論 文

大韓造船學會論文集
第 31 卷 第 3 號 1994年 8月
Transactions of the Society of
Naval Architects of Korea
Vol. 31, No. 3, August 1994

객체지향 선체모델링을 위한 모델의 자료구조 및 내부처리 기법에 관한 연구

서승완*, 강원수*, 이규열*, 이규옥*

A Study on the Data Structure and Implementation Techniques for the Object Oriented Ship Structure Modeling

by

Seung-Wan Suh*, Won-Soo Kang*, Kyu-Yeul Lee* and Kyu-Ock Lee*

요 약

CSDP(Computerized Ship Design and Production) 연구사업에서는 선체설계업무를 지원하는 "선체CAD시스템"개발을 수행하였다. 본 시스템 개발을 위해서 시스템 요구분석을 수행하여 현재의 선체설계에 관련된 설계업무의 내용과 흐름을 분석하였다. 이러한 요구분석 결과를 토대로 선체CAD 시스템을 이용할 때 설계단계마다 요구되는 설계 정보를 생성하기 위한 전산화된 선체 모델을 정의하였다. 선체모델의 표현과 구현을 하기 위해서는, 형상정보 뿐만 아닌 설계개념까지도 표현할 수 있는 제품모델의 구현을 위해서 객체지향기법을 적용하므로써 실제의 선체설계업무를 전산화된 선체모델에 반영할 수 있는 모델의 자료구조와 컴퓨터 내에서의 처리기법을 제시하였다.

Abstract

The development of "Ship structural CAD system" has been performed in CSDP(Computerized Ship Disign and Production) project to support ship structural design works. The procedures and contents of current design practice concerning the ship structural design were analyzed.

To generate design information required at each design stage using the ship structural CAD system,the computerized model was created. To represent and implement the model information, a data structure which represents design concept as well as shape information was proposed using the concept of the product model. To embody the computerized model, the object-oriented techniques were established to represent requirements of real world.

발표일자 : 1993년도 대한조선학회 추계연구발표회('93.11.13)
접수일자 : 1993년 1월 31일, 재접수일자: 1994년 6월 10일
* 정회원, 선박해양공학연구센터

1. 서 론

기존의 조선용 CAD/CAM 시스템이 갖고 있는 문제점을 극복하는 것이 조선 CIM을 성취할 수 있는 출발점이라는 인식하에 , 기존의 조선용 CAD/CAM 시스템들이 갖고 있는 문제점을 제품 정보처리와 제품모델링 측면에서 분석한 결과, 첫째로 종래의 시스템에서는 응용프로그램, 전산모델, 데이타베이스라는 세가지 시스템 구성요소가 정보처리 경계와 역활의 구분이 명확하지 않음을 알게되었다. 이로인해 응용프로그램 중에 모델링 기능이 산재하게되고 각각의 응용프로그램은 설계단계에 따라 별도로 적용되고, 더우기 설계단계의 세부작업 단위에 따른 기본설계용,상세설계용,생산설계용 등의 별개의 응용프로그램에 의해서 설계가 진행될 수 밖에 없었으므로 이들 응용프로그램 사이에는 데이타 전달 및 데이타 변환이 요구되고 이에 따른 많은 정보전달의 비효율성과 정보의 흐름이 원활치 못했다[2].

둘째로, 전산모델과 데이타베이스에서 취급되는 정보는 형상 중심의 기하학적 속성만을 모델링하므로써 설계과정을 표형하는 의미론적(Semantics)이고 동적인 설계개념을 정보로서 표현하는 것이 불가능한 상태이었다.

세번째 문제점으로는, 종래의 조선용 CAD/CAM 시스템의 개발 개념이 절차식 방법에 기초를 두고 있으므로, 설계 및 생산에서 요구되는 사항을 적절히 표현 할 수 없었다는 문제점을 갖고 있었다.

