©SNAK, 2009

Inter J Nav Archit Oc Engng (2009) 1:78~88
http://dx.doi.org/10.2478/IJNAOE-2013-0010

Development of software for computing forming information using a component based approach

Kwang Hee Ko[1], Jung Seo Park[2], Jung Kim[2], Young Bum Kim[2] and Jong Gye Shin[3]

[1]*Dept. of Mechatronics, Gwangju Institute of Science and Technology, Gwangju, Korea*
[2]*Dept. of Naval Architecture and Ocean Engineering, Seoul National University, Seoul, Korea*
[3]*Research Institute of Marine Systems Engineering, Seoul National University, Seoul, Korea*

ABSTRACT: *In shipbuilding industry, the manufacturing technology has advanced at an unprecedented pace for the last decade. As a result, many automatic systems for cutting, welding, etc. have been developed and employed in the manufacturing process and accordingly the productivity has been increased drastically. Despite such improvement in the manufacturing technology, however, development of an automatic system for fabricating a curved hull plate remains at the beginning stage since hardware and software for the automation of the curved hull fabrication process should be developed differently depending on the dimensions of plates, forming methods and manufacturing processes of each shipyard. To deal with this problem, it is necessary to create a "plug-in" framework, which can adopt various kinds of hardware and software to construct a full automatic fabrication system. In this paper, a framework for automatic fabrication of curved hull plates is proposed, which consists of four components and related software. In particular the software module for computing fabrication information is developed by using the ooCBD development methodology, which can interface with other hardware and software with minimum effort. Examples of the proposed framework applied to medium and large shipyards are presented.*

KEY WORDS: Thermal forming; ooCBD; Forming information; Software architecture; Automation; Ship fabrication.

INTRODUCTION

In shipbuilding industry, the technology for automatic manufacturing has been developed at an unprecedented pace so far. In particular various operations such as welding, cutting and erection are performed automatically and increases productivity in practice. For example, use of small welding robots has led to a remarkable increase of efficiency in welding stiffners for a double hull. In cutting, automatic plasma cutting machines have improved the precision of cutting and at the same time reduced the production time. Automation for other manufacturing operations is now being developed. Such effort results in high efficiency and productivity in ship production and eventually strengthens a company's competitiveness.

However, automation of curved hull fabrication, which is one of the most critical manufacturing processes in shipbuilding, remains at the beginning stage. Given the fact that 30%~70% of a typical ship hull consists of curved parts and all the curved parts are fabricated manually in current practice, full automation of this process is expected to contribute to the improvement of overall productivity. So many companies and research laboratories around the world have done much research on the automation but failed to produce any meaningful result.

The most important thing in the curved hull fabrication is to know which part of a plate should be formed in order to fabricate a desired shape. Finding out this information is the most critical step in the automation. Currently, such forming information is obtained from experience or know-how of skilled workers and has not been compiled in a quantified manner. Therefore, automation of obtaining such forming information is needed for the full automatic fabrication process.

Computation of forming information for fabricating a desired curved plate is a reverse engineering problem with multiple solutions involving a wide spectrum of different engineering fields such as solid mechanics, heat transfer and geometric modeling. Shin, Ryu and Nam (2004) and Shin and Ryu (2000) established the theoretical background for computing forming information based on heating using geometrical information and provided a computation algorithm. This algorithm, however, may not be directly applied to a real automatic system due to different requirements and conditions of the manufacturing process of each shipyard. Instead, it should be used to develop a software module for forming information computation

Corresponding author: *Jung Seo Park*
e-mail: *emot97@snu.ac.kr*

tailored for a fabrication process and then applied for automation.

The software of computing forming information cannot be considered as an independent software routine but as an integral part of the whole automation system consisting of measurement, design and form hardware systems as shown in Fig. 1, which illustrates the overall conceptual structure of an automatic curved hull plate fabrication system. Different component systems may be used by each shipyard and the forming process may be different accordingly. However, the overall conceptual structure of the forming system can be assumed to be the same.

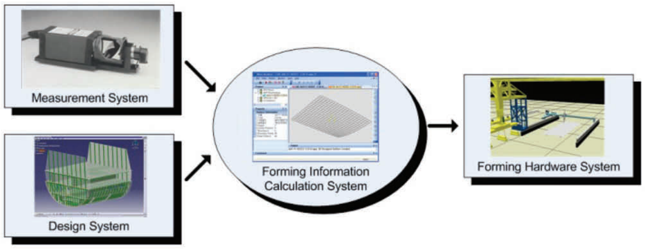


Fig. 1 The overall structure of an automatic curved hull plate fabrication system.

One-time forming of a curved plate may be required by a certain shipyard or iterative forming may be needed at another shipyard to obtain the best result. Due to such diverse conditions, the specification of each device should be taken into consideration and the interfaces of the computation algorithm to each component should be well defined. Namely, the data exchange among the components with different in/out data formats should be performed consistently. In order to handle such heterogeneous components effectively, two different approaches can be considered. The first one is to construct software modules specific for each case and the other is to construct a software system based on modular structure consisting of diverse components that are flexible enough to allow a minimum modification to meet different requirements for the interfaces of each component. Between the two methods, the latter has been chosen for the development of a software system for forming information computation in this paper.

The selected approach can be realized by using the CBD (Component-Based Development) methodology. The CBD methodology is a systematic approach which decomposes a system into either logical or functional components and provides interfaces among the components for communication (Sommerville, 2001). Each component works for itself and interacts with others through the interfaces. The component is more an abstract unit than an object in the method.

There exist a few different CBD methods. They emphasize critical advantages such as reusability, interoperability and upgradeability over the conventional software development methods (Qureshi and Hussain, 2008). The CBD methodology has been widely used in the development of diverse software such as finite element program (Dolenc, 2004). Different approaches based on the methodology are available such as RUP(Rational Unified Process (Kruchten, 2003), SELECT Perspective™ (Allen and Frost, 1998) and ooCBD (Jeon, 2004). The RUP method is very formal, intended to design a large-sized system and requires extraneous effort in developing a system up to a certain size. Therefore, it may not be appropriate to use such a method to develop a small sized project like a forming information computation system. Among those methods, the ooCBD methodology has been applied for a small sized project without much difficulty.

