Available online at www.sciencedirect.com

ScienceDirect

Publishing Services by Elsevier

International Journal of Naval Architecture and Ocean Engineering 8 (2016) 496−510
http://www.journals.elsevier.com/international-journal-of-naval-architecture-and-ocean-engineering/

A study for production simulation model generation system based on data model at a shipyard

Myung-Gi Back [a], Dong-Kun Lee [b], Jong-Gye Shin [a,c], Jong-Hoon Woo [d,*]

[a] *Dept. of Naval Architecture and Ocean Engineering, Seoul National University, Seoul, South Korea*
[b] *Dept. of Naval Architecture and Ocean Engineering, Mokpo National Maritime University, Mokpo, South Korea*
[c] *Research Institute of Marine Systems Engineering, Seoul National University, Seoul, South Korea*
[d] *Dept. of Naval Architecture and Ocean Systems Engineering, Korea Maritime and Ocean University, Busan, South Korea*

Received 30 June 2015; revised 15 March 2016; accepted 2 May 2016
Available online 22 July 2016

Abstract

Simulation technology is a type of shipbuilding product lifecycle management solution used to support production planning or decision-making. *Normally*, *most shipbuilding processes are consisted* of job shop production, and the modeling and simulation require professional skills and experience on shipbuilding. For these reasons, many shipbuilding companies have difficulties *adapting simulation systems*, *regardless of the necessity for the technology.* In this paper, the data model for shipyard production simulation model generation was defined by analyzing the iterative simulation modeling procedure. The shipyard production simulation data model defined in this study contains the information necessary for the conventional simulation modeling procedure and can serve as a basis for simulation model generation. The efficacy of the developed system was validated by applying it to the simulation model generation of the panel block production line. By implementing the initial simulation model generation process, which was performed in the past with a simulation modeler, the proposed system substantially reduced the modeling time. In addition, by reducing the difficulties posed by different modeler-dependent generation methods, the proposed system makes the standardization of the simulation model quality possible.

Keywords: Modeling and simulation; Digital shipbuilding system; Simulation model generation; Shipyard data model; Panel block assembly shop; Shipbuilding

1. Introduction

Today's shipbuilding industry is facing a serious depreciation as a result of being severely affected by the current global economic recession. Because of low ship prices, small-and-medium shipbuilding companies are experiencing liquidity crises, and even large shipbuilding companies are threatened by the competition from emerging shipbuilding countries. To counteract these situations, many shipbuilders are turning their attention to high value-added lines, such as offshore plants or drill ships. They are increasingly upgrading their operation capacities from the traditional management system with commercial vessel-centered structures and databases to a system that prioritizes the development of new high value-added vessel types. In particular, shipbuilders are placing their effort into establishing a strategically efficient production system that integrates new production technology with the Advanced Planning System (APS) and the Manufacturing Execution System (MES) (Song et al., 2011).

This paper presents a method for the simple and systematic application of modeling and simulation that has been attracting attention as a new production support system. Extensive research has been conducted on manufacturing simulation designed to set up plans with high accurate capable of predicting imminent production-related problems (Woo et al., 2009; Wang et al., 2009). Considering the simulation of the

* Corresponding author.
E-mail address: jonghun_woo@kmou.ac.kr (J.-H. Woo).
Peer review under responsibility of Society of Naval Architects of Korea.

http://dx.doi.org/10.1016/j.ijnaoe.2016.05.005

evaluation of the shipyard panel line capacity as an example, shipbuilding expert analysis of the target factory or line is required, followed by long-term modeling by a specialized simulation modeler. Sophisticated modeling should be performed by a modeler with corresponding background and experience. Because most shipbuilding companies do not have adequate human resources, modeling projects are usually implemented by modeling specialists based on commission. Against this background, this study is intended to develop and present an automated simulation model by redefining the shipbuilding-related data from the simulation perspective, and developing a system based on the data obtained thus far.

For the development of an automated simulation modeling technology, three structural components should be established: a data model with definitions based on the shipbuilding data, simulation control scripts for simulation modeling, and an interface system for simulation software monitoring. Research into simulation automation techniques began in the 2000s with the National Institute of Standards and Technology (NIST) as the leading institute. A variety of practical studies have been reported ever since. Lu et al. (2003) proposed a simulation interface specification for automatic factory modeling for the aircraft manufacturer Boeing. Harward and Harrell (2006) created a neutral file for simulation based on the NIST shop data model and validated its efficacy. More recently, with the development of the neutral simulation schema (NESIS), which integrates related systems by analyzing a range of simulation software architectures and previous studies on simulation data models for the assembly line production industry, a web-based model exchange service was presented (Lee et al., 2011a). A practical interoperability technology that offers integration of heterogeneous software components was implemented using a simulation data model, thus verifying the efficacy of the simulation data model.

Fig. 1 presents the NESIS simulation data model where the data required for simulation are structured in an integrated system that consists of three model elements, namely: a) product, process, and resource, b) configuration for simulation model environment setting, and c) Sim_List structured based on the routing data of the model elements. In this model architecture, the structure for the product, process, and resource is the part that can express the simulation software data in sharable formats, thus posing no difficulties for general factory simulation. The specialized routing expression part is addressed by the Sim_List structure. However, in the shipbuilding industry that constitutes the research object of this study, job shop type processes typically occupy the bulk of the entire operation, and schedule-related data occupy a large proportion of the system. This particularity and the high number of resources that influence the process steps act as limitations for expressing the shipbuilding process simulation with the NESIS architecture. To overcome this difficulty, this paper explores a NESIS-based improved simulation data model that considers the product, process, layout, facility, labor, and scheduled data, specific to the shipbuilding production system. Subsequently, a case study is conducted in a field situation to validate the data model and test the application method.

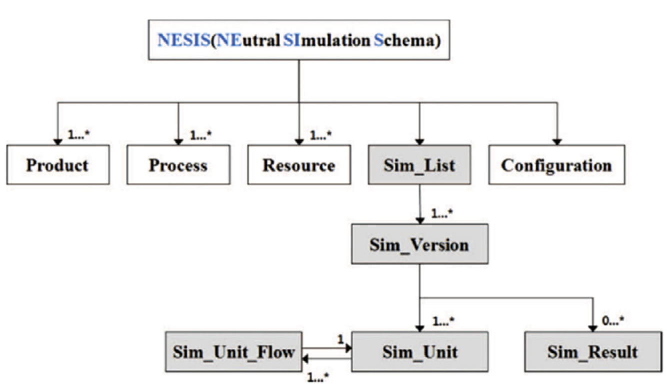

Fig. 1. Simulation data model schema for NESIS (Lee et al., 2011b).

2. Background of automation technology for shipbuilding simulation modeling

In most cases, the simulation for product development or design is embedded in the corresponding Computer Aided Design (CAD) tool as a module. Correspondingly, the simulation does not require an additional module for data conversion, mapping, or system control, unless a separate simulation tool is employed. However, in the case of production simulation, such simulation should run based on data other than product information, such as information on equipment, resources, and production and project schedule (Watson et al., 1997). To address this limitation through the application of the simulation technology to a shipyard field situation, Song et al. (2009) established a simulation model that supports detection in advance and solution of the problems likely to occur in a block assembly factory. However, problems arose in the process of field application of the relevant technology by the field manager after learning the necessary technique, merely because the application of such technique greatly depends on the skills of the simulation modeling engineer. This experience made it clear that a simple “foolproof” application method should be developed so that even a manager without sufficient knowledge of simulation technology can create and manage simulation models.

Woo (2005) defined the simulation model generation procedure in three major steps, as shown in Fig. 2, while conducting a study on the simulation methodology for the prediction of shipyard productivity. In step 1, the problem to be identified or solved is formulated, the system generation project plan is set up, and the system goal is defined. In step 2, input data necessary for the simulation model generation is collected, the collected input data are analyzed to allow the definition of the data for the simulation, and the simulation model is specified based on the analyzed data. In step 3, the simulation model is implemented for productivity prediction, followed by validation and verification. Information pertaining to the simulation goal defined in step 1 is extracted by applying the simulation model constructed. A simulation model is generated by following this series of processes and sub processes. In consideration of the time requirements, although the time required in step 1 for formulating the problem and defining the goal of the simulation model generation is inevitable, the time

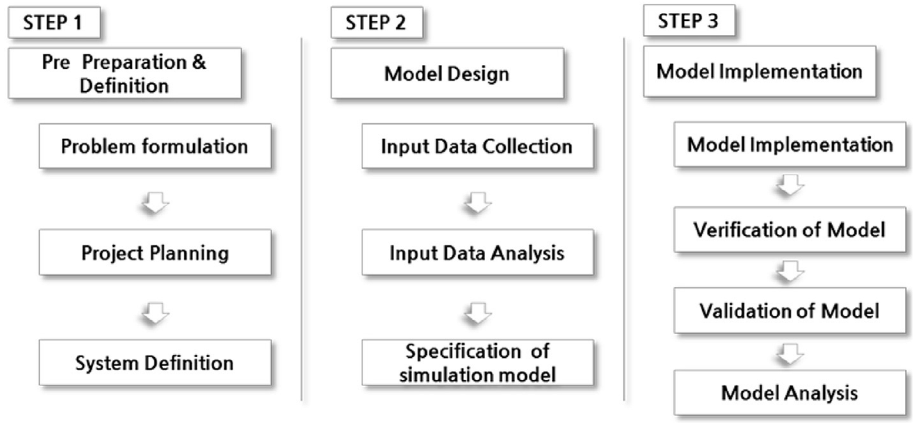

Fig. 2. Model building procedure for shipyard production simulation.

required for steps 2 and 3 for collecting the input data and generating the simulation model based on the collected and analyzed data can be reduced if a system that separates and supports the iterative processes can be employed. This can be achieved by reducing the procedures for generating and managing the simulation model and generating a standardized model.