선박의 제품모델링 기술에 관련된 연구동향을 살표보면, 일본에서는 조선 CIMS(Computer Integrated Manufacturing System)개발의 일환으로 "조선 CIM을 위한 설계 생산정보 획득 지원 시스템"인 SODAS(System of Design and Assembling for Shipbuilding)시스템을 개발하여 이를 선체 생산 정보를 설계초기 단계에서 지원하는 기능을 갖는 Prototype 시스템을 제시하고 있다[10].

SODAS 시스템에서는 선박의 초기 구조모델로 부터 블록분할과 개략 물량정보의 산출을 가능케 하는 시스템 기능에 촛점이 맞추어져 있으나, 선박 제품모델의 구축을 위한 자료 구조로는 아직은 미흡한 상태이다.

Bronsart[9] 논문에서는 STEP (Standard for the Exchange of Product Model Data)의 "Part 102:Ship Structure"[11]에 대한 고찰과 함께 STEP에서 제안한 모델 기술 언어인 EXPRESS를 사용해서 선체구조를 구성하는 객체에 대한 표현 방법에 대한 기초적 개념을 제시하고 있다. 본 논문의 선행연구인 "객체 지향 선박 구획정의 표현방법론"[8]에서는 선박 구획을 대상으로 객체 지향기술을 토대로 한 선박 구획 배치모델의 표현 방법론을 정의하였으며, 본 논문에서는 앞선 연구결과를 확장하여, 선체구조 설계 정보로부터 생산부품 정보까지 망라한 선체모델로서 정합성을 유지하면서 완전히 표현될 수 있도록 하였다.

본 논문에서는 모델정보의 표현과 처리라는 측면에서는, 제품모델의 개념을 도입하여 제품의 형상정보만이 아닌 설계개념과 정의과정 등을 표현할 수 있는 자료구조를 제시하였고, 전산화된 모델의 구현이라는 측면에서는, 객체지향기술을 기초로 하여 현실 세계의 요구사항을 전산화된 모델로 표현하는 기법을 정립하였다. 또한 시스템 개발이라는 측면에서는 객체지향 모델링과 객체지향 프로그래밍 기술을 도입하므로써 개발된 시스템의 확장성과 효율성을 최대한 유지할 수 있도록 하였다[1]

2. 제품모델의 기본개념

선체 기본설계 단계와 선체 상세설계 단계시 요구되는 광범위한 모델 정보와 물량정보를 컴퓨터내에 정확히 표현하기 위해서는, 제품모델 (Product Model)개념에 기초를 둔 선체 모델을 구현해야 하며, 이를 구체화하기 위한 자료구조가 확립 되어야 한다. 또한 이 모델은 선박 설계 단계뿐만 아니라 생산단계에 있어서의 모델로도 사용되어야만 조선 CIM을 실현화 할 수 있는 기초가 되므로, 선박의 기능정보, 제품정의정보, 생산공정정보, 생산관리정보를 총 망라한 광범위한 제품정보가 포함되어 효율적으로 전산모델로 구축되어야 한다.

제품모델이 포함하여야 할 포괄적인 정보구조인 기능정보, 제품정의 정보, 생산공정정보, 생산관리 정보를 도식화 하면 Fig.1과 같으며, 본 논문에서는 첫단계로써, 선체 기본설계가 완료된 시점에 이로부터 정보를 받아 정확한 표현을 할 수 있도록 선박제품모델의 구조를 제시하고 이들 자료구조의 정합성과 확장성을 검증하였다.