A software module can be categorized into four different units depending on the size: function/routine, class, component and application package, and they have a relation with respect to cost-effectiveness and flexibility as follows (Jeon, 2004; Katharine, 2002):

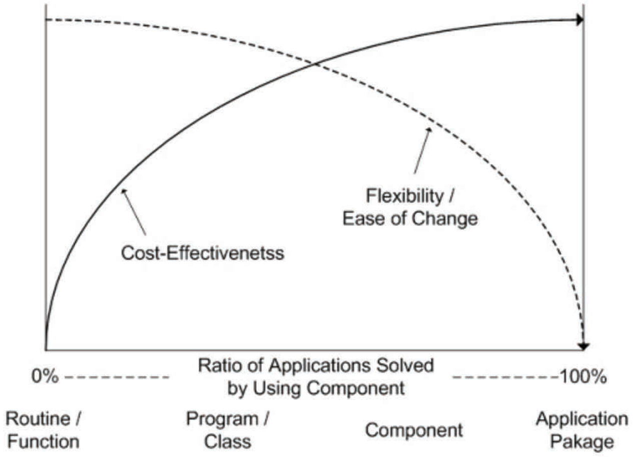


Fig. 2 Performance along with constituent elements of software (Jeon, 2004 and Katharine, 2002).

As shown in Fig. 2, much effort can be saved by using an application package which is specialized for a certain target application. In this manner, the use of an application package becomes cost-effective since it provides all the necessary functions of the target application. However, it is not easy to find the most appropriate package which can meet different specific requirements. Namely, the flexibility and ease of modification, which are two essential features in handling various requirements, can be compromised. On the other hand, software modules in small size (routine/function) can be flexible since they can be reused for different purposes easily. However, using such modules cannot be cost-effective since more and more small modules should be modified as the size of them becomes smaller when modification is required. From this observation, it is noted that the size of a software module should be determined by taking into account both cost-effectiveness and flexibility, and a software module of "component" size, which lies between "program/class" and "application package", is considered to be optimal in terms of cost-effectiveness and flexibility (Jeon, 2004).

Hence, the ooCBD approach is considered the most appropriate choice since it is the methodology based on components each of which is a software module providing independent services. It has all the advantages of the CBD methodology as well as conforms to COM+ (component object module +) (Rogerson, 1997) specifications to promote explicability and flexibility in the system development.

In this paper, the development of forming information computation software is presented using the ooCBD methodology. It is shown how the software, a central component of the forming automation system, interacts with subsystems through different interfaces such as measurement, design and heating systems. The software is made to have a modular structure that allows easy interactions with other different software and hardware necessary for constructing the whole forming automation system. This type of structure, what is called the "plug-in" structure, enables the software to be used for various types of shipyards with minimum modifications. In addition, using the ooCBD methodology, documents for the software are compiled and maintained systematically, so the maintenance of the software can be performed methodically.

This paper is structured as follows: in Section 2, an overview of the whole forming system is briefly introduced and some details of the forming information calculation software along with other subsystems are explained. In Section 3, how the software architecture has been developed using the ooCBD methodology. Then, the issues associated with data specification between the software and subsystems are discussed when the architecture is used in the real automation system in Section 4. Two examples are introduced in Section 5 where how the developed software has been applied to a medium and a large sized shipyards. Section 6 concludes this paper with future work.

AN AUTOMATIC HULL PLATE FABRICATION SYSTEM BASED ON PLUG-IN CONCEPT

Overall System Structure

An automatic curved hull plate forming system is composed of four sub-systems: a heating system, a measurement system, a design system and a forming information calculation system. The overall relations between them are depicted in Fig. 1. At the center of the system is the forming information calculation system, a software unit which accepts input from the measurement and design systems to compute forming information. Then the computation system forwards commands to the forming system to form a plate to a desired shape. Once the forming is finished, the measurement system measures the shape of the fabricated plate and generates the shape information for further computation.

Although the conceptual structure of the automation system can be represented by using the four abstract units, constructing the actual system is much more involved. Namely, the characteristics and features of each subsystem are in general different from shipyard to shipyard. Therefore, the framework of the overall system should be designed to be least dependent upon the hardware units (forming and measurement units) and the design system. The central part of this framework, the forming information calculation system, is constructed using components which can be adaptively modified to handle different subsystems. The ooCBD methodology is used for the development of software architecture for the calculation system and by composing the components conforming to COM+ rules, a system based on the concept of "plug-in" can be developed. The proposed framework is depicted in Fig. 3. In the figure different types of measurement devices are provided and they interact with the forming information computation system through a unified interface offered by a component of the computation system. So does the design and forming systems. Using this framework, each component, no matter what type of the component is used, can be plugged into the forming information computation system and interact with it.

Subsystems

In this section, the characteristics of each subsystem for the automatic forming system are discussed.

Forming Information Calculation System

Unlike the forming and measurement systems which are the hardware components in the whole automation system, the forming information calculation system is a software component which serves as a brain of the entire automation system. Because of this reason, the development of the forming information calculation system is very important to achieve the automation of the entire system but is a difficult task since it requires a lot of effort and experience including a theoretical aspect (thermo-mechanical analysis), a reverse engineering technique and input from field workers.

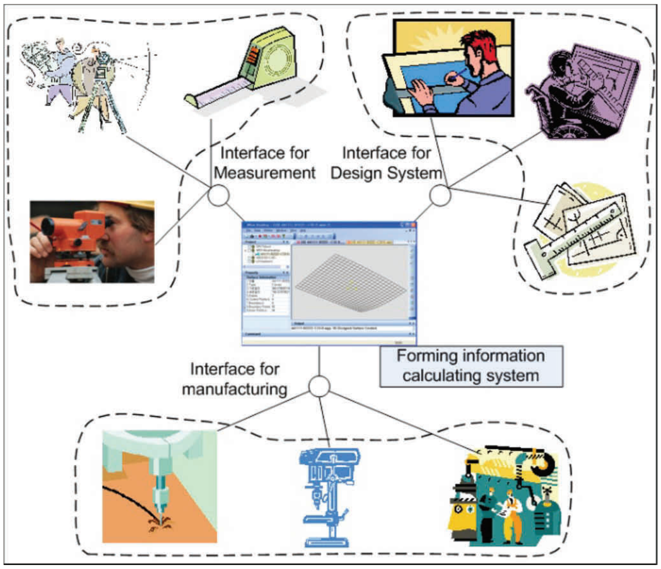

Interface for Measurement
Interface for Design System
Interface for manufacturing
Forming information calculating system

Fig. 3 The automation system for forming a curve plate using the plug-in concept.