In order to apply the simulation technology to the production system, data necessary for achieving the simulation goal, such as ERP, APS, and MES, are collected from the shipyard legacy system, and the data necessary for the simulation are analyzed. Based on the analyzed data, the simulation modeler generates the simulation model. Because the formats and types of the collected data are different from project to project, classifying and analyzing the collected data is a time-consuming process. This increases the time and cost for applying the simulation to the shipyard field situation, and thus acts as a hindrance factor for the onsite application of simulation in the shipbuilding industry. In an effort to reduce such difficulties, this study is conducted with the intent of supporting shipbuilding simulation model generation and applications. To this end, we analyze the data generated in a shipyard production environment, and define the data model necessary for shipbuilding simulations. Based on the results of data analysis and the data model definition, we analyze the simulation engine model generation process required for the production simulation model generation, and define the interface between the data model and engine, as a basis for performing the study on the control and communication module. The proposed method is expected to efficiently generate and support shipbuilding simulation models.

3. Data model of shipbuilding for the simulation model generation system

3.1. Data analysis of shipbuilding for the generation system of the simulation model

The definition of the shipbuilding simulation data model should be preceded by the analysis of physical elements and production data managed in the shipyard specific to the shipbuilding industry. First, we analyze the manufacturing data structure, which is the data model managed internally and used by the engine capable of implementing the production simulation. Then, we define the shipbuilding simulation data model based on the review of previous studies conducted on the simulation data model implemented in the shipbuilding industry or in other industrial sectors.

In the analysis of the manufacturing data structure managed internally in a shipbuilding procedure, the production-related particularity of the shipbuilding industry should be considered. This is a typical project-based industry that produces on orders, i.e., its production basis is manufacturing to order, not the mass production of identical products. Therefore, the shipyard production system has data structures that vary from shipyard to shipyard. We analyze the data flow that is common to shipyard data structures from the simulation perspective and present it in Fig. 3. This figure shows the results of the shipyard production flow analysis where the production procedure is regrouped into Product, Process, Resource, and Schedule (PPRS) views, and the input timing and attributes of key information, such as the engineering bill of materials (E-BOM), manufacturing bill of materials (M-BOM), basic procedural data, facility operation data, and work order, can be attained. Although the production flow for mass production uses a unit of productivity, such as the Unit Per Hour (UPH), as an important index, order-based production focuses on schedule compliance of individual products according to the ordered vessel or its blocks (Lee et al., 2014; Woo et al., 2005). Because of this particularity of shipyard production, the shipbuilding industry has data structures that generate an individual schedule timeline for each product.

Among the studies on shipbuilding simulation data models, a study investigated major shipbuilding procedures by creating a digital shipyard, and presented PPRS structure-based flow simulation methods and cases (Woo, 2005). Other studies established a system design for supporting production simulation at the level of implementation planning, and applied the system to the panel line whose standardization is relatively simple among the shipbuilding block production procedures (Back et al., 2013; Hwang et al., 2013). These studies focused

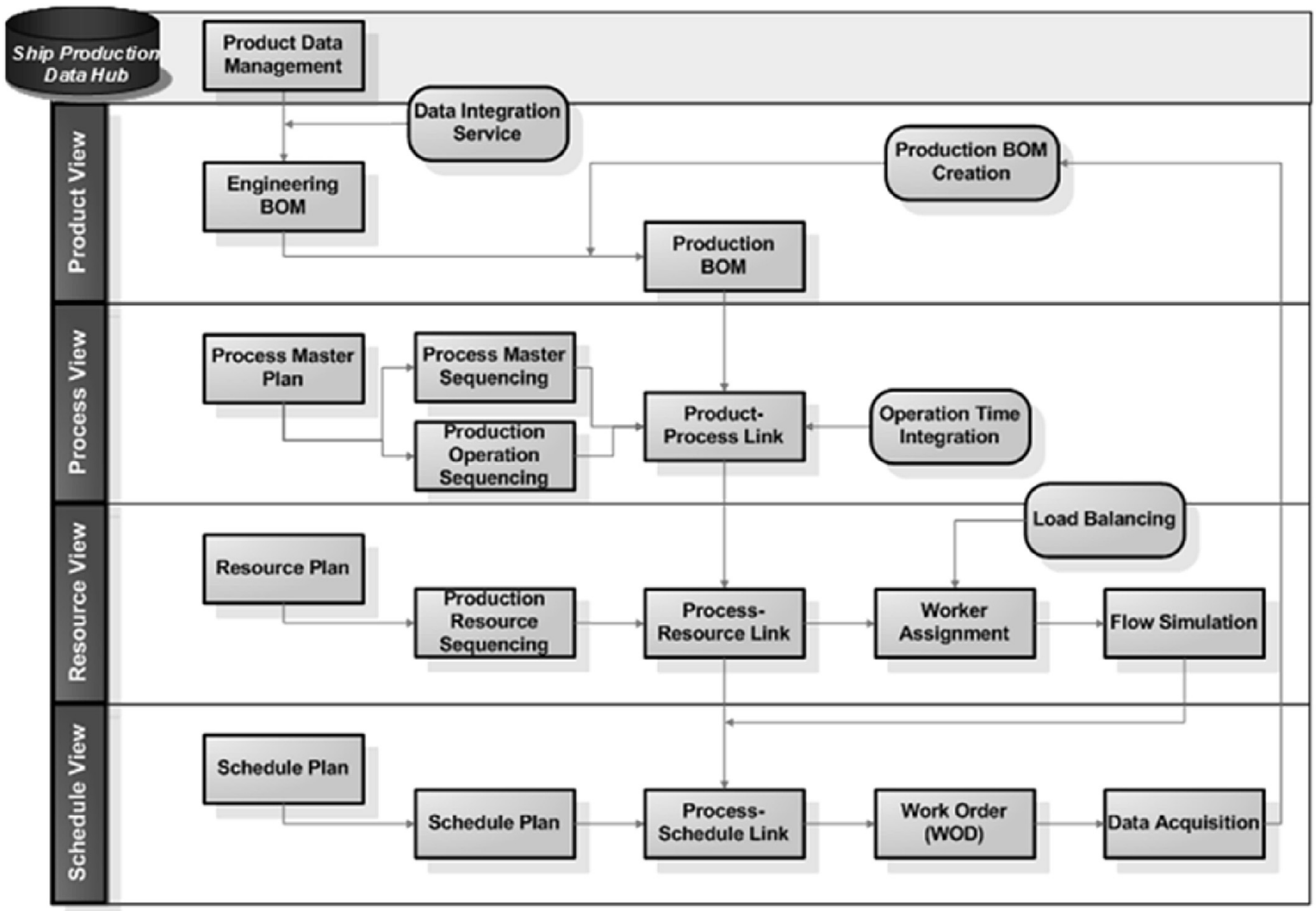

Fig. 3. Production information flow of shipyard.

on verifying production simulation cases and efficacies, thus eliciting the limitations in specifying production simulation data models. Lee et al. (2013) conducted an extensible markup language (XML)-based expandable simulation data model, but showed limitations in reflecting the data that is essential for simulation model generation and management. In the case of other industries, studies on simulation data models have been conducted under the NIST, for example, simulation data exchange (SDX) allows data exchange among heterogeneous simulation engines and data exchange interfaces (Sly and Moorthy, 2001; Johansson et al., 2007; Lee et al., 2011b; Kang, 2007). These studies were intended to allow simulation model exchange, and thus focused on the data on the simulation itself rather than the object of the simulation application. Moreover, their data models revolved around resources suitable for flow production modeling, and thus have limitations in their applications to the process-centered shipbuilding industry.

We defined the data model structure outline for the shipbuilding simulation based on the shipbuilding data model and general simulation models. As illustrated in Fig. 4, the entire data structure is regrouped into four major PPRS groups, and group sub data are specified up to seven step levels. The product view schematically defines the product structure, including product data. By dividing the product structure into engineering and production domains, according to the shipbuilding feature of equal emphasis on the basic technical design and production design, independent archiving and management of the design and production data is ensured. Process view contains the process and production plan data, thus separately reflecting the process characterized by standard information, and the production characterized by actual working information. The schedule view comprises the same structure as the process view in consideration of its relationship with the production plan because it should contain the date assigned to the corresponding working unit. The resource view is segmented into jug necessary for actual work operation, the resource plan that includes the transporter, and the shipyard plan for shipyard space composition.