3. 자료 구조의 상세화

3.1 개요

개발하고자 하는 선체 CAD시스템에 대한 기능 분석 결과를 [5][6][7] 토대로 선체 모델을 구축하기 위한 자료구조를 객체지향 모델링 기술을 이용하여 정의하

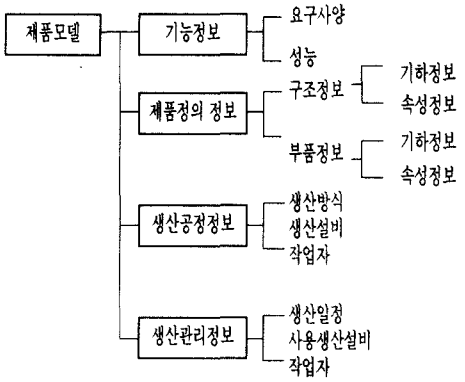

Fig. 1 Information components of product model

였다. 객체지향 모델링 기술은 모델링 대상이 되는 문제 영역을 전산 시스템화하기 위한 한 가지 방법으로서 문제 영역내에 포함되어 있는 객체요소들을 도출하고, 그 객체들이 어떤 상관 관계를 유지하고 있는지를 정의하므로써 시스템을 모델링하는 방식이다. 즉, 객체라는 정보 단위를 사용하여 시스템을 모델링하는 방식으로서 시스템 설계자 입장에서는 문제 영역으로 부터 객체를 도출하고, 그 객체들간의 상관 관계를 정의해 주는 것이 핵심 기술이다. 객체는 현실 세계에 존재하는 개념적 개체로서, 예를 들면 상갑판면, No.1 cargo hold 등으로 자기 자신의 고유한 성질을 표현하기 위한 데이타 구조(이하 "멤버데이타"라 함)와 데이타를 처리하는 기능(이하 "멤버함수"라 함)을 함께 갖는 정보단위로서, 객체를 정의하는 것이 결국 해당 시스템의 자료구조를 결정하는 것이 된다. 따라서, 본 논문에서는 이중 선각구조 대형유조선의 중앙 화물창부에 대한 선체모델을 구축하기 위해 필요한 객체를 도출하고, 이들 객체들간의 상관관계를 정의한 후, 각각의 객체에 대한 멤버데이타와 멤버함수를 결정하였다.

3.2 선체 모델 구축을 위한 객체의 도출

선체 구조는 부재의 접합에 의해서 구조가 완성되지만, 설계자는 초기 단계에서부터 하나 하나의 부재를 고려하면서 설계하지는 않는다. 즉, 초기 단계에서는 이중저의 폭/높이, floor의 수와 크기, bilge hopper tank의 형상 등 형상 구조에 관한 설계정보 중심으로 설계가 진행되는 반면, 세부 부재의 치수가 어느 정도 되어야 하는지는 상세 설계를 통해 결정된다. 이와같이 각 설계단계에 따라 설계대상이 다르게 되는데 구조의 기본설계에서 상세/생산 설계로 진행되는 설계흐름을 자연스럽게 표현할 수 있는 선체모델의 구축을 위해서는 우선 실제 부품으로서 존재하지 않는 개념적인 부품의 표현을 고려한 모델을 정의해야 한다. 이를 위해 "CSDP-선체 CAD 시스템 개발" [4]이라는 연구과제에서는 일본 조선 CIMS의 연구결과의 일부인 선체유닛 모델 개념[10]을 자료구조의 정의를 통해서 구체화 하였으며, 이는 사용자가 모델링 작업을 효율적으로 수행할 수 있도록 정의한 유닛 개념을 토대로 실제 선체모델 내부에서 계층적인 구조를 가지면서 정의될 객체들을 정의하였다. 정의된 객체를 그룹별로 보면 크게 네가지로 분류할 수 있는데, 이는 유닛 개념에 대응되는 것으로서 다음과 같다.

Level 1 : 대상 선박 유닛 : Ship 객체
Level 2 : 구획 배치 유닛 : DCU(Design Compartment Unit)객체
Level 3 : Tank 유닛:Tank객체
Level 4 : 구조 부재 유닛 : Plate, Stiffening_Member 객체

Fig.2는 유닛개념을 고려한 선체 모델을 구축하기 위해 필요한 세분화된 객체 및 객체들간의 관계를 모델링한 객체 모형도로서, 선체 모델의 계층적 자료구조는 Ship,Design Compartment Unit(DCU), Tank, Plate 및 Stiffening_Member 객체로 구성 하였으며, 이들 객체는 자기자신의 고유한 성질을 나타내는 속성정보와 geometry 정보 및 topology 정보에 의해서 표현되도록 하였다.