This is one primary reason why many research institutes and companies have failed in developing an automated forming system. Without such a software module, the whole system is at best a mechanized system, not a real automated one.

Calculation of forming information has been researched by a few research groups. The study of Shin and Ryu (2000), which proposed a method to produce the information by using kinematics analysis, is frequently referenced in the area of thermal forming in ship manufacturing. In the paper, a designed surface is developed to a flat plate, so deformation between the designed surface and the flattened plate can be represented using strains. The strains are divided into the bending and in-plane strains and the forming information can be calculated based on these strains. Based on this result, the potential for the automation of iterative hull forming was demonstrated by employing surface comparison and mapping data extraction algorithms proposed by Park, Shin and Ko (2007). In addition there are several studies of forming information creation based on FEM analysis (Dolenc, 2004). These previous results are used as essential theoretical tools in developing a forming information computation system and in this paper such a system is proposed based on the algorithm by Park, Shin and Ko (2007) and Park, Shin, Ko, Hyun and Doh (2007).

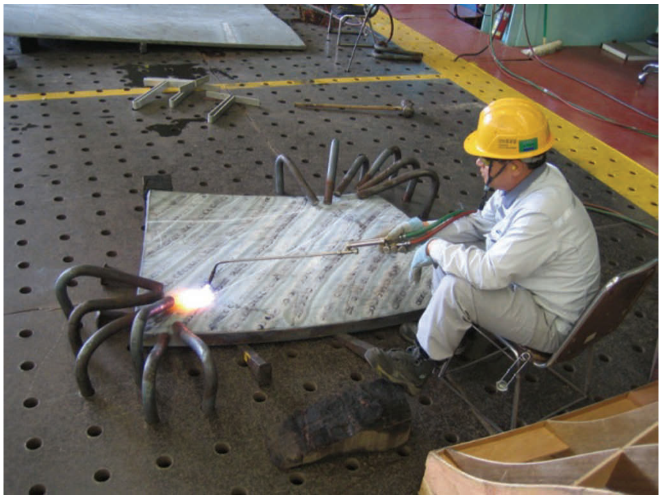


Fig. 4 Thermal bending method by skillful worker (Park, Shin, Ko, Hyun and Doh, 2007).

Forming System

Thermal forming is primarily used in most shipyards these days. It is a method to induce angular deformation to a plate by heating and cooling a local area of the plate, and yielding shrinkage. A method using a gas torch has been widely used for thermal forming in most of shipyards for several decades and a significant amount of know-how for using the tool has been accumulated over the years. In addition, developing an NC (Numerical Control) heating machine using a gas torch is regarded relatively easy since it is not difficult to simulate the worker's manual movement of the torch as shown in Fig. 4. One example of an NC heating machine using a gas torch is shown in Fig. 5, which has been developed by a Korean shipyard. However, difficulty in controlling the amount of heat and generation of ear-deafening noise are two drawbacks of the gas torch based heating system.

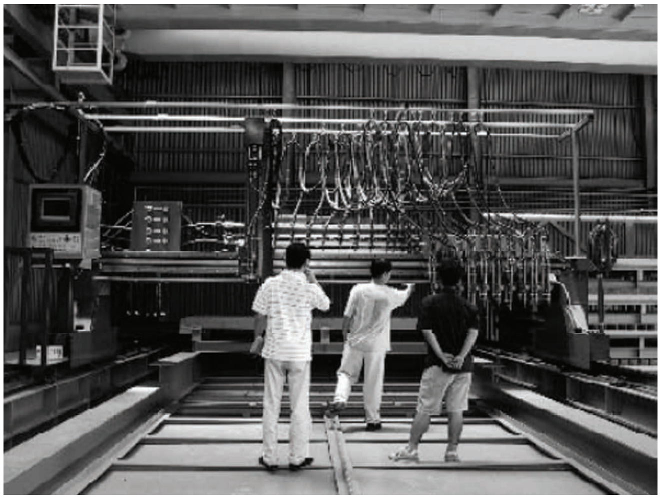


Fig. 5 Hull plate forming system in the 21C shipyards (Shin, Ryu and Nam, 2004).

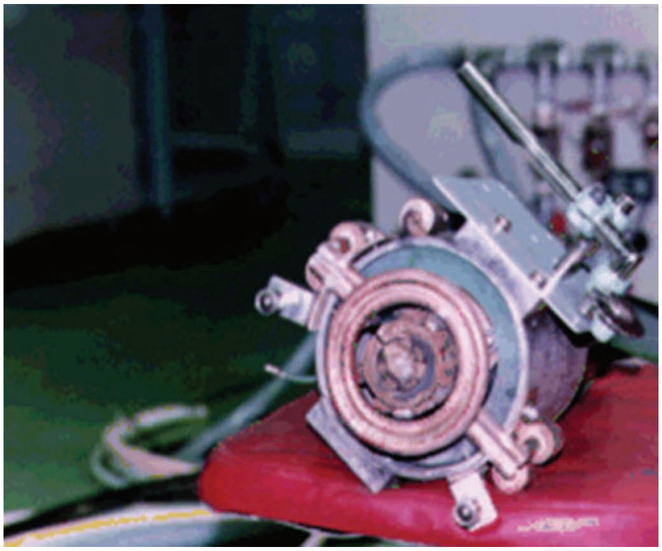


Fig. 6 The coil of an induction heating machine (Jang, Kim and Ha, 2002).

To overcome the drawbacks of torch-based heating, high frequency induction heating has recently been attracting many developers' attention. This method generates heat by creating a magnetic field of a high frequency, inducing resistance and in turn heat in the plate. This approach is recognized as a heating method that can replace the torch-based heating method in that the amount of heating can be controlled easily by controlling electric power and the noise can be maintained at a minimum level. One example of a high frequency induction heating device is shown in Fig. 6. In order to construct such a heating system, a significant amount of investment is required. Therefore, medium and small shipyards may not afford such a heating system for their manufacturing process.

