAIST, Daejeon, Republic of Korea.
E-mail address: hchung@cnu.ac.kr (H. Chung).

https://doi.org/10.1016/j.simpat.2019.101934
Received 20 November 2018; Received in revised form 1 May 2019; Accepted 21 May 2019
Available online 22 May 2019

developing an accurate process plan [3]. In order to solve the problems, shipbuilding companies focus on applying production techniques based on digital-based process plan and management [4].

Nevertheless, shipbuilding companies experience difficulties applying simulations due to the characteristics of ship production process. The detailed product information is not predefined at the beginning of the production stage although it is gradually elaborated as the production progresses. Given that a shipyard typically exhibits multiple ships that are constructed with different delivery dates, there are differences in the available product information level based on different degrees of progress. The characteristics lead to difficulties in the planning, scheduling, and simulation of a production system when compared to the other mass production industries. Traditionally, shipyards divide their production schedule into hierarchical structures including master, operation, and detail schedules based on the level of the available information. In the early stage, shipyards primarily create a master schedule, which is the basis for contract signing and production management. Conversely, operation and detail schedules are further determined because more detailed intermediate product information develops during the production period. Thus, it is necessary for the master schedule to inherently exhibit a high level of uncertainty, and thus shipyards tend to exhibit a generous margin in their master schedule to absorb the uncertainty, thereby directly affecting their overall competitiveness in the market. Thus, under construction, products exhibit different level of information, and it is necessary to integrate all information to enable accurate prediction.

In order to confirm the necessity and direction of the study, we analyze past studies related to the topic. Among them, we compare and analyze studies on shipyard production scheduling.

J.G. Shin [5] examined an object modeling technology and simulation based design to propose theoretical methodology for multiple job-shop control. The author suggested a development methodology termed as “evolutionary simulation-based design” and analyzed the molding factory of shipbuilding. The study was in the same direction as the present study in terms of indicating the critical points of existing simulation-based design and also suggesting new simulation-based design. Nevertheless, it focused on organizing abstract methodology, and thus it is difficult to apply the results of the study to real shipbuilding. D.K. Oh [6] proposed e-manufacturing methodology based on product lifecycle management (PLM) and materialized simulation architecture with linkages between the simulation model and database. However, it used Delmia QUEST, which corresponded to the existing general discrete event simulation (DES) tool for a simulation engine. P.L. Lee [7] noted the problems of the simulation model in digital manufacturing and attempted to solve the same via applying it to the panel block factory of shipbuilding. The study was significant since it re-used simulation model and maximized its use. However, it did not examine the accumulation and application of simulation data. In addition to the aforementioned cases, several studies applied DES to shipbuilding production. However, DES is used in fields, such as process and logistics, as opposed to scheduling [8,9].

We also analyzed discrete event specification (DEVS) based modeling methods of war-game studies that focus on multi-resolution modeling (MRM). B. P. Zeigler [10] described the approach of MRM modeling via the modeling of the system entity structure / model base (SES / MB) although it did not extend to a formal analysis. Paul K. Davis [11] discussed integrated hierarchical variable resolution modeling (IHVR), which corresponds to an extended concept of variable resolution modeling (VRM). It focuses on model structure and information as opposed to model definition and uses bottom-up connections to improve the accuracy of calculation. However, MRM corresponds to a structure that emphasizes vertical relations [10–12]. It is not suitable for manufacturing system modeling because it emphasizes on data conversion consistency between resolutions as opposed to interlocking and concurrent utilization of data of each resolution [13].

In present study, we propose a layered discrete event system specification (LDEVS) for a ship production scheduling model. Layers divide components of models such as the ship production system. It supports layered modeling structures and makes it possible to run the simulation under an environment where the level of detail of each part is different and mixed. It can be applied to shipbuilding and to various other situations. Additionally, we construct a ship production-schedule model with the proposed modeling method.

2. Ship production scheduling system

Shipbuilding industry is an engineering-to-order system. Ships are customized products based on customer requirements and require high costs and long production times. The production process consists of various processes such as cutting, assembly, and outfitting. Thus, it is very complicated and difficult to manage [1,14].

Thus, a shipyard manages production schedule with a leveled management structure as previously mentioned. Shipbuilding companies traditionally use a hierarchical or layered production schedule management system as shown in Fig. 1. The master schedule is based on a drydock schedule based on the number of ship orders. Operation schedule is based on the master schedule, and

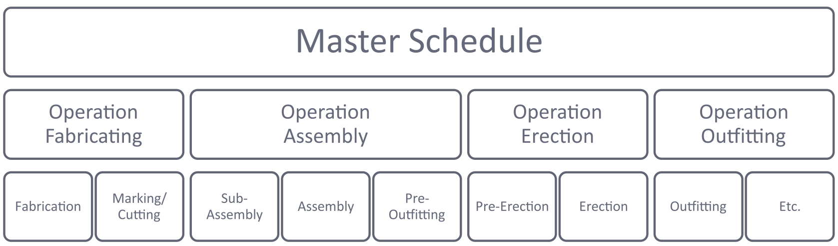

Fig. 1. Scheduling structure of shipbuilding.

it considers workload of each process and/or plant. Detail schedule considers all practical constraints and issues that occur on the factory floor [15].

The delivery date is determined at the contract stage, and a huge penalty is imposed if the delivery schedule is not met. Thus, master schedule determines the approximate production schedule, and it used as a reference for operation/detail schedule. It is necessary for companies to predict all plans as previously discussed, thereby forcing themselves to determine the period of shipbuilding. Subsequently, they implement a detailed process schedule in the limited master schedule. Thus, unnecessary overwork occurs if all the first plans exhibit significant prediction error. In the opposite case, the efficiency of work is reduced. Therefore, shipbuilding requires rough planning and scheduling although the processing information of each part is not predetermined.

The shipbuilding industry exhibits a lower automation level than mass production industries, and labor significantly affects production Therefore, it is difficult to estimate the workload and schedule by considering worker distribution. Given the characteristics of the ETO industry, design and schedule frequently change during production. Furthermore, it is difficult to manage a long-term production plan. Process managers perform a control of each task within a defined schedule. In this case, it is difficult to negotiate and coordinate with other processes, and the possibility of adopting an inefficient decision exists. Hence, the modeling and simulation system for the shipbuilding process can potentially serve as a solution. In order to improve this, it is necessary to perform a flexible schedule change via integrated information management [16].