Ship 객체는 모델링 대상 선박의 선종, 주요요목 등을 보관하는 객체로서, 주요 수밀횡격벽을 정의하고 이들로 부터 생성되는 공간을 DCU 객체로 정의하는 기능을 수행한다. DCU 객체는 선박을 위치별로 크게 선미부, 기관실부, 화물창부 및 선수부 구역으로 나누어 이들 각각에 대한 모델링 결과를 보관하기 위해 정의한 객체이다. Tank객체는 용적, 중량 중심, 용도 등과 같은 각 tank의 특성과 해당 tank내에 포함될 보강판 및 보강재를 보관하기 위한 객체이다.

Plate 객체는 선체 모델을 구성하는 판 부재들을 정의하기 위한 객체로서 판의 특성에 따라 주판 (Main Plate)과 보강판(Stiffening Plate)으로 구별하여 각각을 객체로 정의하였다. 주판 객체는 갑판, 격벽류와 같은 내판(Inner Plate)과 선체 형상을 표현하는 외판(Shell) 객체로 다시 분류하였고 보강판 객체는 Web Plate, Ring Plate 및 Large Bracket으로 분류하였고 이들을 각각 표현하기 위한 객체를 정의하였다. 또

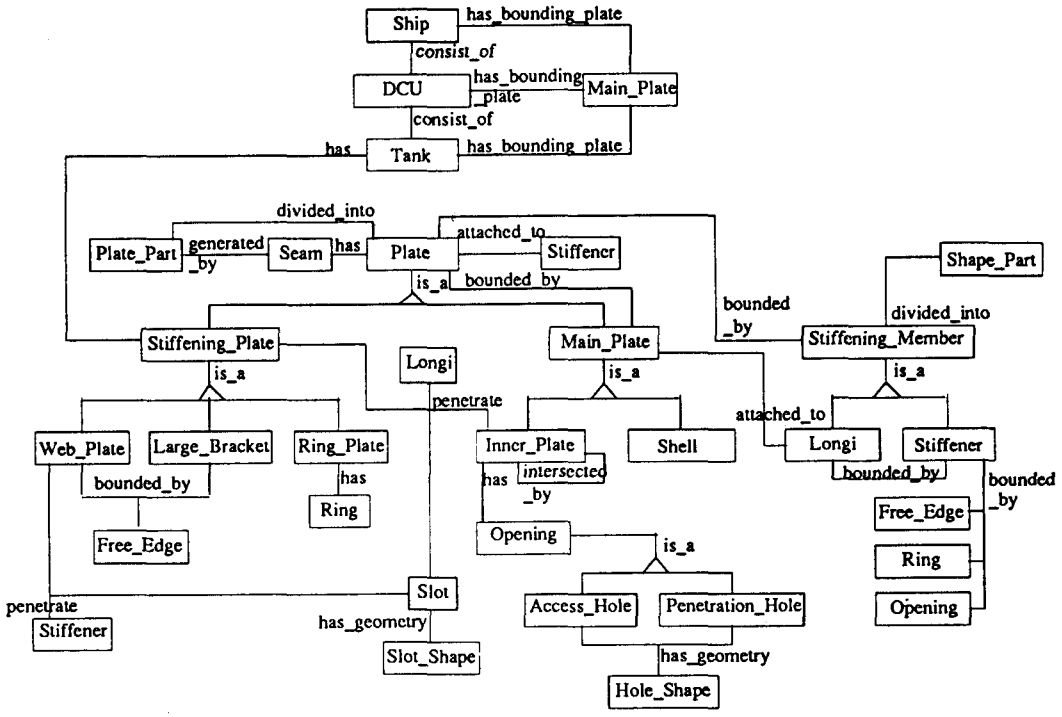

Fig. 2 Object diagram for structural element

한, Plate 객체와는 별도로 보강재(Stiffening Member)객체를 정의하여 보강재에 대한 정보를 표현할 수 있도록 하였는데 종통재와 방요재 등과 같은 보강재를 표현하기 위해 Longi와 Stiffener 객체를 정의하였다.