Measurement System

Measurement is essential for the automation of the curved hull forming process. It consists of measurement of a physical plate and alignment of the plate against a designed one. For this measurement, various types of devices have been used in practice. Shin, Lee and Nam (2004) proposed a measurement algorithm for a surface plate of large size using a laser based device. Park, Shin and Ko (2007) proposed an algorithm that produces forming information iteratively by using a measuring system based on laser and CMM (Coordinate Measurement Machine) equipment. However, the measurement system is still evolving since it needs to handle various practical problems encountered in the real manufacturing process of each shipyard. So, it should be designed for each shipyard. Typically, in a large sized shipyard, it is proper to use a laser vision system or a GPS (Global Positioning System) which uses a laser or supersonic waves. These methods can provide vertex, edge and inner point information of a manufactured plate. But in medium or small sized shipyards, it is more suitable to use a contact-type measuring system such as a CMM because of cost and accuracy.

Design System

For computing thermal forming information, data representing a designed surface is required. Such design data can be extracted from a design system. However, it is almost impossible to extract 3D design data directly from the system because most data used by shipyards are provided in either 2D or 2.5D form. Therefore, appropriate definitions and procedures are needed to obtain necessary data information from various design systems in a uniform manner for the forming information computation system.

SOFTWARE ARCHITECTURE FOR COMPUTATION OF FORMING INFORMATION

In this section, a brief introduction of the ooCBD methodology is presented and the development process for a software architecture for computation of forming information is explained.

ooCBD Methodology

A component based development methodology is a method to develop a software system using software components each of which is a software module providing independent service (Jeon, 2004). From the comparison of software components with functions, classes and application packages, a relationship can be established as shown in Fig. 2 (Jeon, 2004; Katharine, 2002). In the figure the horizontal axis indicates the ratio of applications solved by using components and the left vertical axis shows the degree of flexibility. The right vertical axis is the degree of easiness of modification and cost-effectiveness. From the figure, it can be observed that routines/functions are easy to be modified and flexible but least cost-effective, whereas application packages have no flexibility but show high cost-effectiveness. Components lie near the middle of the spectrum where the component-based approach is the optimum choice in the software development in terms of cost-effectiveness and flexibility. Considering the inherent properties of forming information computation, the component-based development methodology is the most appropriate for the development since the forming information software should be flexible and interfaced with diverse subsystems and cost-effective.

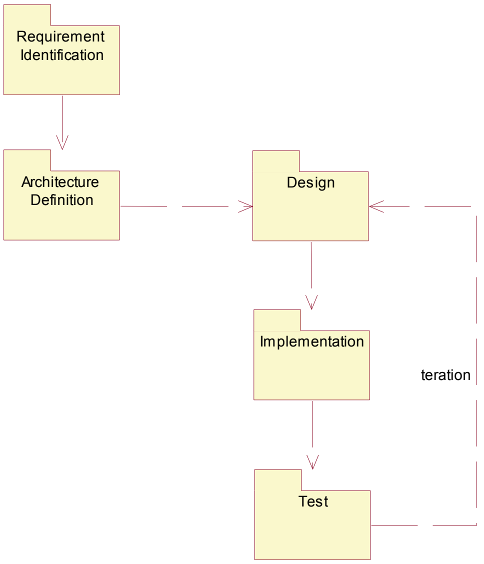


Fig. 7 The development process of the ooCBD methodology.

The development process of the ooCBD methodology consists of requirement analysis, definition of architecture, design, implementation and test. The relations of those steps are depicted in Fig. 7. Following the process iteratively, software can be developed which satisfies all requirements. The first step of the process is ‘requirement analysis.’ At this step, essential development components are identified which can determine the performance of a system. This step consists of three sub-steps: identification, definition and refinement of requirements and each sub-step is also composed of another sub-step. After analyzing requirements, a suitable architecture is defined based on the requirements. Then, the three steps, or the design, implementation and test steps, are performed iteratively to generate the final product.

If a system is developed through the development process, detailed diagrams and documents about the system architecture are obtained, which show the specifications of the system to be developed and allow it to be adaptively modified depending on the change of the requirements.

Software System for Computation of Forming Information Based on ooCBD Methodology

In this section, how a software module for computation of forming information has been developed by using the ooCBD methodology is presented.

Requirement Identification

Requirement identification consists of three activities:

1) collecting stakeholders' requirements,
2) understanding them
3) and defining terminologies used by them.

Table 1 Requirement Document.

ID	Name	Descriptions	Type	Concerned people
R001	**Management designed surface**	● Check the list of the designed surfaces of each block. ● Check the heating information about the designed surfaces. ● Select several designed surfaces at a time.	Functional	Designer
R002	**Calculate heating information**	● Calculate the heating information (heating location and recommended velocity) of the selected surfaces. ● Calculate the forming information for cold forming. ● If several surfaces are selected, calculation is done at one time. ● The computation results are saved automatically. ● The heating information should be calculated for every polygon. ● Heating information can be calculated considering initial shape ● It should be able to compute the forming information for a plate with initial curvature, a case for an iterative forming process	Functional	Designer
……	……	● ……	……	……
……	……	● ……	……	……
R013	**Definition of Surface**	● Surface data is defined by NURBS	Non-functional	

Table 1 shows a sample of a requirement document. Based on those requirements, a 'use-case' is made and realized through the requirement definition step which determines the priority of the requirements. Then the requirement identification is finished through the requirement refinement step which structuralizes the use-case.

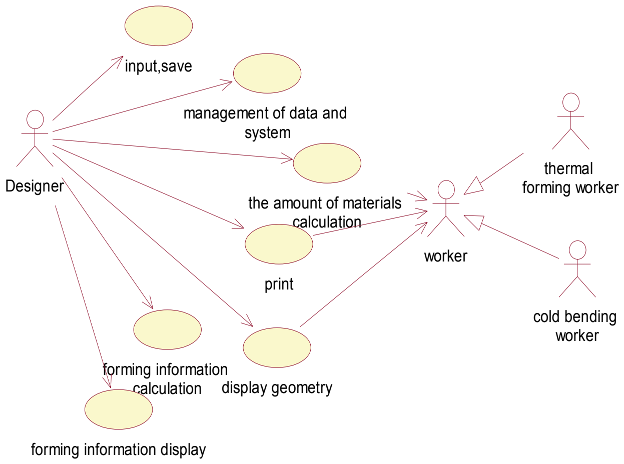

input,save
management of data and system
Designer
the amount of materials calculation
print
worker
thermal forming worker
cold bending worker
forming information calculation
display geometry
forming information display

Fig. 8 The use-case diagram of software.