We also analyzed the requirements of DES to implement this type of a system. Specifically, DES is based on DEVS (Discrete Event Specification) formalism. It is easy to verify and one-to-one correspondence between the actual system and model is evident. Therefore, it is easy to apply to large-scale systems such as shipbuilding production. Thus, several commercial simulation systems are developed via the DEVS formalism [17].

However, commercial DES systems are generally specialized in mass production systems and require an accurate simulation model construction via the definition of all information. While applying this to shipbuilding production, it is not efficient to simulate this to the information level of parts because the process does not require repeated production of the same parts. Given that detailed design is performed after the contract, forecasting of production period and material requirements is required with limited information. Therefore, it is necessary to develop a simulation system that predicts the schedule at the contraction stage while solving the aforementioned issues.

3. Layered discrete event system specification (LDEVS)

In this section, we summarize DEVS and LDEVS status based on the LDEVS's concept and formal specification.

3.1. Discrete event system specification (DEVS)

The model through the DEVS divides a system into small modules (atomic model) and constructs hierarchical models (coupled model) [17,18]. Specifically, the meaning of layer is not the same as the meaning of hierarchical modeling, which is proposed in the present study.

$M = \langle X, Y, S, \delta_{int}, \delta_{ext}, ta, \lambda \rangle$

X:the set of input ports and values

Y:the set of output ports and values

S:the set of states

$\delta_{int}: S \rightarrow S$, *the internal transition function*

$\delta_{ext}: Q \times X \rightarrow S$, *the external transition function*

$Q = \{(s, e) | s \in S, \ 0 \le e \le ta(s)\}$:the total state set

e:the time elapsed since last transition

$\lambda: S \rightarrow Y$, *the output function*

$ta: S \rightarrow R^{+}_{0,\infty}$, the time advance function

An atomic model consists of two event sets (input and output events), a system state variable set, and four functions (i.e., external transition function, internal transition function, output function, and time advance function) [19–22]. The external transition function transits the system state based on its state and external events. Conversely, the internal transition function transits its state based on its state and internal events. The other functions, namely the output function and time advance function, generate next events and determine the life of a state. A coupled model connects the sub-components and sets priorities between them. Fig. 2 describes the structure of a DEVS model.

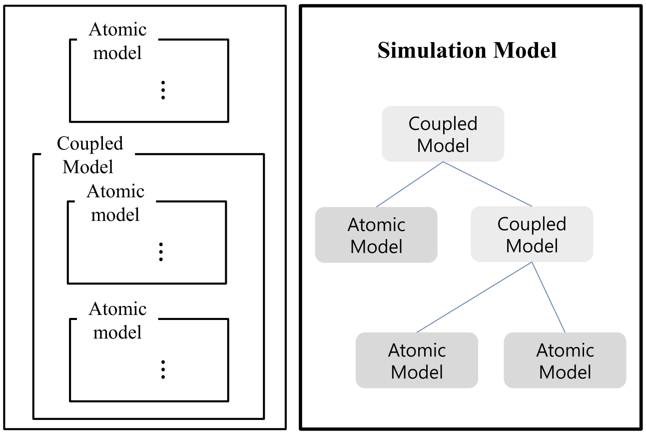

Fig. 2. General structure of DEVS model.

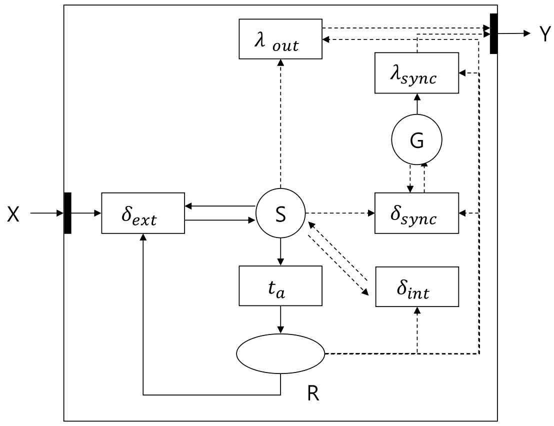

Fig. 3. Atomic model of LDEVS.

$\mathrm{N} = \langle \mathrm{X}, \mathrm{Y}, \mathrm{D}, (M_d | d \in D), EIC, EOC, IC, SELECT \rangle$
X:the set of input ports and values
Y:the set of output ports and values
D:the set of the component names
$\mathrm{M_d}$:the set of all component models in DEVS
EIC: the externel input coupling
EOC: the external output coupling
IC: the internal coupling
$\mathrm{SELECT}:2^D - \{\} \rightarrow D,\ \ the\ tie - braking\ function$

The coupled model constructs hierarchical structures via connecting atomic models and the other coupled models. The characteristic is useful to systematically organize the components in a complex system [23].

3.2. Concept of LDEVS

Specifically, LDEVS is a modeling formalism to simulate production planning and management in an ETO industry such as the shipbuilding industry. Its aim involves improving inefficient modeling structure of existing DEVS-based software and usability. The modeling structure is similar to the production environment of a shipyard to facilitate the ease of application. This is because the production schedule is divided into the master, operation, and detail schedules. This is due to the characteristics of the shipbuilding industry. Shipbuilding industry is a representative ETO industry, and the amount of production per each design typically corresponds to a product. However, significantly high similarity exists between intermediate products. Therefore, there is a tendency to operate processes via general-purpose equipment as opposed to using designated equipment for a target process. Additionally, production does not commence after the completion of a detailed design. Detailed design and production concurrently proceed when contract design is completed. It is difficult to schedule all products at the same abstraction level because detailed design is reflected in production when it exhibits a certain degree of progress. Thus, the management of the detailed schedule proceeds via internal decision-making within the operation schedule and the data management of the detailed schedule is not accurate. Furthermore, it is impossible to manage detailed schedules due to labor-intensive environment. Thus, we focus on the master and operation schedules. In the study, we propose a simulation model that manages an integrated master and operation schedule to solve the limitations of existing DEVS based software. It separates the layers via product breakdown structure (PBS) and work breakdown structure (WBS) and manages the entire information. Actual model configuration and logic definitions are extremely important, and LDEVS is a method to achieve this. Specifically, LDEVS corresponds to an extension theory of DEVS, and it exhibits a model structure to implement the same system divided modeling technique to simulate production planning and management environment utilized in ETO industries such as shipbuilding industry.