이상과 같이 선체 모델을 구축하기 위해 필요한 객체들을 도출한 후, 이들 객체들간의 상호 관련성(topology relationship)을 포현하기 위해 다음과 같은 상관 관계를 정의하였다.

-Bounded_by 관계 :

임의의 Plate 객체가 다른 Plate 객체의 경계면을 제한하고 있음을 나타내는 관계를 표현한다. 예를 들어, 임의의 종격벽을 정의할 때, 종격벽의 기본 형상을 평면상에 존재하는 무한 평판으로 정의한 후, 이 무한 평판을 상갑판면과 shell을 경계면으로 제한하므로서 종격벽의 실제 최종 형상을 구할 수 있다. 이때 상갑판면과 선체는 종격벽의 Boundary Plate라고 하며 이들은 서로 Boundary 관계에 있다고 정의한다.

-Intersected_by 관계 :

임의의 Plate 객체가 다른 Plate 객체를 절단하면서 지나가는 관계를 표현한다. 이때 절단되는 Plate 객체는 두개의 Plate 객체로 나누어진다.

-Attached_to 관계 :

임의의 보강재 객체가 어떤 Plate객체에 취부되어 있음을 나타내는 관계를 표현한다. 예를 들면 상갑판 종통재는 상갑판면에 취부되는데 이때 상갑판 종통재와 상갑판면은 attached_to 관계에 있다고 한다.

-Has_geometry 관계 :

부재 특성을 표현하는 객체와 형상정보를 표현하는 객체를 연관 지어주는 관계를 표현한다.

-Penetrate 관계 :

Longi, stiffener등의 보강재 객체가 Plate 객체를 관통해서 지나가는 관계를 표현한다. 이 관계에 의해서 Plate 객체는 slot을 갖게 되고 slot의 특성은 slot 객체에 보관된다.

-Divided_into 관계 :

Plate 객체 혹은 보강재 객체가 seam, butt등에 의해 부품화됨을 나타내는 관계로서, 이 관계에 의해 Plate 객체는 여러개의 Plate_Part객체로, Stiffen-

ing_Member는 여러개의 Shape_Part객체로 나누어진다.

-Is_a 관계 :

두 객체간의 속성 계승 관계를 나타낸다. 예를 들면 Fig. 2에 Plate 객체와 Main_Plate가 is_a관계로 연결되어 있는데, 이는 Main_Plate객체는 Plate 객체에서 파생된 객체로서 Plate 객체가 갖는 속성 정보를 계승받아 정의된다는 것을 표현한다.

-Has_bounding_plate 관계 :

Ship, DCU 및 Tank 객체와 같이 폐위된 공간으로 표현되는 객체와 경계면으로 사용될 Plate 객체와의 관계를 나타낸다. 예를 들어, No.1 cargo hold라는 Tank 객체가 상갑판면,외판, 이중저판 및 횡격벽등과 같은 Plate 객체를 경계면으로 사용하여 정의된다고 하면, No.1 cargo hold는 상갑판면등의 Plate 객체와 has_bounding_plate 관계에 있다고 말한다.

-Consist_of 관계 :

두 객체간의 집합 관계를 나타낸다. 예를 들면, Ship 객체는 선미부, 기관실부, 화물창부 및 선수부 DCU 객체들의 집합으로 정의된다. 이때 Ship 객체는 여러개의 DCU 객체로 구성된다고 말하고, 이러한 관계를 표현하기 위해 consist_of 가 사용된다.