Fig. 8 shows a 'use-case' diagram which has been derived from the requirement identification. Such a 'use-case' is very important in the development of software architecture in that it shows the flow of information of the forming information calculation system. In this use-case diagram, the use-cases concerning 'input', 'storage', 'data' and 'system management' are closely associated with subsystems of the overall automation system such as forming, measurement and design systems.

Architecture Definition

Software architecture is defined as "the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them" (Bass, Clements and Kazman, 1998). In the ooCBD methodology, the system structure is defined using four different types of architecture: business, application, techniques and data architecture as shown in Fig. 9.

Diagrams and documents of each architecture are generated in the ooCBD methodology during defining each architecture. Fig. 10 shows an example of a diagram, which shows the components comprising the forming information computation programs, are separated and the relations among them are defined.

Design, Implementation and Test

The architecture defined in the previous section is realized into a real software program through the iterative process of design, implementation and test. In particular the business components are implemented, and the user interfaces and interfaces between the components are designed and tested. Namely, in this iterative process, each business component is changed into a real component.

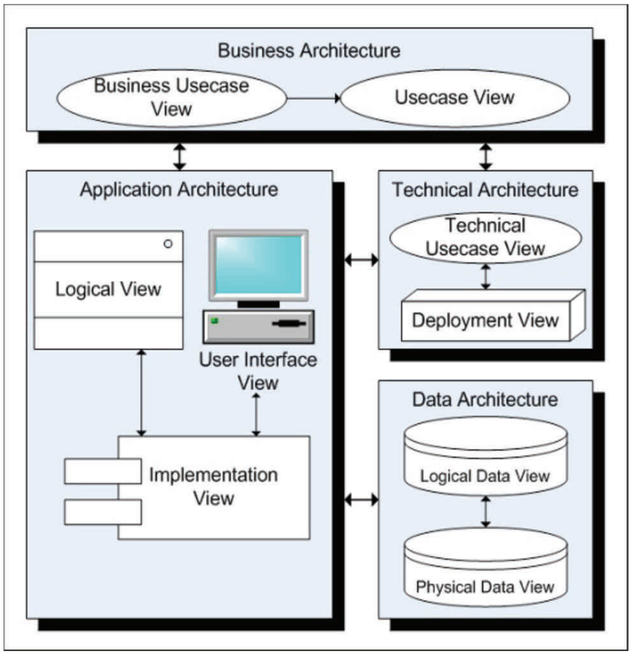


Fig. 9 The system structure defined in the ooCBD methodology (Jeon, 2004).

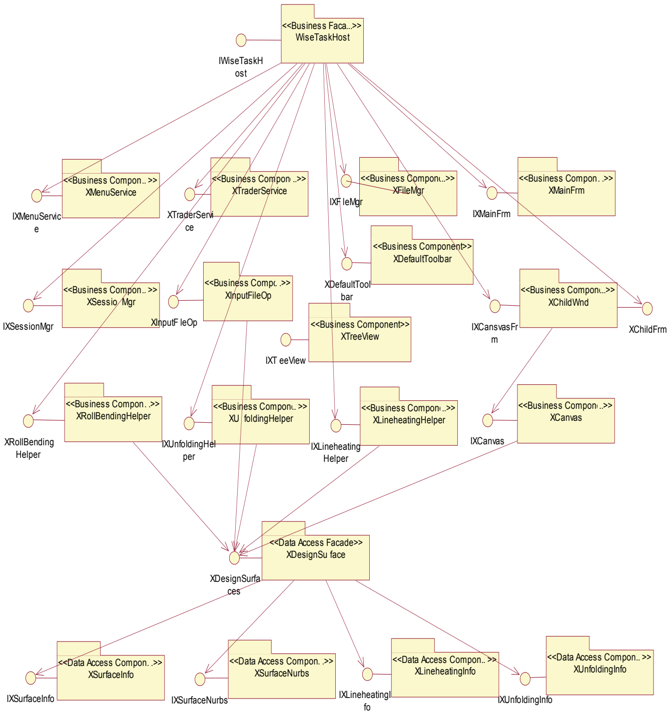

<<Business Faca..>> WiseTaskHost
IWiseTaskHost
<<Business Compon..>> XMenuService
IXMenuService
<<Business Compon..>> XTraderService
XTraderService
<<Business Compon..>> XFileMgr
IXF leMgr
<<Business Compon..>> XMainFrm
IXMainFrm
<<Business Component>> XDefaultToolbar
XDefaultToolbar
<<Business Compon..>> XSessio Mgr
IXSessionMgr
<<Business Compo..>> XInputFileOp
XInputF leOp
<<Business Compon..>> XChildWnd
IXCansvasFrm
XChildFrm
<<Business Component>> XTreeView
IXT eeView
<<Business Compon..>> XRollBendingHelper
XRollBending Helper
<<Business Compon..>> XU foldingHelper
IXUnfoldingHelper
<<Business Compon..>> XLineheatingHelper
IXLineheating Helper
<<Business Compon..>> XCanvas
IXCanvas
<<Data Access Facade>> XDesignSu face
XDesignSurfaces
<<Data Access Compon..>> XSurfaceInfo
IXSurfaceInfo
<<Data Access Compon..>> XSurfaceNurbs
IXSurfaceNurbs
<<Data Access Compon..>> XLineheatingInfo
IXLineheatingInfo
<<Data Access Compon..>> XUnfoldingInfo
IXUnfoldingInfo

Fig. 10 The business component model and relations among the components.

Fig. 11 is one example diagram drawn at the design stage, which shows a use case storyboard for requirement analysis and architecture definition. This storyboard shows a procedure through which a designer with information of a ship obtained from the design data registers the information to the forming information computation program.