3.3. LDEVS formalism

We maintained DEVS formalism and defined additional elements for layer and group management. The sharing of information between layers is defined with general DEVS formalism. However, structural definition becomes difficult and complicated.

$M = \langle X, Y, S, G, L, \delta_{sync}\delta_{int}, \delta_{ext}, \ ta, \ \lambda \rangle$
X:the set of input ports and values
Y:the set of outputports and values
S:the set of states
G:the set of group and resources
$\delta_{sync}: G \times Q \rightarrow G \times S$, *the synchronization transition function*
$\delta_{int}: S \rightarrow S$, *the internal transition function*
$\delta_{ext}: Q \times X \rightarrow S$, *the external transition function*
$Q = \{(s, e)|s \in S, \ 0 \leq e \ \leq ta(s)\}$: the total state set
e:the time elapsed since last transition
$\lambda_{out}: Q \rightarrow Y$, *the output function*
$\lambda_{sync}: Q \times G \rightarrow Y$, *the synchronization output function*
Ta:S $\rightarrow R^{+}_{0,\infty}$, the time advance function

In the case of the Atomic model, G,δ_{sync}, *and* λ_{sync} are added to the DEVS formalism. Specifically, G stores the index information of the models describing the same system and defines the synchronization range and shares information when an input or output occurs. It also defines the layer level and orders between models such as position in the layered structure, information level, and connection flow between the models. Additionally, δ_{sync} corresponds to a synchronizing function for the states in a group. It synchronizes information when an input/output or information change occurs. Furthermore, λ_{sync} corresponds to an output function that is distinguished from λ_{out}, and it synchronizes with δ_{sync} to send a sync signal in a group. If additional elements are not defined, the model behavior is the same as that of DEVS.

The connection information of coupled model corresponds to IC, EIC, and EOC. With respect to the LDEVS, we define Gr and GIC that facilitate interworking of data within a group. Specifically, Gr includes model information of each group, and GIC stores connection information in a group. Fig. 4 shows an example of a coupled model that is used to build a shipbuilding production schedule system.

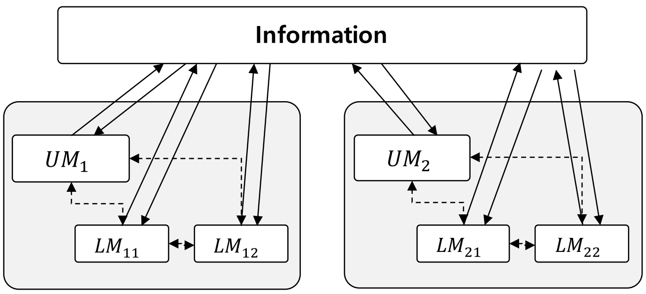

Fig. 4. Coupled model of LDEVS.

$N = \langle X, Y, M, Gr, EIC, EOC, GIC, IC, SELECT \rangle$
X:the set of input ports and values
Y:the set of output ports and values
M_d:the set of all component models in DEVS
Gr: Define groups and set of component models in groups
EIC: the externel input coupling
EOC: the external output coupling
GIC: the internal coupling in groups
IC: the internal coupling
$\mathrm{SELECT}{:}2^D - \{\} \rightarrow D,\ \ the\ tie - braking\ function$

4. Ship production system modeling

Given the characteristics of the shipbuilding industry, such as lack of some information, factories or processes must require different levels of abstraction as mentioned in a previous chapter, The production management system controls schedule through internal decision-making within an up-level schedule, and practical cases of modifying the master schedule do not exist. However, there is a high probability that decisions are not optimal given the difficulty of considering the situations of other processes. In order to solve the problem, a flexible management system is required through integrated information management. Additionally, it is required to develop a simulation that operates even in the absence of detailed information. It is possible to predict the best result by mixing operation schedule information of production and master schedule information of inaccurate production such as schedules, contract, and initial stage product. Thus, we propose a layered (multi-layer) structure. Additionally, the layered structure enables the overall information management in a single system, and it interacts between each process and level. The LDEVS-based ship production model exhibits two main characteristics. The first corresponds to a layered structure, which is divided into upper and lower layers for individual processes. The second corresponds to the operation flow based on events as opposed to products. Based on the LDEVS formalism, we simply define the inter-link structure of the model. The group and layer information of each model is stored, and this facilitates the classification and interlink of information between layers. Additionally, it is possible to adjust the operation range for each element if necessary.

It is impossible to define the information of all simulations in the master schedule stage. Additionally, changes in a few independent variables forces re-simulation of the master schedule due to the structural characteristic of DES. This is confirmed by applying DEVS theory, thereby implying a simple model of simulation, and it aims at a hierarchical model per simulation. In the model case, increases in the size of model make the model more complicated. Furthermore, it increases the time to perform and modify the simulation. Increases in the complexity occurs with increases in the size of model, and this makes it considerably difficult

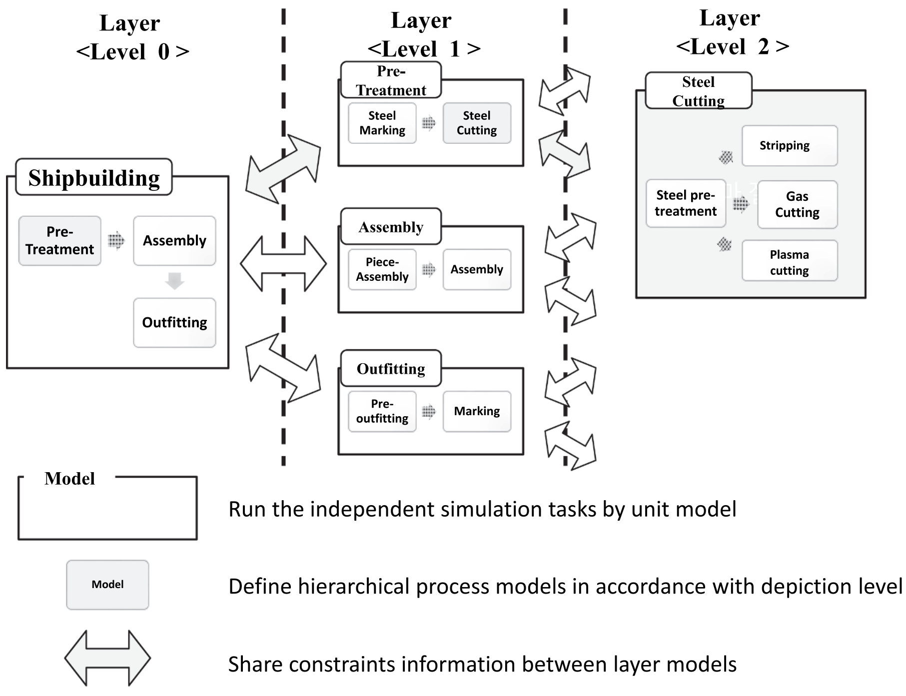

Fig. 5. Concept of the layered model structure for ship production.