-Has관계 :

임의의 객체가 다른 객체를 속성정보로 갖고 있음을 나타낸다. 예를 들어, 임의의 Tank 객체에는 보강판 객체들이 포함되는데, 이때 Tank 객체와 보강 판객체는 has 관계로 표현된다.

이상에서 설명한 객체들간의 상관 관계를 고려하여 각 객체들의 멤버데이타와 멤버함수를 정의하였다. 한편, 각 객체들의 형상 정보는 geometry 관련 정보를 별도로 관리하는 객체를 정의하여, 해당 객체들과 has_geometry 관계를 통해 정의되도록 하였다.

Fig.3은 앞서 정의한 객체들의 형상정보를 표현하기 위해 정의한 geometry 관련 형상 객체 및 그들간의 상관 관계를 모델링한 객체 모형도이다.

Surface 객체는 Plate 객체의 형상을 정의하기 위한 객체로서 형상에 따라 bulkhead류와 같은 Inner_Plate 객체의 형상을 정의하기 위한 Planar Surface객체와 deck, 선체형상등과 같은 3차원 곡면 형상을 정의하기 위한 Sculptured_surface 객체로 구분하였다. Planar_Surface 객체는 해당 부재 객체의 무한 평판 형상 정보와 다른 부재 객체와의 boundary 관계에 의해서 결정되는 edge 정보가 저장되도록 자료구조를 정의 하였다.

Moulded_Curve 객체는 객체들간의 상관 관계에 의해 생성되는 형상 정보를 관리하는 객체로서 상관 관계의 종류에 따라 Boundary_Curve, Trace_Curve,