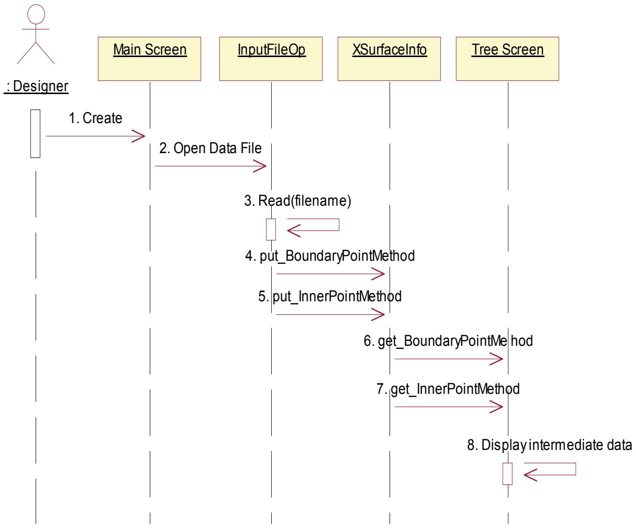
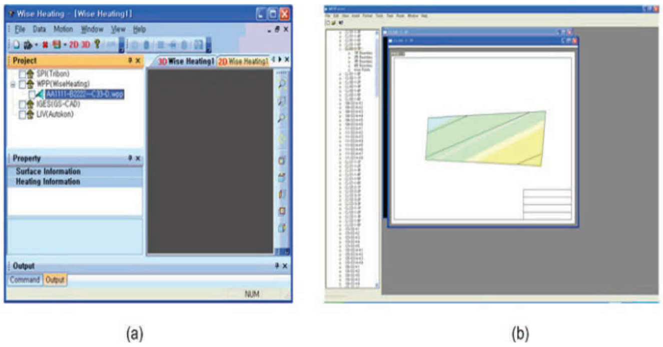


Fig. 11 The sequence diagram of a use case storyboard.

Using this storyboard, software can be implemented. Screenshots of the developed program are given in Fig. 12, where Figs. 12 (a) and (b) show different graphical user interfaces.



(a) (b)

Fig. 12 A software application developed based on the ooCBD methodology with components using different graphical user interface modules.

APPLICATIONS

In this section, what type of data the developed software exchanges with external systems and how the software should be modified when the external systems are modified along with the data changed is presented in detail.

Definition of Data for Systems

Data Definition with Design System

For obtaining thermal forming information, it is necessary to receive data from the design system. However, since each shipyard does not use the same design system and most of design systems used in the shipyards do not support 3D CAD capabilities, it is hard to extract 3D design surface data directly from those systems in a coherent manner. Therefore, to overcome this problem, a set of specifications necessary for computing forming information should be defined and reflected in the interface design.

The most critical information for forming information calculation is the 3D shape of a designed surface. So a specification for defining the shape should be provided for each design system. Once the surface is defined by the specification, then it is given as input to the forming information calculation system.

The shape information that can be obtained from the current CAD systems is point data of the designed shape. However, such shape data in point form is not enough for the forming information calculation since no surface information is available. Therefore, a method should be included which can estimate the 3D shape from those point data.

Given the 3D shape information (vertices and edges) of the designed surface, a surface can be modeled using Coon's patch modeling method (Farin, 1999). Since this method only relies on the vertices and edges of the designed surface, however, the modeled surface may not reflect the accurate shape of the designed surface and the estimated forming information may not be used in the forming process. To mitigate this problem, more accurate approximation of the designed surface is required, which can be achieved by using additional information as follows:

- 3D point data defining the inner shape of the designed surface
- the properties and the thickness of a steel plate
- the margin length and the location for forming

In general, however, it is hard to obtain dense point data representing the inner shape from the design systems used by shipyards since they do not handle 3D surface data directly. Nevertheless, most of the plates used in the shipyards hardly contain areas of high curvature. So, it is enough to approximate sparse inner data points for modeling the designed surface without dense data points. In a certain case, much effort is required to extract 3D data from a CAD system. For example, some design system provides intermediate files that have something to do with 3D design data, and through several steps to translate them, the required 3D design data can be obtained. Otherwise, 3D design data can mostly be extracted using drawings like as landing body plans, which is a tedious job (Kim, Park, Jo, Shin, Kim and Ko, 2008). Recently, some shipyard has introduced a 3D design system in the ship design. In this case design surfaces can be obtained accurately and efficiently by using the IGES (Initial Graphics Exchange Specification) format.

Data Definition with Forming Hardware System

In general 3 or 6-axis instruments can be considered for thermal forming. Although the 3-axis one is cheaper, it is hard to be used when a plate contains initial curvature since the direction of a heating torch cannot be normal to the plate in the curvature region. Most heating instruments support a point-to-point motion of a torch and information required for the forming hardware system is as follows:

- 3D starting points of heating
- 3D ending points of heating
- the torch velocities during the motion from the starting to ending points

If a 6-axis instrument equipped with a rotational motion of a torch is used, it can be possible to heat a plate perpendicularly and produce a better result. In this case, additional information is needed:

- the normal vector of a plate at the heating position

Data Definition with Measurement System

Two types of data are needed by the forming information calculation software from the measurement system: the position of a plate being measured and its 3D surface shape. Recognizing the location of the plate can be done in various ways by using any types of measurement devices. However, obtaining the accurate shape of the plate is not an easy job due to the limitation of the measurement devices, the plate size and the harsh environment where the measurement devices operate. These conditions ought to be taken into consideration for measurement automation. Therefore, it is important to determine what type of measurement instruments need to be used and how each of them should be interfaced with the forming information computation software.

For small sized shipyards, the need for iterative forming is not high in general since the automatic measurement system is not cost-effective, and relatively thinner plates are mostly used which are easily formed than thick plates. In this case only mapping information is required between the coordinates of a plate for the forming hardware system and the coordinates of forming information. On the other hand, plates with more than 40mm in thickness are frequently used in larger shipyards and in order to form such thick plates iterative forming needs to be employed. Therefore, it is necessary to develop an iterative forming system and one of the critical peripherals of the system is an automatic measurement device. In this case, not only the position information of a plate is important but also the accurate shape is very critical for the further computation. For this purpose, the shape data need to be obtained from the automatic measurement system, among which inner data points are used for the iterative forming process:

- 3D vertex data
- 3D edge data
- 3D inner point data

Reflection of Modifications

Using the ooCBD methodology, it is possible to compose a software system more efficiently through systematic requirement understanding and architecture definitions. The method also offers a guideline for maintenance of the software. So whenever there is a change for the software, it can be handled efficiently through the management of associated documents. Moreover, based on these documents and related diagrams, various versions of software can be maintained. Since components which satisfy COM+ rules are used and the software has been structured accordingly in the development, new functions or updates can be added to the software by only modifying related components and replacing the existing ones by the updated ones (Sommerville, 2001) without compiling the whole system.