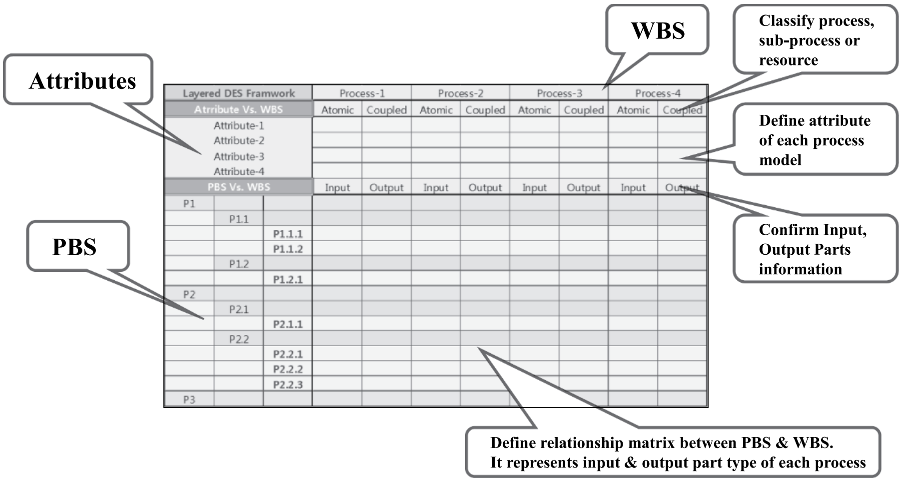

Fig. 6. Matrix of layered DES framework [24].

to compose the complete model. Furthermore, increases in the model size increase the time to perform simulation and the difficulty in determining and fixing problems. Fig. 5 shows the structure of the proposed layered model. The layered model structure exhibits each model independently and dividing the by layer level is different from the single model mentioned above. At the higher level, simply described models involve constructing the system, and the required level of information detail is simple and approximate. However, at the lower level, the system consists of more detailed models to describe the target system. Each model conducts simulation independently, and the connection and smooth accumulation of information correspond to important points in the model. It is difficult to express the structure via general DES software that is currently available.

In order to implement the layered concept by designing the system as coupled models, the employment of several additional coupled models and route models is unavoidable. For example, as shown in Fig. 6, the system is constructed via three levels (i.e., 0, 1, and 2), and the number of coupled models must exceed the number of models in level 1. The modeling approach reduces the difficulty of modeling. In the initial stages, processes are similar to the approach of Critical Path Method (CPM). Additionally, after sufficient information is generated, it describes the detailed model to obtain reliable results. The total model becomes more complex and large due to several additional connections between models. Thus, it is difficult to obtain advantages including high reusability, fast simulation speed, and easier modification. We also devised a matrix framework to explain standardized layered models as shown in Fig. 6.

The framework matrix consists of a product breakdown structure (PBS), work breakdown structure (WBS), and attributes (which define the relation variables between layers and models). The matrix is defined to declare the components in the layered system, and model relationships among the components are defined. Fig. 7 shows an example of conforming how to fill out the matrix. There are four types of processes in the PBS vs. WBS matrix.

Based on the input and output pattern of each process, the process is classified into four types. The first type corresponds to only the output information in state, and the model is executed based on output information. The second type corresponds to when in/out information is the same such as movement, heat treatment, and turnover. The third corresponds to that when an output product is upper level as opposed to an input product although there is only one piece of information in the lower level. For example, a process such as curving surface. The last type corresponds to typical assembly work. The process receives the plurality of sub-product as the input and outputs the assembled product. In types one to three, the input and output correspond to a one-to-one match. Thus, we divide them into two types of common models, namely assembly and non-assembly. Given that the basic model type corresponds to an extended form for multi-layer and group interlink structure, it is expected that existing DEVS based models are applicable.

We simplified model types and applied modeling to simplify model construction. Shipbuilders produce various vessels based on a

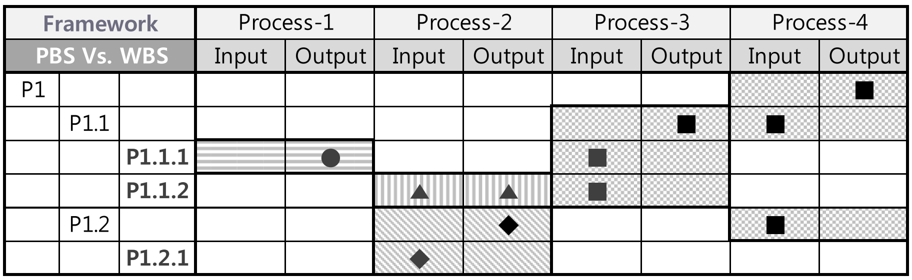

Fig. 7. Example of PBS Vs. WBS relations.

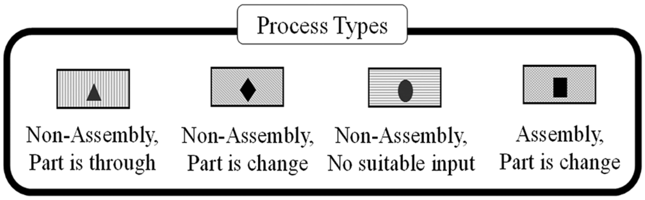

Fig. 8. Patterns of PBS Vs. WBS relations.