3.2. Data model and database for the simulation model generation system

In this study, data analysis for simulation modeling was performed from the PPRS views, as shown in Fig. 4, based on the analysis results for E-BOM for shipyard design, M-BOM for production, procedural data for work management, and work assignment process that considers schedules and degrees

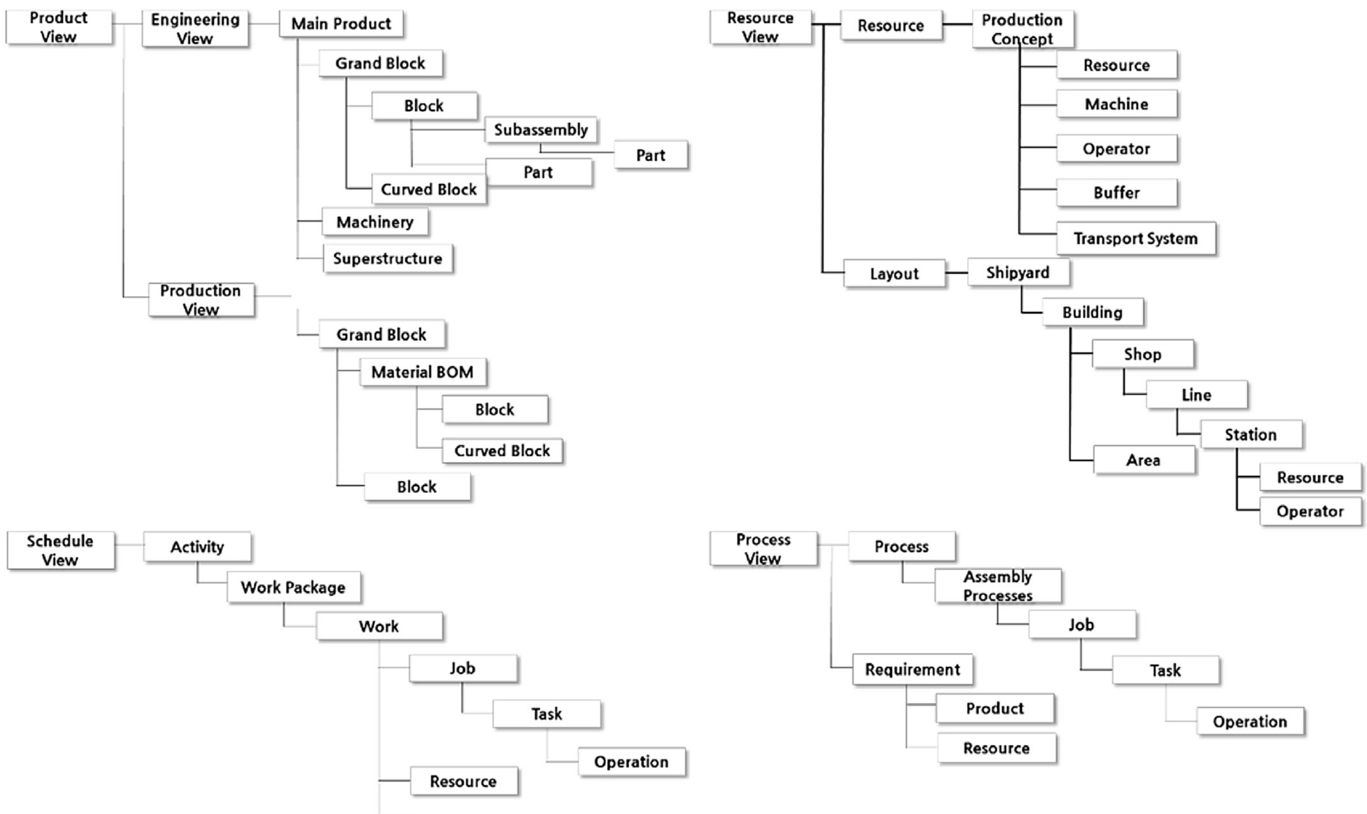

Fig. 4. Production simulation data model structure of shipyard from PPRS perspective.

of difficulty. Based on this, basic data attributes necessary for simulating the data model were specified, and data necessary for simulation modeling independently of shipyard data were extracted as described in the previous study on simulation modeling (Lee et al., 2011a). The data schema defined for simulation modeling was redefined from the Product, Process, Resource, Schedule, Model (PPRS-M) views by adding the data required for the simulation model view to the Product, Process, and Resource (PPR). Generally, this is required for production simulation, and schedule (S), reflecting the particularity of the shipbuilding industry where each product has its own schedule.

Table 1 describes the data schema used for the automated simulation model generation. Entity-relationship Diagram (ERD), which depicts the inter-data relationships, was designed from the PPRS-M view, as shown in Fig. 5. First, in regard to the product view, the vessel is a complex structure characterized by complicated vertical structural relationships among its component parts. A product is generally classified into the pre-erection block, grand block, middle block, sub-block, assembly, member, equipment in accordance to procedure and size, and panel block and curved block according to shape. The attributes of each block can vary from product to product, and these attributes are divided into basic data, such as size, mass, and derivative data, such as the welding length and shape detail, derived after the implementation of the production design. In addition, product data vary, even for the same product, depending on the update version (Lee et al., 2011b). The product data model for schedule verification simulation contains product type and shape definitions under ProductType and ShapeType, as presented in Table 1, including physical data, such as shape and mass. The production data on the vertical structure relationships are made identifiable by defining them as recursive relationships, and by connecting them with a foreign key in order to identify their relationship to the process, schedule, and model data. In the resource view, the design data are largely defined as equipment objects, such as buffer, machine, operator, and transport system, and combination equipment and space, such as the station, line, and shop, as illustrated in Fig. 5. Common attributes of single equipment involve position information, shape information, purpose, related CAD, and image data. Special attributes that differ according to the resource type are defined in separate tables. The transport system contains information on speed and the transport system, and the transport paths contain their respective coordinates. The work cell that denotes the space, in which various shipbuilding processes functions as a job-shop procedure, should indicate data on space, quantity, and capacity. Reflecting these conditions, the resource schema is designed by defining the common attributes in the resource table, and the special attributes that differ according to the resource type in separate tables, connected with the foreign key (Table 1). In the shipyard, the process information for vessel construction is structured in accordance to the vessel type, and managed according to the activity type that is connected in a vertical structure. An actual activity

Table 1
Detailed database schema for shipyard production simulation model generation.

View	Table name	Table description	Data		
			Roll	Name	Type
Model View	Simulation model	Management of simulation model information composed product, process, resource etc.	Primary key	SimulationModelID	varchar (20)
			Attribute	Name, Description	
	Configuration	Environment information of simulation model and execution composed unit, time etc.	Primary key	SimulationModelID	varchar (20)
			Attribute	Name, Description, Creator, TimeUnit, MassUnit, DistanceUnit, RunningTime, CreatedDate,	
	LinkedData	Management of file information related object like cad, image etc.	Primary key	LinkedDataID	varchar (20)
			Foreign key	RelatedObjectID, SimulationModelID	varchar (20)
			Attribute	RelatedObjectTable, Name, Description, Creator, DataSize, DataLocation, Purpose	
	Plant	Management of plant information including multiple resources like machine, transporter.	Primary key	PlantID	varchar (20)
			Foreign key	SimulationModelID, ParentShopID	varchar (20)
			Attribute	Name, Description, ShopType	
Product View	Product	Product information of block, member, material, equipment composing ship.	Primary key	ProductID	varchar (20)
			Foreign key	ParentProductID, SimulationModelID	varchar (20)
			Attribute	Name, Description, Quantity, ProductType, Length, Breadth, Height, ProjectCode, BlockCode, ShapeType, Tonage, volume, WeldingLength	
Process View	Process	Management of Process Information like cutting, assembly for ship construction.	Primary key	ProcessID	varchar (20)
			Foreign key	SimulationModelID, ParentProcessID	varchar (20)
			Attribute	Name, Description, ProcessType, ProcessLevelType	
	Product Requirement	Information of product requirement for process execution.	Primary key	ProductRequirementID	varchar (20)
			Foreign key	ProcessID, ProductID	varchar (20)
			Attribute	ProductType	
	Resource Requirement	Information of resource requirement for process execution.	Primary key	ResourceRequirementID	varchar (20)
			Foreign key	ResourceID, ProcessID	varchar (20)
			Attribute	ResourceType, ResourceQuantity	
Schedule View	WorkPackage	Management of work package schedule information in shipyard	Primary key	WorkPakageID	varchar (20)
			Foreign key	ProductID	varchar (20)
			Attribute	Name, Description, WorkPackageType, StartDate, EndDate	
	Work	Management of work unit schedule information in shipyard	Primary key	WorkID	varchar (20)
			Foreign key	WorkPackageID	varchar (20)
			Attribute	Name, Description, WorkType, StartDate, EndDate	
ResourceView	Resource	Management of resource	Primary key	ResourceID	varchar (20)
			Foreign key	SimulationModelID, Plant	varchar (20)
			Attribute	Name, Description, ResourceType, Capacity, Length, Breadth, Height, PurposeType,	
	TransportSystem	Facility information to transfer product Composed crane, transporter, forklift etc.	Primary key	ResourceID	varchar (20)
			Foreign key	ResourceID	varchar (20)
			Attribute	Speed, LoadedSpeed, CurveSpeed, RotationSpeed, Acceleration, Deceleration, TransportType, Accumulation	
	TransportPath	Path information to specify the path of the transport resource	Primary key	ResourceID, PointIndex	varchar (20) Integer
			Foreign key	ResourceID	varchar (20)
			Attribute	PorintX, PointY, PointZ, Width, Height, PathType	
	WorkCell	Information of work cell to manufacturing products	Primary key	ResourceID	varchar (20)
			Foreign key	ResourceID	varchar (20)
			Attribute	AreaCapacity, VolumeCapacity, QuantityCapacity	
	Location	Location information of individual resource composed position, translation, rotation	Primary key	ResourceID	varchar (20)
			Foreign key	ResourceID	varchar (20)
			Attribute	LocationX, LocationY, LocationZ, TranslationX, TranslationY, TranslationZ, RotationX, RotationY, RotationZ,	
	UserAttribute	Multiple Attribute information having individual resource.	Primary key	ResourceID, AttributeIndex	varchar (20)
			Foreign key	ResourceID	varchar (20)
			Attribute	Name, Value	