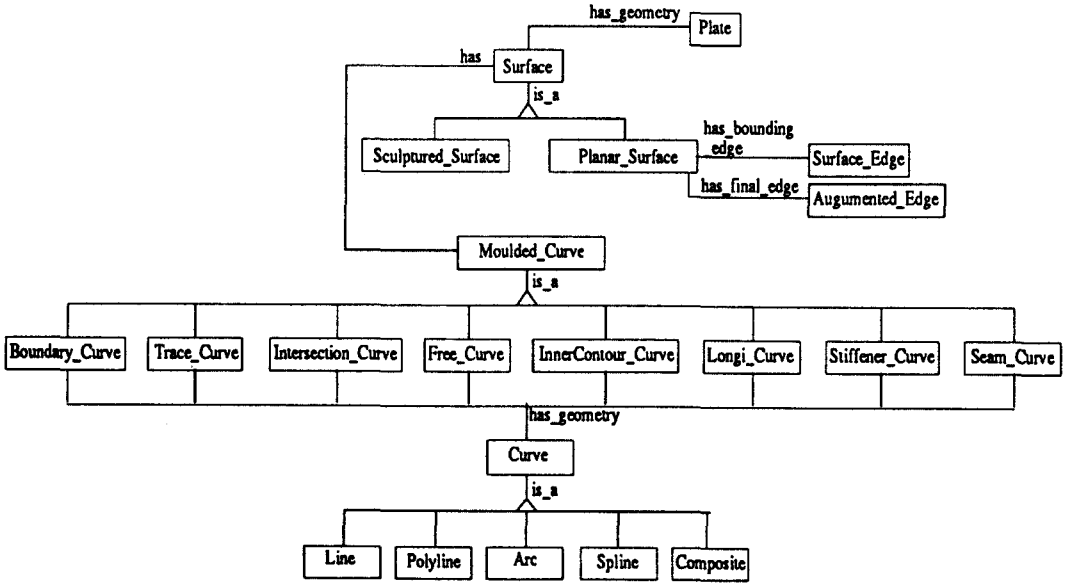

Fig. 3 Object diagram for geometry information

Table 1 An example fo class defintion ("mainPlate Class")

```
class MainPlate: Plate{
LinkedList   boundingPlates;
LinkedList   attachedLongis;
LinkedList   boundingTanks;
LinkedList   boundingDCU;
LinkedList   dividedTanks;
public:
                    MainPlate();
"
int     findBoundingPlate(char* argPlateID);
int     setBoundingPlate(char* argPlateID);
int     removeBoundingPlate(char* argPlateID);
void    findAllBoundingPlates();
void    listBoundingPlates();
"
int     findAttachedLongi(char* argLongiID);
int     setAttachedLongi(char* argLongiID);
int     removeAttachedLongi(char* argLongiID);
void    removeAllAttachedLongis();
void    listAttachedLongis();
"
int     findBoundingTank(char* argTankID);
int     setBoundingTank(char* argTankID);
int     removeBoundingTank(char* argTankID);
void    removeAllBoundingTanks();
void    listBoundingTanks();
"
int     findDividedTank(char* argTankD);
int     setDividedTank(char* argTankD);
int     removeDividedTank(char* argTankD);
void    removeAllDividedTank();
void    listDividedTank();
void    showBoundingPlate(char* argPlateID);
void    showAllBoundingPlates();
void    drawTraceCurve(char* argPlateID);
void    drawAllTraceCurves();
void    showLongi(char* argLongiID);
void    showAllLongis();
void    drawLongiCurve(char* argLongiID);
void    drawAllLongiCurves();
void    showDividedTanks();
void    showBoundingTanks();
void    showBoundingDCU();
"
void    eraseBoundingPlate(char* argPlateID);
void    eraseAllBoundingPlates();
void    eraseTraceCurve(char* argPlateID);
void    eraseAllTraceCurves();
void    eraseLongi(char* argLongiID);
void    eraseAllLongis();
void    eraseLongiCurve(char* argLongiID);
void    eraseAllLongiCurves();
void    eraseDividedTanks();
void    eraseBoundingTanks();
void    eraseBoundingDCU();
```

Interection_Curve, Free_Curve 객체등을 별도로 정의하였다. 그리고, 이들 각 curve들의 실질적인 형상정보(좌표값들)들은 Curve 객체를 통해 정의되도록 자료구조를 결정하였다.

3.3 객체의 멤버 데이타 정의

선체 모델을 구축하기 위해 필요한 객체들을 도출한 후, 이들 각 객체들의 멤버데이타를 정의 하였는데 각 개체들이 갖는 멤버데이타 중에서 주요한 것으로 Plate 객체와 MainPlate 객체를 살펴보면 다음과 같다(Table 1 참조).

■ Plate 객체

-PlateID : 해당 Plate 객체의 이름

-Surface ID:해당 Plate 객체의 형상정보를 관리하고 있는 Surface 객체의 이름

-boundaryPlates: 해당 Plate 객체의 경계면을 결정짓는 Plate 객체들의 이름으로서 연결 리스트로 정의된다.

-boundingLongis: 해당 Plate 객체가 Longi 객체의 경계면으로 사용될 경우 해당 Longi 객체들을 정의하기 위해 Longi 객체의 이름을 갖는 멤버데이타로서 연결 리스트로 정의된다.

-boundingStiffeners: 해당 Plate 객체가 Stiffener 객체의 경계면으로 사용될 경우 해당 Stiffener 객체들을 정의하기 위해 Stiffener 객체의 이름을 갖는 멤버데이타로서 연결 리스트로 정의된다.

-attachedStiffeners : 해당 Plate 객체에 취부되어 있는 Stiffener객체들의 이름으로서 연결리스트로 정의된다.

-seamCurves : 해당 Plate 객체의 seam 정보를 보관하고 있는 seam_Curve 객체들의 이름으로서 연결 리스트로 정의된다.