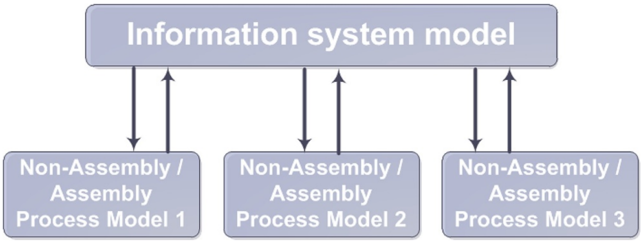

Fig. 9. Operational structure of the model.

common process and shipyard resources. Additionally, the flow of products is not uniform, and their starting positions are different. Thus, if all the connections between models are represented, then unnecessary connections are increased and the complexity of implementation increases sharply. The maintainability also deteriorates. Therefore, the implementation of ship production model is implemented in a simple flow structure. We focused on simplifying the connection logic and not reducing the number of connections. Thus, we designed a centralized method as shown in Fig. 4.

General DEVS-based manufacturing system exhibits product information flow to operate the simulation. It is not easy to collectively generate and manage the approach in the early stage and to control the process flow. When this type of a method is applied, it is not easy to generate and control the process flow. In this case, it is necessary to improve individual models based on the information on the product. Therefore, we propose an event-driven model structure to minimize the variation in the model structure and facilitate the implementation of layered structure.

We define the block information database (DB) that inputs various operation related information with system components and event handling model. When a state transition occurs in the component model, it sends a signal to the event handling model to change data information (such as the position and processing information of the product in DB) and activates the operation of the next process. The product elements include members, plates, and steel, and their combination is irregular. Thus, the flow of products is different. Therefore, it exhibits a complicated flow structure, and thus the configuration and management are difficult.

Signal-type processing facilitates handling complicated connections and each product flow. It is suitable to manage large quantity of product information because it is easy to define a connection structure and to manage each product flow. It only changes variable values in object, and thus it is easy to construct a standardized information system and improve reusability. It significantly reduces complexity for connections such as branching or convergence.

4.1. Elements

Elements of the ship production system model consist of information, non-assembly, and assembly models. The information model connects all the element models. However, it only performs the activation of the next process model without performing calculation. Therefore, it does not include additional elements of LDVES. The connection diagram is shown in Fig. 10.

The state sets of the model consist of three states, namely ‘Idle’, ‘Check’, and ‘Action’. Specifically, ‘Check’ involves checking the next processes, and ‘Action’ sends an activation signal to the next processes. Time flow is absent given that it takes charge of the information flow in the model.

Assembly is an important process in shipbuilding. In the upper layer, the details of the assembly process may not be included based on the level of information. The assembly process performs through a dual queue structure. The formalism definition is as

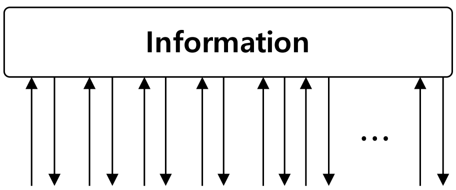

Fig. 10. Information model structure.

follows.

$Assembly = \langle X,\ Y,\ S\ ,\ G,\ \delta_{ext},\ \ \delta_{sync},\ \delta_{int},\ ta,\ \lambda_{out},\ \lambda_{sync} \rangle$

X: {active, *syncin*}

Y: {out, *sync*out}

S: {(n, Sync, Busy_m, Queue1, Queue2_m)|n{0. .maxcapa}, Sync{done, sync},
Busy_m{idle, busy}, m{1, ..,NumCapa}}

G: {name, layer, *order*, *references*}

δ_{int}: $((n \neq 0, -, \text{busy}_m), \text{ta}(\text{busy}_m)) = ((n-1, -, \text{idle}_m)$

δ_{sync}: (G, ((−,sync, −), e)) (G, (−,done, −))

$(G, ((-,-,-), e))$à$(G, (-,sync, -))$: when G values are changed

δ_{ext}: $(((-,-, \text{idle}_m), \text{e}),\ \text{active}) = (\text{busy}_m,\ 0)$: m = 1 or m − 1 = busy

λ_{out}: $((n \neq 0, -, \text{busy}_m), \text{ta}(\text{busy}_m)) = \text{out}$

λ_{sync}: (G, (−,sync, −), 0) = *sync*out

Ta: $ta(busy) = processtime$, $ta(idle) = \infty$

There are four parts of inputs and outputs for signal and synchronization. Signal and synchronization are handled differently. In the case of the signal, it proceeds via changing the model state. Synchronization shares resources when G has an update.

There are two types of queues to perform assembly work. When the process receives assembly parts, it generates assembly block (Parent) information in the primary queue. Furthermore, the process inputs assembly parts to the secondary queue. When secondary queue processes all the required parts, the process updates block information and outputs a signal and synchronization after clearing queues.

The assembly process procedures via the operation of transition functions. In case of synchronization function, it operates to synchronize information in the group when a synchronization signal corresponds to the input or when the reference value of *G* changes. The *G* consists of various information such as the level and order in the group key variables.

In level 1, the most simplified block information is required to perform the simulation. Initial parts are located in the pre-process, and they are transported to the next model. The next model of pre-process can be different based on the layer where the part belongs. The format of the pre-process model is similar to that of the queuing model and it assigns processing time to the parts based on the pre-determined rule. In level 1, the assembly model (which is the next model of the pre-process) cannot implement an assembly process because the input data is excessive approximate to implement the procedure such that the model acts like a buffer model, and this results in a few time delays with respect to the parts in the model. In contrast to level 1, level 2 and level 3 implement assembly procedures that enable small parts to become an assembled block. Fig. 11 shows an example of the assembly model in the level 1 as described in flowcharts.

Fig. 11 simply delineates the assembly model in the level 1. The external transition function takes the input from the simulation engine and inserts the parts into the input into the queue in the model. During the aforementioned process, the occupied area by the parts in the assembly shop is updated. The time advance function calculates the time when the next event occurs and returns the result. Here, the result includes the process time in the process and transportation time between the current process and subsequent process. The output function releases the processed parts to the next model based on the condition of the treated part in the model and the available capacity of the next model. For example, the process releases a part to the next process when both conditions are satisfied, namely 1) the next process is idle and 2) there is a part that is already processed. The last function, namely the internal transition function, updates the queue and the model clock in the model.

Non-assembly process exhibits similar structures with an assembly process although it exhibits a single queue for applying the various types mentioned above. Fig. 4 shows the overall system with all the elements together.