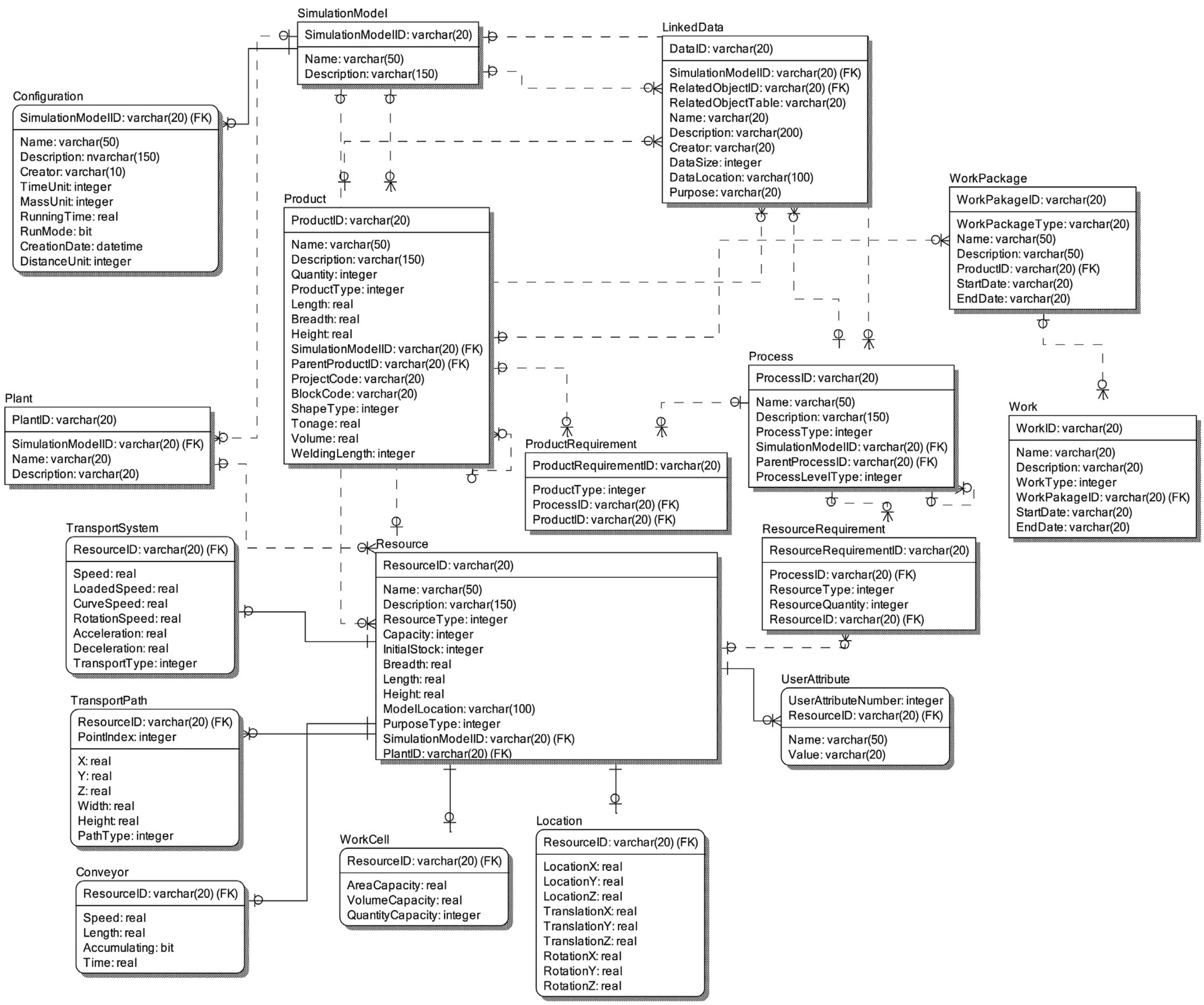

Fig. 5. ERD for shipyard production simulation model generation.

dataset is generated by combining the standard activity data and product-specific activity data. Correspondingly, each generated activity dataset thus contains activity-specific information. When all activity types for vessel construction are determined, each activity type is assigned to a specific factory and resource considering the onsite situation. At the same time, the schedule dataset is generated by inputting the start and end dates (Alfeld et al., 1998). The schedule planning is then completed by determining the resources for the implementation of the processes that have been assigned and scheduled. In order to reflect these features, the process table for activity management is arranged to express the vertical structure using the recursive relationship, as shown in Fig. 5. A schema was constructed to link the resource and product information for detailed activity determination. Given that shipyard work planning is set up at the middle or grand block or level, and not at the small-unit level, the schedule is defined as described in Table 1 with the work package using upper level types, such as cutting, processing, and assembly, and the work table with activity segmentation. According to previous studies on the definition of the simulation data model, data on the simulation model itself is necessary in addition to basic PPRS data for simulation modeling (Sly and Moorthy, 2001; Lee et al., 2011b). Drawing upon the NESIS and SDX implemented in previous studies on simulation of data models, data necessary for simulating the model management are defined by classifying them from the model view structure into data for managing individual simulation data, environmental parameters for implementing simulations, or by defining dimensions, time, and shape data related to model and equipment products (Table 1).

3.3. Automated modeling process

In order to define the procedure for system-based production for the simulation model generation, the modeling procedure and steps should be first analyzed by selecting a solution commonly used in shipyard production simulations.

Table 2
Simulation model generation process for shipyard assembly line.

Group	Simulation model making process	Assembly shop modeling
Data modeling	Information input data modeling	Data-set modeling for simulation <Block dimension/Block BOM/Ship infor/Block assembly network/Assembly activity schedule/Mid-term plan >
Geometry modeling	2D layout cad modeling	2D layout modeling <Assembly shop layout drawing/Stage layout drawing/Workcell layout drawing/Material handling resource route drawing>
	3D part CAD modeling	Ship block 3D CAD modeling <Grand assy block/Mid assy block/Sub assy block/Member >
	3D resource CAD modeling	Shop facility 3D CAD modeling <Gantry crane/Crawler crane/Work stage/Transporter >
	3D sub-resource CAD modeling	Shop sub-resource 3D CAD modeling <Building/Jig/Trestle>
Flow modeling	Process flow modeling	Work procedure and rule modeling <Assembly procedure/Inspection procedure/Preoutfitting procedure/ Prepainting procedure/Routing rule/Work stage priority rule>
	Logistic flow modeling	Material handling procedure and rule modeling <Crane/Transport procedure and priority rule >
Model implementation	Digital factory model implementation	Element relationship implementation <Part − resource relation/Process − resource relation/Part − process relation/Resource − resource connection>
Model validation	Digital factory model validation Digital factory model run test	
Application of simulation case & scenario	Scenario planning Scenario plan simulation	
Result Analysis	Result output data modeling Output data analysis	

For this study, we selected DELMIA D5 QUEST of the software company Dassault Systems that is extensively used as a solution in the shipbuilding industry. Many studies have been conducted using this solution for assembly shop productivity validation, mid-term scheduling validation, and transporter utilization prediction (Back et al., 2013; Woo, 2005; Lee et al., 2007). Drawing on the simulation model generation procedure of these studies, we analyze the modeling and simulation for a shipyard assembly shop using the selected solution. Table 2 describes the defined modeling procedure based on the results of this process. The work performed during the assembly shop modeling procedure is classified as the simulation model-making process regrouped by type, in order to perform mapping as an automated modeling process. As a result, a total of seven modeling groups are derived from which geometry modeling, flow modeling, and model implementation groups—that only use real modeler modules—are selected as candidate groups for automated modeling. Subsequently, their potential for automated modeling is assessed by verifying script-based implementation performance levels, and manageable tasks are analyzed and outlined.