EXAMPLES

In this section, two examples are provided which demonstrate how the forming information computation software developed in this work can be applied to different manufacturing processes of a medium and a large sized shipyard.

Medium Sized Shipyard

The forming information calculation software has been applied to the curved hull fabrication process of the 21C Shipbuilding Co. Ltd. This company uses ‘Tribon’ as a design system, which has been widely employed in the shipbuilding industries. Heating is performed using a 3-axis gantry type machine which was originally designed and used for cutting operation. There is no instrument for measurement. Instead, the plate is measured manually by moving the heating hardware, touching the plate with the tip of a heating torch and obtaining the position of the tip.

A couple of features specific for this shipyard are as follows: first, it uses relatively thin plates compared with larger sized shipyards. Second, iteration is not needed in the forming process since the manufacturing process has been established such that during the assembly process, an operator checks the completion of the fabricated plate and forms a desired shape through more manual heating if not completed. Since no iterative forming is used, the system cannot fabricate the desired plate accurately but this shipyard estimates that it is reasonable to form 50~70% of the desired shape through one forming cycle and complete it through the manual operation.

Under these conditions, the proposed forming information computation software has been adjusted to meet various requirements and different environments. Fig. 13 shows a fabrication process using the proposed forming information calculation software, which has been applied for the hull plate forming process in this medium sized shipyard (Park, Shin, Kim, Kim, Cho, Leem and Ko, 2008). In the figure, the steps in the thick rectangles indicate the software module, which receives data from the CAD system, ‘Tribon’, and generates necessary heating information for forming, which is then used by an NC machine to heat a plate.

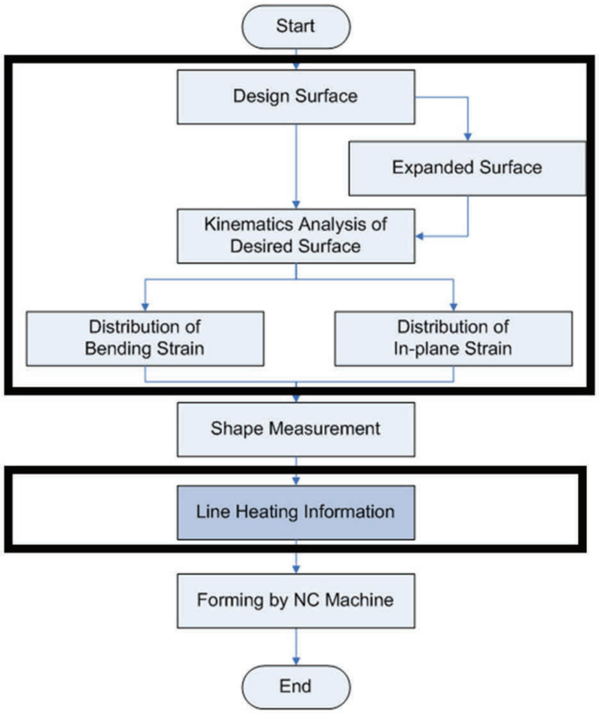


Fig. 13 A curved plate forming process for a medium sized shipyard.

In this workflow, the shape measurement process is not for accurate measurement but for estimation of rough data of vertices and edges of a plate for mapping. Fig. 14 shows the computed heating information for thermal forming and Fig. 15 shows the actual heating operation using this information.

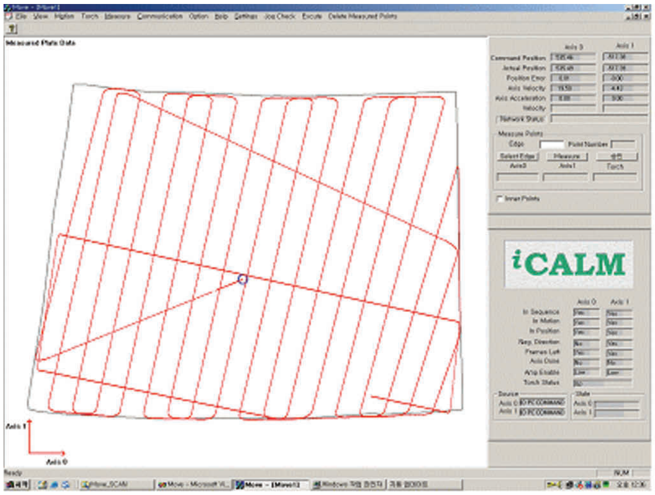


Fig. 14 A screenshot of heating information.

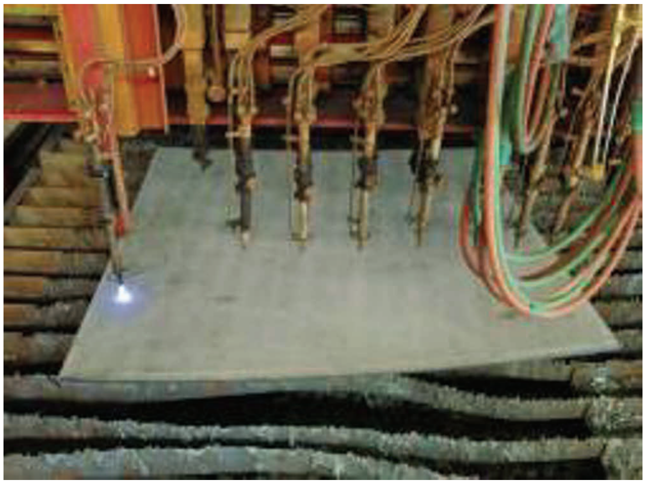


Fig. 15 A heating operation using a gas torch as a heat source.