4.2. Interlocking logic

The most important factor in constructing the framework corresponds to the composition of the connection logic between each layer. If the interworking of information between layers at different levels is not working smoothly, the results of the integrated simulation presented are not sufficient. In the case of interlocking logic, it operates separately from various variables and decision logic of each process mode, and performs information-interlocking work between upper and lower layer for the same process model. Therefore, common parameters should be equally applicable to other levels of information. The most important factor in connecting each layer corresponds to a common reference variable. If the process is determined via work capacity or speed, then the volume, weight, or quantity processed can correspond to the basis. Therefore, it is extremely important to determine the reference variables

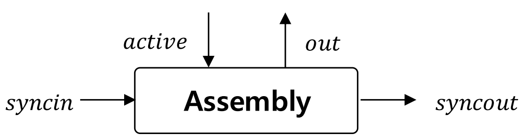

Fig. 11. In/out ports of the assembly model.

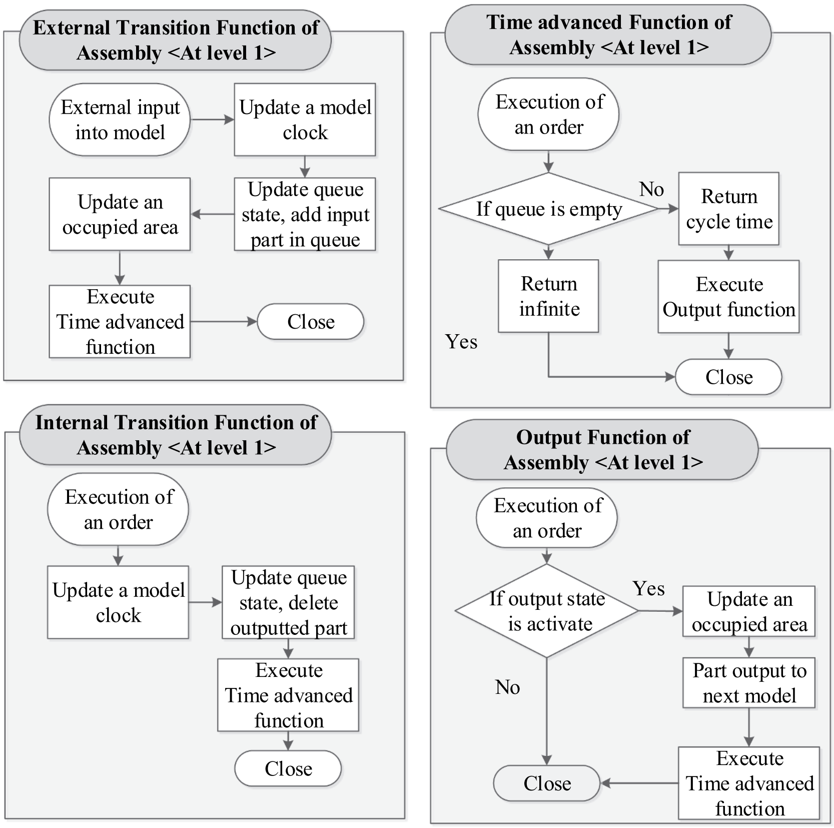

Fig. 12. Specifications of the assembly model in level 1 [25].

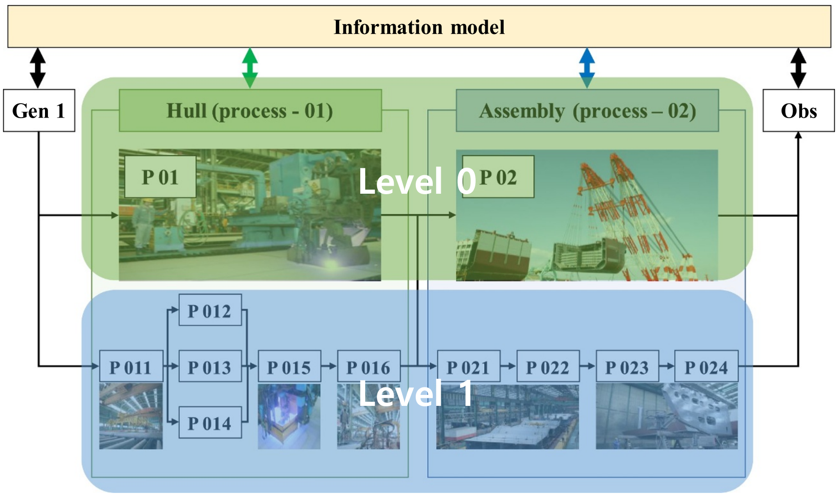

Fig. 13. Layers and groups of the assembly shop model.

and incorporate it into the linkage.

The other corresponds to the occupancy rate per unit period of processes, and this is estimated by reference variables. It is possible to determine a workload and establish an information interworking structure between the layers via load management. If we define a reference variable as main factor, it is expressed in the form of a function such as parameter (Util, Main factor).

Two main factors should be determined through information linkage. The first factor corresponds to the availability of work at each layer, and the second factor corresponds to the duration of work required. In contrast to the flow shop problem, job allocation and production period in the shipbuilding production system are changed based on the workload of process. In a single-layer simulation, it is easy to determine the availability of work at each process. However, with respect to the division into multiple layers, the determination of the availability of work must consider the layer and other layer information.

The level of description is also an issue. A process model of lower layer describes detail resources including internal equipment based on the input information although it is difficult to describe the same level in the upper layer due to the small amount of information. Therefore, it is difficult to consider job allocation of each assembly part due to information limitations. Interlocking logic reflects the problems and supports the operation of simulation.

An additional factor to consider involves changes in production duration with occupancy rate. A few processes are performed irrespective of the occupancy such as the flow shop. However, if the process is performed with limited assignment of workers or resources, then this extends the work duration when the occupancy rate is high. Conversely, when the occupancy rate is low, the process is performed faster than the expected duration. This part is also included in the logic via the formula calculation.

5. Case study

Specifically, LDEVS corresponds to an extended theory of DEVS. It requires additional implementation of components, and the operation structure of the model is also configured in a different way. Therefore, it is appropriate to perform further implementation of necessary functions and components via a library-type engine. We choose Adevs through analysis of strengths and weaknesses among library-type open source engines. In order to confirm feasibility, we plan to construct a simulation system using the actual shipyard data. We reduce the technical and temporal requirements of LDEVS by applying library-type engine.

We reduce development period and ensure the stability of the simulation engine via implementing basic functions using Adevs. We focus on the implementation of layers and group components inside the Adevs model. The scenario model constructed assembly process models and sample data based on actual shipyard data.