Table 3 presents the task list derived based on the shipyard assembly shop modeling example. The automated simulation modeling process is finally selected based on the simulation data model for shipbuilding. For the real automated model generation, the QUEST environment configuration is necessary prior to reflecting the shipyard PPRS data, although this process was omitted in the modeling process from the given example. This process is necessary because all modeling situations cannot be monitored in the automated modeling process, unlike user-based modeling where the screen scale or modeling floor can be adjusted dynamically during the process. This difficulty is resolved in this study by the preprocessing scale, length, modeling area size, etc., as the chosen environmental parameters. The creation and setting are then implemented in the order of product, resource, and connection process. This order is set according to whether other elements are involved in the modeling of the respective elements. For example, the resource class creation process, given that it requires data on the part class to be generated, is placed after the creation of the product class. By applying this method, the number of scripts for automated modeling could be minimized, and the algorithm for the Batch Control Language (BCL) component necessary for system development could be simplified. The "execution control logic" described in Table 3 can be considered as the minimal logic necessary for model generation for the script sample that is actually used for the unit process.

4. Development of automated model generation system for shipyard simulation

4.1. System architecture

As illustrated in Fig. 6, the framework for the automated simulation generation system consists of three layers. The first layer is the user interface management layer that is equivalent to the user screen. The second layer consists of the legacy system controller responsible for extracting data using the legacy system and converting them into the system-specific

Table 3
Detailed simulation model generation procedure and sample.

Generation process	Related simulation resources	Execution control logic samples
1. Simulation model environment setting	–	SET GRID TO 50, 10000.0
		SET LENGTH UNITS TO METERS
		SET REPORT TIME UNIT TO SECONDS
2. Product class creation	Part Class	CREATE PCLASS ‘PART_A’
3. Resource class creation	Source Class	CREATE SOURCE CLASS ‘SRC_A’
		SET ‘SRC_A’ IAT TO 10
		SET ‘SRC_A’ PART FRACTIONS TO 10 FOR PCLASS ‘PART_A’
		SET ‘SRC_A’ ROUTE LOGIC TO ‘ROUTE_NB_NEXT_FREE_RES’
		LOCATE ELEMENT ‘SRC_A’ AT 0,0,0
	Buffer Class	CREATE BUFFER CLASS ‘BUFF_A’
		SET ‘BUFF_A’ QUEUE LOGIC TO ‘QUEUE_LIFO’
		SET ‘BUFF_A’ CAPACITY TO 100
		LOCATE ELEMENT ‘BUFF_A’ AT 2,0,0
	Machine Class	CREATE MACHINE CLASS ‘MC_A’
		LOCATE ELEMENT ‘MC_A’ AT 3,0,0,
	Sink Class	CREATE SINK CLASS ‘SNK_A’
		LOCATE ELEMENT ‘SNK_A’ AT 4,0,0
4. Material handling resource classcreation	Conveyor Class	CREATE EXTRUDED_CONVEYOR CLASS ‘CNV_A’
		CREATE LINEAR SEGMENT FOR ‘CNV_A’ FROM COORD 2.1,0,0 TO 2.9,0,0 []
		SET ‘CNV_A’ SPEED TO 10
	AGV Class	CREATE AGV_CONTROLLER CLASS ‘AGV_CON_A’
		CREATE AGV CLASS ‘AGV_A’
		SET ‘AGV_A’ CONTROLLER TO ELEMENT ‘AGV_CON_A’
		SET ‘AGV_A’ SPEED TO 30
		SET ‘AGV_A’ ACCELERATION TO 1
		SET ‘AGV_A’ DECELERATION TO 1
		SET ‘AGV_A’ CAPACITY TO 2
	AGV Path Class	CREATE AGV_PATH_SYSTEM CLASS ‘AGV_PATH’
		SET ‘AGV_PATH’ DIRECTION TO BIDIRECTIONAL
		CREATE LINEAR SEGMENT FOR ‘AGV_PATH’ FORM COORD 2.1,1,0 TO 2.9,1,0 []
		CREATE AGV_DEC_PT CLASS ‘AGV_DC_A’
		CREATE AGV_DEC_PT OF ‘AGV_DC_A’ ON SEGMENT ‘AGV_PATH’ AT OFFSET 30
	Labor Class	CREATE LABOR_CONTROLLER CLASS ‘LBR_CON_A’
		CREATE LABOR CLASS ‘LBR_A’
		SET ‘LBR_A’ CONTROLLER TO ELEMENT ‘ LBR_CON_A’
		SET ‘LBR_A’ SPEED TO 30
		SET ‘LBR_A’ CAPACITY TO 2
	Labor Path Class	CREATE LABOR_PATH_SYSTEM CLASS ‘LBR_PATH’
		SET ‘LBR_PATH’ DIRECTION TO BIDIRECTIONAL
		CREATE LINEAR SEGMENT FOR ‘LBR_PATH’ FORM COORD 2.1,2,0 TO 2.9,2,0 []
		CREATE LABOR_DEC_PT CLASS ‘LBR_DC_A’
		CREATE LABOR_DEC_PT OF ‘ LBR_DC_A″ ON SEGMENT ‘AGV_PATH’ AT OFFSET 30
5. Resource connection creation	–	CONNECT ELEMENT ‘SRC_A_1’ TO ‘BUFF_A_1’ []
		SET ‘SRC_A_1’ OUTPUT 1 RESTRICTED FOR PCLASS ‘PART_A’
6. Process class creation	–	CREATE CYCLE PRCESS ‘PROC_A’
		SET PROCESS ‘ PROC_A’ PART REQUIREMENT TO PART_CLASS
		‘PART_A’ 1 THRU SLOT 0
		SET PROCESS ‘ PROC_A’ PART REQUIREMENT TO ANY PART_CLASS ‘0
		SET PROCESS ‘PROC_A’ TIME TO 100
		SET PROCESS ‘PROC_A’ EXTERNAL PRODUCT TO 1 OF PCLASS ‘PART_B
		SET PROCESS ‘PROC_A’ PRODUCT_MODE TO DESTROY PCLASS ‘PART_A’
		SET ‘MC_A’ CYCLE PROCESS 1 TO ‘PROC_A’
7. Simulation run	–	SET ANIMATIO MODE OFF
		SAVE MODEL TO ‘C:\DELMIA_MODEL\MODELS\’
		SET SIMULATION INTERVAL TO 1
		RUN 1000
		SAVE STATS TO C:\TMP\REPORTS.XML
		EXIT

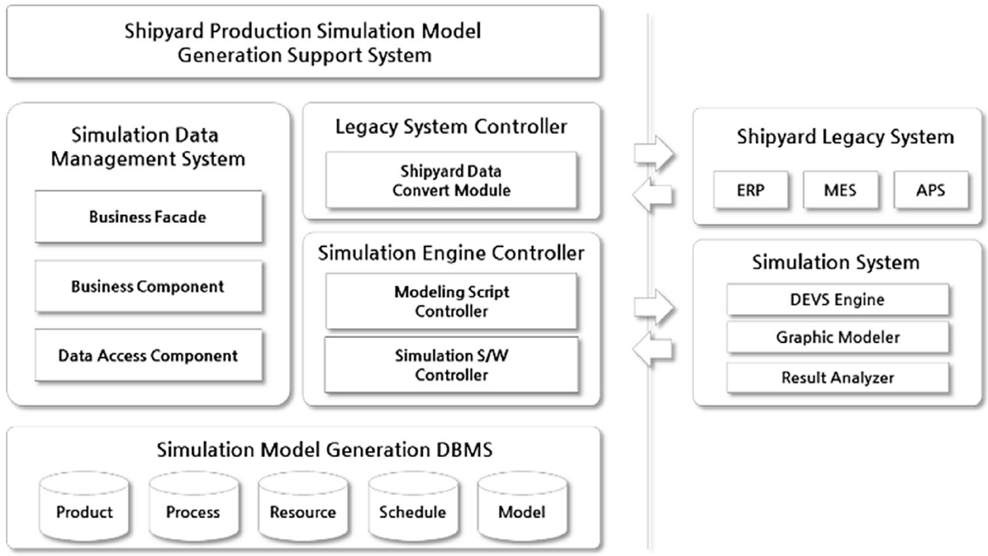

Fig. 6. Architecture of shipyard simulation model generation system.

data format, the simulation data management system responsible for compatibility checks of the extracted data and for the management of the Database Management System (DBMS), and the simulation engine controller responsible for generating commands for simulation model generation based on the converted data and controlling simulation software. The third layer is the DBMS responsible for storing and managing the data defined by the PPRS-M structure (Fig. 5).

Herein, we describe in more detail the second layer, that is, the legacy system controller. This controller extracts the data necessary for the simulation model generation from the shipyard legacy system. This consists of MES, ERP, APS, and others, that in turn contain most production information. The controller then converts them into the data for the simulation model generation that consists of the PPRS-M. The simulation data management system is the part that implements the behavior functions within the system, with the exception of the inter-system connection in the business façade, business component, and data access component layers. According to the Component Based Development (CBD) methodology, the business façade plays the role of the access point for the function based on the system design method, whereas the business component implements major functions, and the data access component implements the unit function related to the database connection (Whitehead, 2002; Oh et al., 2009).