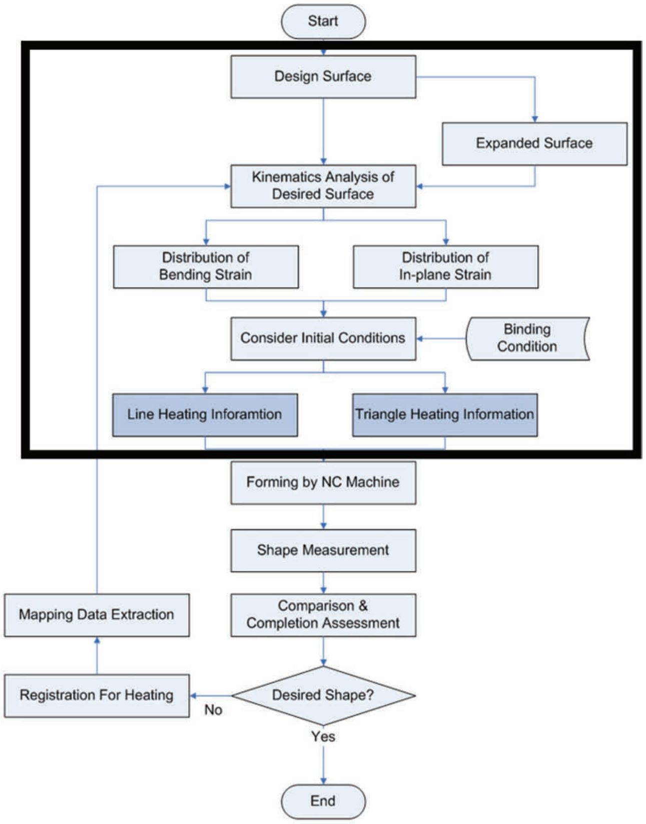

Start
Design Surface
Expanded Surface
Kinematics Analysis of Desired Surface
Distribution of Bending Strain
Distribution of In-plane Strain
Consider Initial Conditions
Binding Condition
Line Heating Inforamtion
Triangle Heating Information
Forming by NC Machine
Shape Measurement
Mapping Data Extraction
Comparison & Completion Assessment
Registration For Heating
Desired Shape?
No
Yes
End

Fig. 16 Iterative plate forming process for a large sized shipyard.

Large Sized Shipyard

The proposed software has been applied to a large sized shipyard in Korea. The shipyard uses both 'Autokon' and 'GS-CAD' as its design system. The forming hardware system is of gantry type with 6-axis devices. For measurement a laser and a CMM are used. The plate forming workflow, which is applied to this shipyard, is shown in Fig. 16 (Park, Shin, Ko, Hyun and Doh, 2007). In the figure, the thick box constitutes the forming information computation part.

CONCLUSIONS

In this study, a software unit for an automation system for the hull plate forming process is developed by applying the plug-in concept. The automation system can be divided into four parts: a forming hardware system, a measurement system, a design system and a forming information calculating software system. Except the software system, the types of the subsystems are different from shipyard to shipyard. So, the plug-in system concept is introduced for the development of the forming information calculating system.

In order to meet the various requirements between the components imposed from constructing an automatic forming system, the software for forming information calculation is developed using the ooCBD development methodology. The software can be maintained efficiently through documents and diagrams which are derived during the development procedure. Using this methodology, modification or elimination of functions or modules can be handled easily through various components satisfying the COM+ rules.

From this work it is shown that a plug-in type system can be adaptively employed in various environments. In particular for the construction of an automatic curved hull forming system, developing a software module based on the plug-in concept is very critical and such a module can be effectively constructed by using components. The applicability of the developed system is demonstrated through examples by applying it for a medium and a large sized shipyard successfully.

ACKNOWLEDGEMENTS

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the National Research Lab. Program funded by the Ministry of Science and Technology (No. M10300000213-05J0000-21310).

REFERENCES

Allen, P. and Frost, S., 1998. *Component-based development for enterprise systems*. New York: Cambridge Univ. Pr.

Bass, L. Clements, P. and Kazman, R., 1998. *Software architecture in practice.* 2nd ed. Massachusetts: Addison-Wesley.

Dolenc, M., 2004. *Developing extendible component-oriented finite element software*. 35, pp. 703-714.

Farin, G., 1999. *NURBS from projective geometry to practical use.* 2nd ed. Massachusetts: A K Peters. Ltd.

Jang, C.D. Kim, H.K. and Ha, Y.S., 2002. Prediction of Plate Bending by High-Frequency Induction Heating. *Journal of Ship Production*, 18(4), pp. 226-236.

Jeon, B.S., 2004. *.NET Enterprise System ooCBD Development Methodology.* Seoul, Korea: Young Jin Dot Com Co.

Katharine, W., 2002. *Component-based development principles and planning for business systems*. Massachusetts: Addison-Wesley.

Kim, J. Park, J.S. Jo, Y. Shin, J.G. Kim, W.D. and KO, K.H., 2008. A study of 3D design data extraction for thermal forming information. *Journal of Ship and Ocean Technology*, 12(3), pp. 1-13.

Kruchten, P., 2003. *The rational unified process an introduction.* Massachusetts: Addison-Wesley.

Park, J.S. Shin, J.G. Kim, Y.B. Kim, H.K. Cho, S.A. Leem, H.K. and Ko, K.H., 2008. Application of heating line computation model in a shipyard: A case study. *Journal of Ship Production*, 24(1), pp. 17-24.

Park, J.S. Shin. J.G. and Ko, K.H., 2007. Geometric assessment for fabrication of large hull pieces in shipbuilding. *Computer-Aided Design*, 39(10), pp. 870-881.

Park, J.S. Shin, J.G. Ko, K.H. Hyun, C.M. and Doh, Y.C., 2007. Development of an automated line heating information extract system for fabrication of curved hull plates. In: *International Conference on Computer Applications in Shipbuilding.* Portsmouth, UK 18 – 20 September 2007.

Qureshi, M.R.J. and Hussain, S.A., 2008. A reusable software component-based development process model. *Advances in engineering software*, 39, pp. 88-94.

Rogerson, D., 1997. *Inside COM. Washington*: Microsoft Press.

Shin, J.G. Lee, J.M. and Nam, J.H., 2004. An efficient algorithm for measurement and comparison of large-scale hull pieces in the line-heating process. *Journal of Ship Production*, 20(1), pp. 60-67.

Shin, J.G. and Ryu, C.H., 2000. Nonlinear kinematic analysis of the deformation of plates for ship hull fabrication. *Journal of Ship Research*, 44(4), pp.270-277.

Shin, J.G. Ryu, C.H. and Nam, J.H., 2004. A comprehensive line-heating algorithm for automatic formation of curved shell plates. *Journal of Ship Production*, 20(2), pp. 69-78.

Sommerville, I., 2001. *Software engineering*, Massachusetts: Addison Wesley.