The example assembly shop consists of two factories. We create block data based on actual data in the shipyard. At the factory level, we define and manage the weight factor, and this corresponds to an interlocking element between layers. Each factory unit manages interworking between models and resources. The product data consists of five assembled blocks, and this corresponds to processing data from actual shipyard data.

The goal of the LDEVS simulation case study corresponds to a running simulation using all available data at each simulation operation timing to obtain more accurate results. Generally, the optimal result may correspond to configuring a simulation of operation schedule level although a few products including new vessels at the contract stage can exhibit a paucity of information at the operation schedule level. Given the limitations, the simulation for the new vessels utilize master schedule information. In this case, the accuracy of the simulation decreases because it does not include the product information of operation schedule level in production. However, LDEVS model simultaneously simulates all levels of information, and thus it is possible to simulate by using all available information.

Additionally, simulation uses added information by progress. The accuracy of simulation is improved when production progresses. It is possible to solve the problem since the simulation for new vessels is only managed at the master schedule level only. Based on this, the main results of the simulation are as follows.

Figs. 14–16 show the result according to the operation timing of simulation for each schedule level and mixed level. In case of B1 and B2 which were initially input among five assembled blocks(B1–B5), it doesn't show significant difference. Thus, we analyzed the

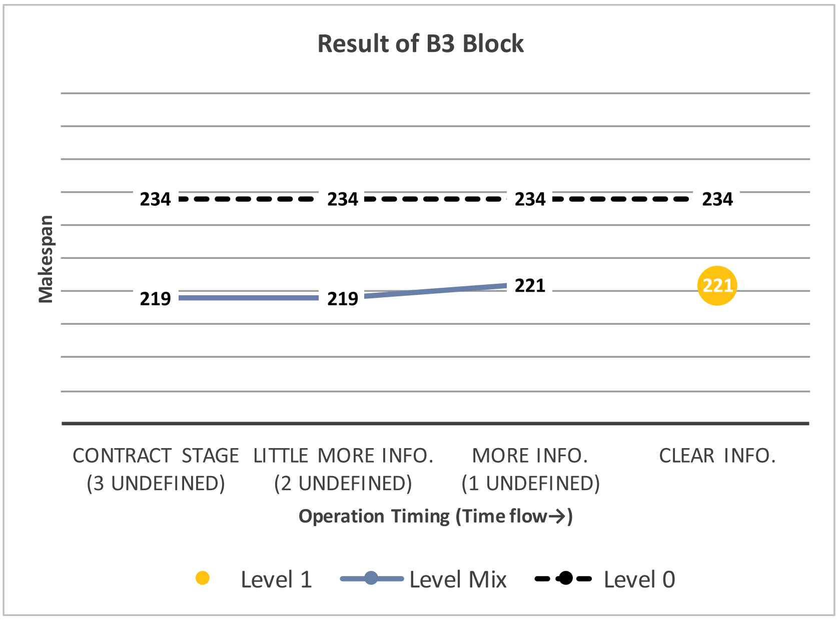

Fig. 14. nitial Comparison of results based on operation timing (B3).

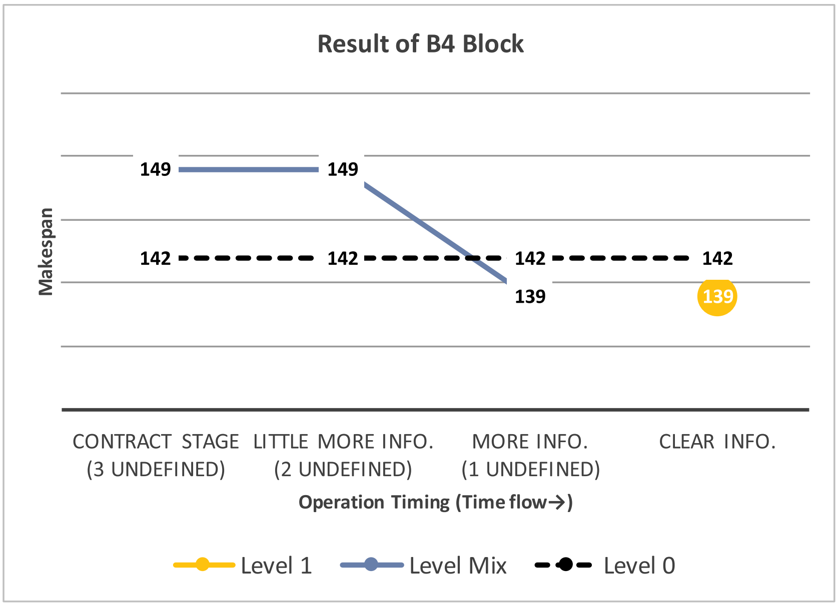

Fig. 15. Comparison of results based on operation timing (B4).

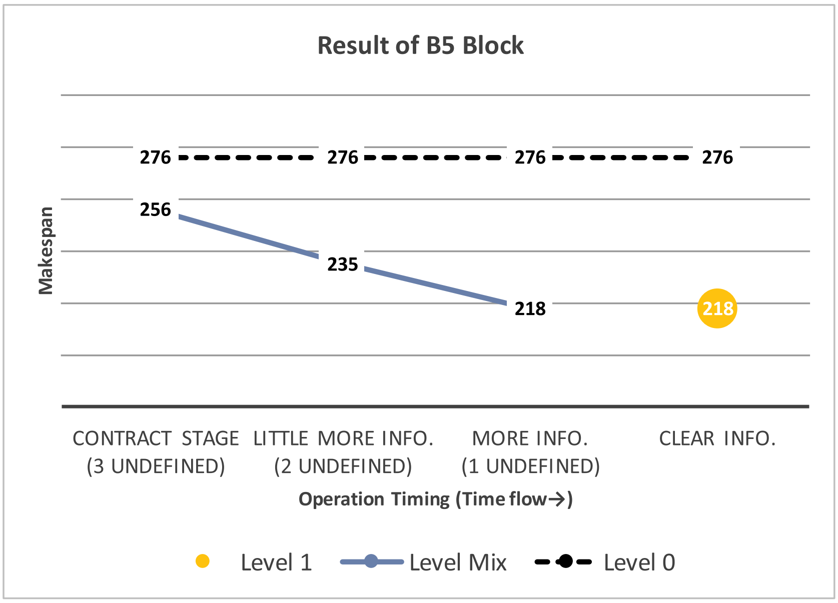

Fig. 16. Comparison of results based on operation timing (B5).

result data of remaining assembled blocks(B3–B5).