The simulation engine controller is consisted of the modeling script controller that generates the simulation model generation command, based on the corresponding data in accordance to the procedure defined in Table 3, and the simulation S/W controller that provides socket communication with the simulation system, in order to generate the simulation model using the generation command. We define the simulation model generation process, the core function of the entire system, using the sequence diagram of Fig. 7. This is extensively used in defining the system flow among the Unified Modeling Language (UML) that is in turn, extensively used in system design. Fig. 7 presents the definitions of the interactions and interfaces between the legacy system controller responsible for data export from the shipyard legacy system chassis, the simulation data management system responsible for simulation model generation and correction, and the simulation engine controller responsible for the simulation engine control.

4.2. Deployment of simulation model generation system

As illustrated in Fig. 7, the simulation model generation system consists of the user interface that implements the screen-associated function for the user in order to verify the process, the legacy system controller that extracts data from the legacy system to generate the simulation model, the simulation data management system that checks the compatibility of the extracted data and manages the DBMS, and the simulation engine controller that controls the simulation model generation and the simulation system. The user interface and module development environment are implemented within the .net framework 4.0 C#, and the DBMS using the SQL Server 2008. First, the legacy system controller obtains the data necessary for the simulation model generation by extracting them from the shipyard legacy system as shown in Fig. 7. The obtained data are thus stored in the database defined from the PPRS-M views. The simulation data management system is implemented in three layers of a business façade, business component, and data access component. The simulation engine controller generates the simulation model generation command based on the data used for the simulation model generation by a user query. The generated command is configured to implement the simulation model generation using the socket communication method known to be stable among the APIs supported by the simulation engine. Lastly, the user interface has a layout that allows the user to employ the functions provided

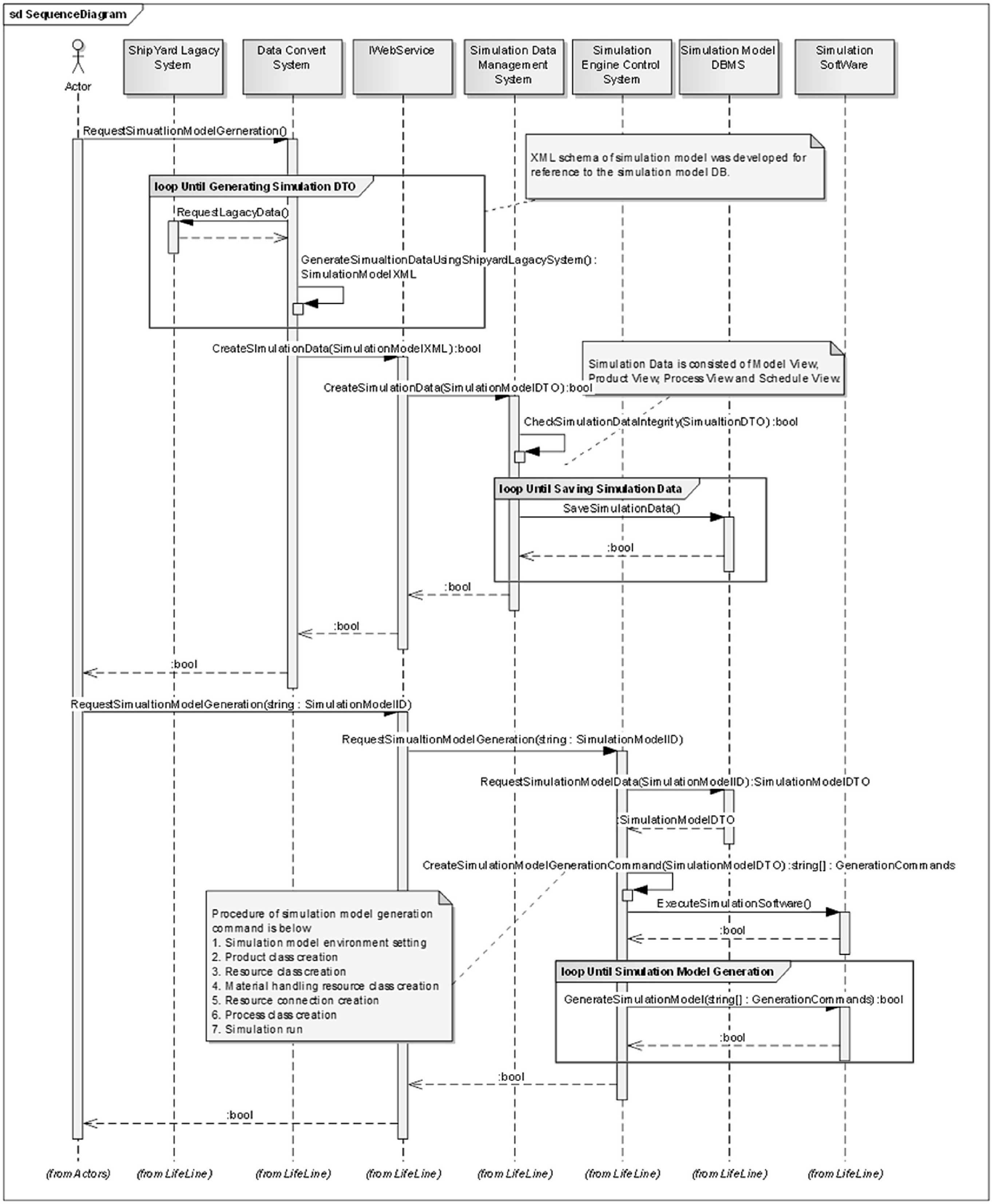

Fig. 7. System flow for shipyard simulation model generation using sequence diagram.

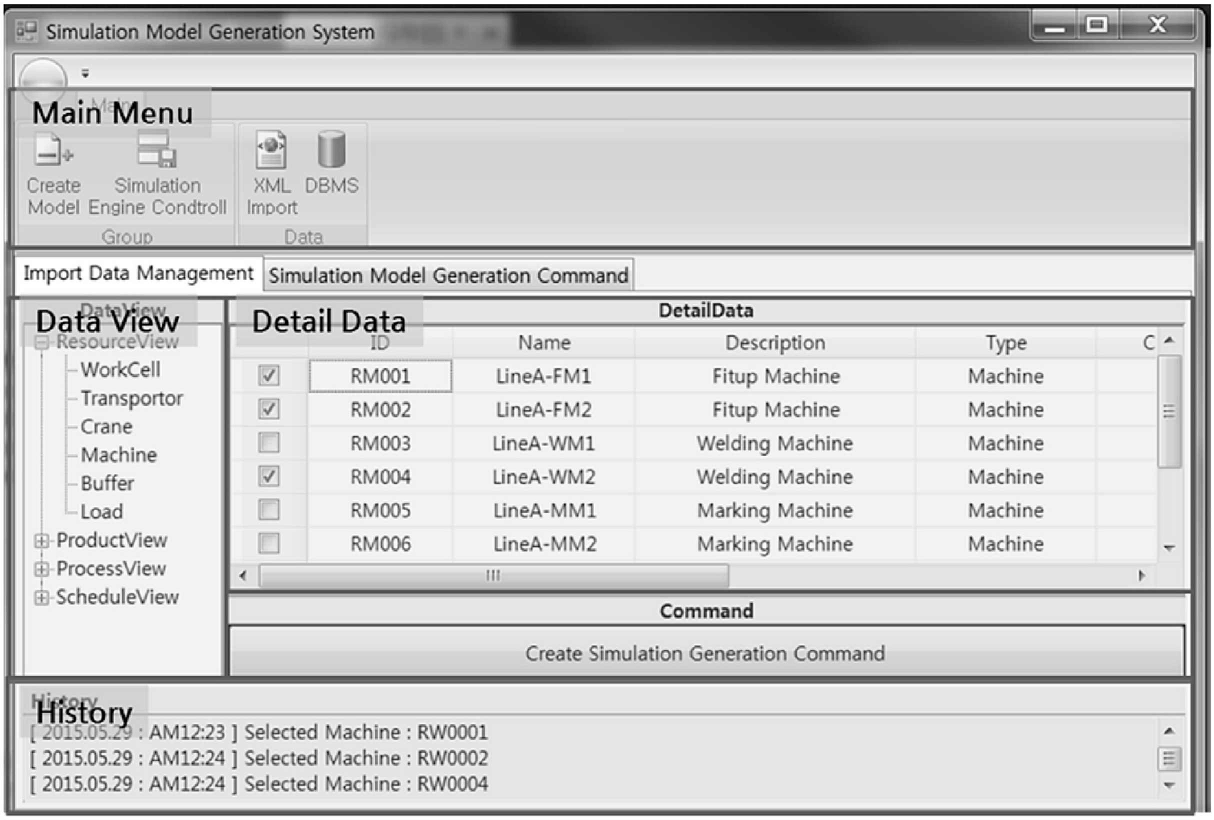

Fig. 8. User interface of automated model generation system.

by the legacy system controller, the simulation data management system, and the simulation engine controller, as shown in Fig. 8. The user obtains the data from the legacy system on the user interface, generates the command for the simulation model generation based on the shipyard data stored in the database, and generates a simulation model with the simulation engine controller.

5. Application of automated model generation system at shipyard panel line

5.1. Analysis of shipyard panel line

The vessels manufactured in a shipyard have a complex product mix and unique order-based pieces, with the

Table 4
Component of shipyard panel line.

Type	Detail type	Name	Description
Product	Member	Plate	Member of composing main plate
		Tap Piece	Rectangular piece of steel used to weld both plates
		Main Plate	Product consisting of multiple plates
		Stiffener	Longitudinal stiffener
	Block	SA-Block	Block of subassembly consisting main plate and stiffener to produce middle block
Resource	Machine	Welding	Line welding machine to assembly product.
		Marking	Machine to work for marking to identify products
		Fitup	Spot welding machine to hold member like stiffener
	Crane	60T OHC	Over head crane used for turn over
		20T OHC	Over head crane used for transfer light product like plates, stiffeners
	Conveyor	Conveyor	Accumulating conveyor to transport main plate and SA-block
Process	Assembly	Tack Welding	Process to arrange plates, tap piece and weld them partially
		Panel Front Welding	Front side welding process for making main plate
		Turn Over	Process to flip over front side to weld plate back side
		Panel Back Welding	Back side welding process for making main plate
		NC Marking	Making process of where to put the stiffeners
		Cutting	Process to cut tap-piece
		Assembly − Fitup	Assembly process of weld stiffeners partially on the main plate
		Assembly − Welding	Assembly process of weld stiffeners on the main plate

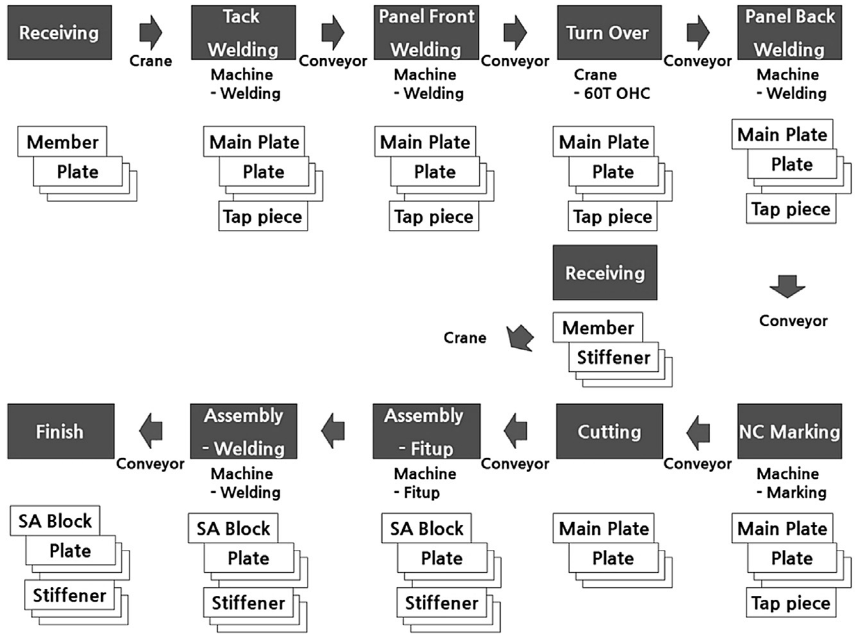

Fig. 9. Process flow analysis considering resource and product.

exception of some series lines. Therefore, most of the shipbuilding processes are job-shop methods with a typically low productivity. However, a panel line adheres to an iterative process for producing panel blocks of similar shapes and its production process can be standardized using the flow production method that allows mass production. Consequently, we chose the panel line as the target process for applying the proposed system.

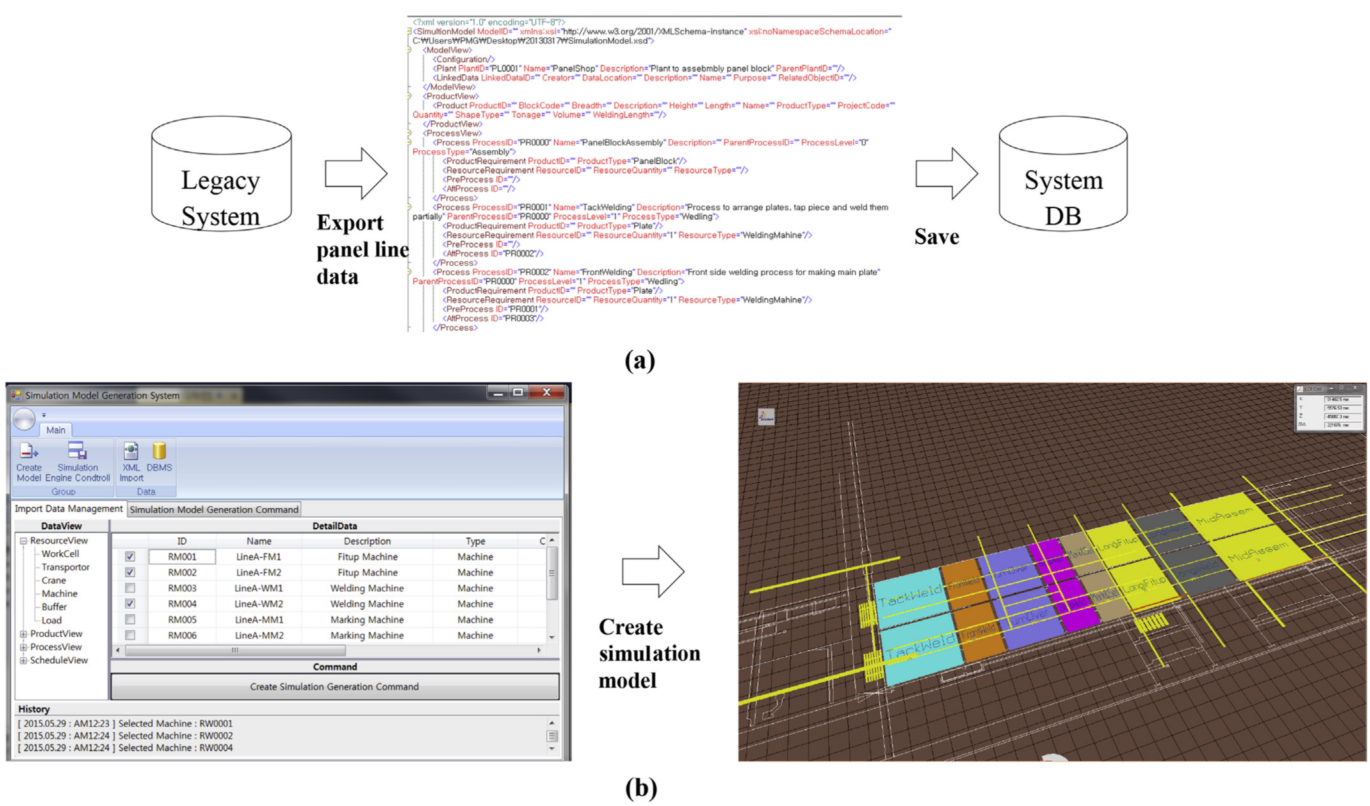

Fig. 10. Data acquisition from legacy system and simulation model generation using system.

For the definition of the major elements of the panel line, we classified them into product, process, and resource, as described in Table 4. The product consists of a main plate that constitutes the most sub member, produced by the welding of other plates, and a subassembly block produced by welding of the main plate and the stiffener. The process consists of tack welding, panel front welding, turn over, panel back welding, NC marking, cutting, assembly fit up, and assembly welding. Fig. 9 presents the main equipment for an individual process and transport equipment, as well as related product information. First, the main plate is produced by welding the tap piece to fix several additional plates, and a welding machine is used to close the front side plate-to-plate gaps. Rear-side welding is then performed after turning the plates over with a crane. A marking machine is used to mark the position of the stiffener on the main plate, and the stiffener is prewelded with a fit up machine. A subassembly panel is then completed by welding the prewelded piece with a welding machine. The main line flow of the panel line occurs through an accumulation-type conveyor, and the members, such as stiffeners and plates, are moved with a 20 T overhead crane. A panel block is produced by assembling several subassembly blocks, but this process is excluded from the system application in this study.

5.2. Application of an auto model generation system

We validated the simulation model generation system to the shipyard panel block production line simulation model. Among the individual systems that constitute the overall system framework, the data convert system is used for extracting the data necessary for simulation modeling using the legacy system. Given that shipyards have different data management methods and data structures, and maintain high-information security, directly obtaining data from a shipyard legacy system is a great challenge. Therefore, we obtain the data from a shipyard legacy system in the neutral XML format (Fig. 10(a)) for simulation (Lee et al., 2013). The obtained simulation model data is converted into the format and function compatible with the proposed system via the simulation data management system. Moreover, the BCL, the simulation model generation command, is generated (Fig. 10(b)) by the simulation control system based on the converted data. Prior to the simulation model generation, the BCL generates and then implements the initial simulation instant data definition, such as the temporal and spatial unit configuration and the initial model dimensions, using the data from the simulation view of the shipyard data model. Subsequently, it implements product class generation, attributes assignment, and size configuration, using the produced view data. Using the resource view data, resource data are generated, such as resource class generation, resource instance generation, attributes assignment, and resource connection. With the BCL implementing these functions, the shipyard panel line simulation model that is composed of the product, process, and resource (shown in Table 4) is generated using the simulation engine controller. Given that no separate model for schedule representation is available, this is defined as the attribute of the individual products. In addition, data that cannot be generated with BCL, such as CAD, are generated preliminarily and used in the form of a library.