We compare the results of mixed level simulations with Level 0 and Level 1. It is observed that the mixed level scenario exhibits a significant effect and it replaces the predicted values of initial level 0 based on the situation. Additionally, a significant advantage corresponds to improving the simulation result based on the increased production information when production progresses. Furthermore, we expect that this is useful since a scheduling manager can share key issues, such as production status and changes, through integrated information management.

6. Conclusion

In the study, we completed the model structure for the simulation of ship production schedule. The proposed layered modeling structure was based on DES for ship production simulation. It was designed to integrate information irrespective of the level of information. We ran a simulation based on DES by using the proposed modeling structure. With the layered structure, a user simulated the production system given the diverse information level of details and followed hierarchical planning and scheduling practices in shipyards. We plan to reinforce the simulation data in future to verify the performance. It should also be noted that a user finds it easy to apply the proposed simulation model into the target system.

Additionally, our future research will focus on pattern-learning research using artificial neural network as a method to complement the connection logic between layers, and a complete system will be implemented in the future.

References

[1] T. Lamb, Ship Design and Construction 1 The Society of Naval Architects and Marine Engineering, Jersey, 2003, p. 883.

[2] K.W. Bang, K.Y. Lee, J.H. Cha, M.I. Roh, H.J. Kim, Proposal of a combined discrete event and discrete time simulation framework and its application to the block erection process in shipbuilding, Proc. Society of CAD/CAM Engineers Conference, 2007, pp. 297–305.

[3] J.H. Cha, M.I. Roh, K.W. Bang, K.Y. Lee, Combined discrete event and discrete time simulation framework for the improvement of shipbuilding process planning, J. Korea Soc. Simul. 17 (4) (2008) 71–80.

[4] K.K. Lee, H. Kang, S.H. Kim, J.Y. Park, J.G. Shin, Digital manufacturing based productivity evaluation according to the change of welding robot torches in subassembly lines of a shipyard, Korean J. Comput. Des. Eng. Vol.10 (3) (2005) 210–216.

[5] J.G. Shin, S.J. Sohn, Simulation-based evaluation of productivity for the design of an automated fabrication workshop in shipbuilding, J. Ship Prod. 16 (1) (2000) 46–59.

[6] D.K. Oh, C.J. Lee, Y.R. Choi, J.G. Shin, J.H. Woo, Simulation modeling methodology and simulation system architecture for shipbuilding process, Korean J. Comput. Des. Eng. 11 (1) (2006) 11–19.

[7] P.L. Lee, D.K. Oh, K.K. Lee, J.G. Shin, The simulation system for scheduling validation of the panel block shop, J. Soc. Nav. Arch. Korea 46 (6) (2009) 641–649.

[8] Y.J. Song, J.H. Woo, D.K. Lee, J.G. Shin, A simulation study for evaluation of alternative plans and making the upper-limit for improvement in productivity of flow-shop with considering a work-wait time, J. Korea Soc. Simul. 17 (2) (2008) 63–74.

[9] H. Chung, T. Lamb, J.G. Shin, A generic shipyard computer model development – a tool for design for production, J. Ship Prod. 16 (3) (2000).

[10] B.P. Zigler, A System Methodology for Structuring Families of Models at Multiple Levels of Resolutions, the Proc. of Conference on Variable-Resolution Modeling, 1992 377-108.

[11] P.K. Davis, Families of models that cross levels of resolution: issues for design, calibration and management, Proceedings of the I993 Winter Simulation Conference, 1993, pp. 1003–1012.

[12] Liu Baohong, A formal description specification for multi-resolution modeling based on DEVS formalism and its applications, J. Def. Model. Simul. 4 (3) (2007) 229–251.

[13] Y.H. Park, An Engineering Context based Multi-level Abstraction Framework for Manufacturing Simulation, Ph.D Thesis Sungkyunkwan University, Republic of Korea, 2014.

[14] R.L. Storch, Ship Production, Second ed., Cornell Maritime Press, 1995, p. 434 pages.

[15] K.K. Lee, et al., Digital manufacturing based modeling and simulation of process in subassembly lines at a shipyard, J. Korea Soc. Simul. (2003) 185–192 2003.

[16] B.G. Goo, H. Chung, C.H. Nan, Layered DES (Discrete Event Simulation) framework for ship production simulation, the Proc. International Conference on Computer Applications in Shipbuilding, 3 2011, pp. 179–186.

[17] K.M. Seo, C.B. Choi, T.G. Kim, J.H. Kim, DEVS-based combat modeling for engagement-level defense simulation, Simulation 90 (7) (2014) 759–781.

[18] J.H. Kim, I.C. Moon, T.G. Kim, New insight into doctrine via simulation interoperation of heterogeneous levels of models in battle experimentation, Simulation 88 (6) (2012) 649–667.

[19] B.P. Zeigler, H.S. Song, T.G. Kim, H. Praehofer, DEVS Framework for modeling, simulation, analysis, and design of hybrid systems, Lect. Notes Comput. Sci. – Hybrid Syst. II (1995) 529–551.

[20] B.P. Zeigler, G. Ball, H. Cho, J. Lee, H. Sarjoughian, Implementation of the DEVS formalism over the HLA/RTI: problems and solutions, Simulation Interoperability Workshop, Orlando, 1999 March. 1999.

[21] B.P. Zeigler, H. Praehofer, T.G. Kim, Theory of Modeling and Simulation, Academic Press, 2000.

[22] T.G. Kim, C.H. Sung, Objective-driven DEVS Modeling Using OPI Matrix for Performance Evaluation of Discrete Event Systems, SCSC, 2007, pp. 305–311.

[23] B.P. Zeigler., Object Oriented Simulation With Modular, Hierarchical Models, Academic Press, 1990, p. 408 pages.

[24] B.G. Goo, A Layered DES Framework for Simulation Based Production of Built-to-order products, M.S. Thesis KAIST, Republic of Korea, 2012.

[25] B.G. Goo, M.J. Choi, H. Chung, A Study on Modeling and Simulation of Layered Ship Production Scheduling System, The Twenty-third International Offshore and Polar Engineering Conference, International Society of Offshore and Polar Engineers, 2013.