The performance test of the simulation model generation system is carried out in an environment with Intel i5 CPU, 4 GB RAM, and 1 GB VGA RAM. The test simulation model is comprised of resources (source, machine, buffer, sink, crane, and path) and a process, which are entities defined in QUEST with 783 classes and 875 class elements. After the entities are generated, the pre- and post-link relations between the elements are set. The simulation model generation system generates about 7000 BCL commands for simulation model generation. Approximately 1 min is required for simulation model generation using the BCL commands in the test environment. To compare this with the modeling time required by an engineer using a general method, scenarios of model generation by a skilled engineer and an unskilled general engineer are defined. It is assumed that a skilled engineer carries out 6 BCLs/min using the user interface and the unskilled, beginner modeler carries out 3 BCLs/min. Under this assumption, engineer using this system for model generation saves approximately 2.6 h than the skilled modeler and 10.2 h than the beginner modeler.

The simulation model generation is completed if detailed modeling is performed by adding rules to the initially generated simulation model, specific to individual shops. However, because detailed modeling is implemented via script coding using a high degree-of-freedom, the resulting model qualities vary greatly from one modeler to another. Furthermore, because of the high degree of modeling complexity, its standardization is a great challenge. The system developed in this study has a feature that allows the generation of a model by BCL, to generate the initial model, based on the predefined data, while. A simulation model generated according to this method requires less modeling time than a conventional modeler. It is expected that the modeler can focus more on the detailed modeling that reflects work-specific rules. In addition, the proposed method has the advantage of preventing simulation model generation for each modeler with other methods, thus contributing to the standardization of the shipyard simulation model generation.

6. Conclusions

In this paper, the data model for shipyard production simulation model generation was defined by analyzing the iterative simulation modeling procedure. The shipyard production simulation data model defined in this study contains the information necessary for the conventional simulation modeling procedure and can serve as a basis for simulation model generation. We developed a system capable of generating simulation models using the internal model generation protocol (BCL), and socket communication by applying the constructed model for simulation model generation to QUEST, a solution extensively used for production simulation. The efficacy of the developed system was validated by applying it to the simulation model generation of the panel block production line for company S. By implementing the initial simulation model generation process, which was performed in the past with a simulation modeler, the proposed system substantially reduced the modeling time. In addition, by reducing the difficulties posed by different modeler-dependent generation methods, the proposed system makes the standardization of the simulation model quality possible. This leads to another advantage, in that, the simulation modeler can focus more on detailed modeling that reflects work-specific rules using script coding.

In regard to the limitations of the proposed system, we can indicate that this is a basic simulation model generation methodology for individual projects, and cannot manage the logic generation for expressing detailed work rules. Work rules are difficult to standardize because of different features and flow on a case-by-case. This will have to be addressed in future research that focuses on modeler development using natural language and diagrams. Another limitation of the proposed system is that it was not applied to multiple production simulation solutions. Nevertheless, despite these limitations, the development of this system is significant in that it simplified the modeling region using technology based on a neutral data format converted from production data. It stores data required for the shipbuilding simulation model generation and defines manageable data models, thus contributing to the reduction of the total modeling time when used for simulation model generation and maintenance/repair. The methods applied in this study are expected to have time and cost reduction effects when applied to digital production techniques in shipyards, which will lead to more active applications of simulation technologies.

Acknowledgments

This paper is composed of results of the “Leaders Industry-university Cooperation” Project, supported by the Ministry of Education, Science & Technology (MEST) and the National Research Foundation of Korea (NRF). Also, the national project (Development of the simulation based production management system for the middle-sized shipbuilding companies; No. 10050495), which is supported by industry core technology development business of ministry of trade, industry and energy, has been supporting this research.

References

Alfeld, L.E., Pilliod, C.S., Wilkins, J.R., 1998. The virtual shipyard: a simulation model of the shipbuilding process. J. Ship Prod. 14 (1), 33–40.

Back, M.G., Kim, Y.G., Hwang, I.H., Lee, K.K., Ryu, C.H., Shin, J.G., 2013. Design and development of scenario-based simulation system to improve shipbuilding execution scheduling assessment : a case study on panel line. Trans. Soc. CAD/CAM Eng. 18 (3), 211–223.

Harward, G., Harrell, C., 2006. Assessment of the NIST shop data model as a neutral file format. In: Proceedings of the 2006 Winter Simulation Conference, Monterey, USA, pp. 941–946.

Hwang, I.H., Song, J.K., Back, M.G., Ryu, C.H., Lee, K.K., Shin, J.G., 2013. Development of shipbuilding execution scheduling support system using mobile device : a case study for a panel block assembly shop. J. Soc. Nav. Archit. Korea 50 (4), 262–271.

Johansson, M., Johansson, B., Sjoogh, A., Leong, S., Riddick, F., Lee, Y.T., Shao, G., Klingstam, P., 2007. A test implementation of the core manufacturing simulation data specification. In: Proceedings of the 2007 Winter Simulation Conference, Washington, USA, pp. 1673–1681.

Kang, H.S., 2007. A Study on the Automated Generation of Simulation Model Based on PPR Information. Dissertation. Sungkyunkwan University.

Lee, C.J., Lee, J.H., Woo, J.H., Shin, J.G., Ryu, C.,H., 2007. A study on discrete event simulation of shipyard outdoor block movement. J. Soc. Nav. Archit. Korea 44 (6), 647–656.

Lee, D.K., Back, M.G., Lee, K.K., Park, J.S., Shin, J.G., 2013. Study on simulation model generation of a shipyard panel block shop using a neutral data format for production information. J. Soc. Nav. Archit. Korea 50 (5), 314–323.

Lee, D.K., Kim, Y.M., Hwang, I.H., Oh, D.K., Shin, J.G., 2014. Study on a process-centric modeling methodology for virtual manufacturing of ships and offshore structures in shipyards. Int. J. Adv. Manuf. Technol. 71, 621–633.

Lee, J.H., Kim, S.H., Lee, K.H., 2011a. Integration of evolutional BOMs for design of ship outfitting equipment. Comput. Aided Des. 44 (3), 253–273.

Lee, J.Y., Kang, H.S., Noh, S.D., Woo, J.H., Lee, P.L., 2011b. NESIS: a neutral schema for a web-based simulation model exchange service across heterogeneous simulation software. Int. J. Comput. Integr. Manuf. 24 (10), 948–969.

Lu, R.F., Qiao, G., McLean, C., 2003. Nist XML simulation interface specification at boeing : a case study. In: Proceedings of the 2003 Winter Simulation Conference, LA, USA, pp. 1230–1237.

Oh, D.K., Shin, J.G., Choi, Y.R., Yao, Y.H., 2009. Development of a naval ship product model and management system. J. Soc. Nav. Archit. Korea 46 (1), 43–56.

Song, Y.J., Lee, D.G., Choe, S.W., Shin, J.G., 2009. A simulation-based capacity analysis of a block-assembly process in ship production planning. J. Soc. Nav. Archit. Korea 46 (1), 78–86.

Song, Y.J., Woo, J.H., Shin, J.G., 2011. Research on systematization and advancement of shipbuilding production management for flexible and agile response for high value offshore platform. Int. J. Nav. Archit. Ocean Eng. 3 (3), 181–192.

Sly, D., Moorthy, S., 2001. Simulation data exchange (SDX) implementation and use. In: Proceedings of the 2001 Winter Simulation Conference, Arlington, USA, pp. 1473–1477.

Watson, E.F., Medeiros, D.J., Sadowski, R.P., 1997. A simulation-based backward planning approach for order-release. In: Proceedings of the 1997 Winter Simulation Conference, Atlanta, USA, pp. 765–772.

Wang, P., Mohamed, Y., Abourizk, S.M., Rawa, A.T., 2009. Flow production of pipe spool fabrication: simulation to support implementation of lean technique. J. Constr. Eng. Manag. 135 (10), 1027–1038.

Whitehead, K., 2002. Component-based Development: Principles and Planning for Business Systems. Pearson Education, London.

Woo, J.H., Lee, K.K., Jung, H.R., Kwon, Y.D., Shin, J.G., 2005. A framework of plant simulation for a construction of a digital shipyard. J. Soc. Nav. Archit. Korea 42 (2), 165–174.

Woo, J.H., Song, Y.J., Shin, J.G., 2009. Research on a simulation-based ship production support system for middle-sized shipbuilding companies. Int. J. Nav. Archit. Ocean Eng. 1, 70–77.

Woo, J.H., 2005. Modeling and Simulation of Indoor Shop System of Shipbuilding by Integration of the Product, Process, Resource and Schedule Information. Dissertation. Seoul National University.