-dividedIntoPlateParts : Plate는 seam 정보에 의해 여러개의 PlatePart 객체로 나누어지는데 해당 Plate로 부터 생성된 PlatePart 객체의 이름을 보관한다.

■ Main_Plate 객체

Main_Plate 객체는 앞서 설명한 Plate 객체를 정의하기 위해 필요한 멤버 데이타 및 다음과 같은 추가적인 멤버데이타에 의해 정의된다. 자료 구조를 정의

하는 측면에서는 Main_Plate 객체가 Plate 객체의 파생 객체임을 선언하므로서 객체지향 개념에서 제공하는 상속 개념에 의해 Plate 객체가 갖는 멤버 데이타 자료구조를 Main_Plate 객체에서 다시 정의하지 않더라도 사용할 수 있다.

-boundingPlates : 해당 Main_Plate 객체를 경계면으로 사용하는 Main_Plate 객체들의 이름으로서 연결 리스트로 정의된다.

-attachedLongis : 해당 Main_Plate 객체에 취부되어 있는 Longi 객체들의 이름으로서 연결 리스트로 정의된다.

-boundingTanks : 해당 Main_Plate 객체를 경계면으로 사용하여 정의된 Tank 객체들의 이름으로서 연결 리스트로 정의된다.

-boundingDCU : 해당 Main_Plate 객체를 경계면으로 사용하여 정의된 DCU 객체의 이름으로서 연결 리스트로 정의된다.

-dividedTanks : 해당 Main_Plate 객체에 의해 나누어진 Tank 객체들의 이름으로서 연결 리스트로 정의된다.

3.4 클래스의 정의

앞서 정의된 객체들을 이용하여 선체 모델을 구축하기 위한 시스템을 전산화하기 위해서는 해당 객체들을 프로그래밍 언어로 정의할 수 있어야 한다. 그러나, 일반적인 프로그래밍 언어로서는 이들 객체를 표현 할 수 없기 때문에 객체지향 프로그래밍 언어를 사용하였다. 객체지향 언어에서는 객체의 표현을 위해 클래스라는 자료형(data type)을 제공한다. 클래스라 함은 임의의 객체를 표현하는데 필요한 멤버 데이타와 해당 객체의 데이타를 처리하기 위한 멤버 함수를 하나의 정보 단위로 정의해서 사용하기 위한 사용자 정의 자료형(user-defined data type)이다.

선체 CAD 시스템 구축을 위해 앞서 정의한 각 객체들의 멤버데이타와 함께 필요한 멤버함수들을 정의한 후, Table 1에 나타나 있는 바와 같이 클래스들을 정의하였다. 각 클래스에 정의되어 있는 멤버함수 중에서 Plate 클래스에 정의되어 있는 주요한 멤버함수들의 기능을 살펴보면 다음과 같다.

-defineBoundaryPlate(char* argPlateID)

임의의 Plate 객체와의 boundary 관계를 설정하는 기능을 수행한다. 함수인자인 argPlateID는 boundary 관계를 갖게 되는 Plate 객체 이름으로서, 사용자가 입력한 값이다. 이 값은 boundaryPlates 멤버데이타에 저장된다.

-setBoundingLongi(char* argLongiID)

임의의 Plate 객체와 argLongiID라는 Longi 객체간의 boundary관계를 설정하는 기능을 수행한다. 즉, 함수인자인 argLongiID를 해당 Plate 객체의 boundingLongis 멤버데이타에 지정한다.

-setAttachedStiffener(chan*argstiffenerID)

임의의 Plate 객체에 argStiffenerID라는 Stiffener 객체가 취부되어 있음을 정의하는 기능을 수행한다. 즉, 함수인자인 argStiffenerID를 해당 Plate 객체의 attachedstiffener 멤버데이타에 지정한다.

-divideIntoPlateParts$

임의의 Plate 객체가 갖고 있는 seam 정보를 이용해서 해당 plate를 분할하여 PlatePart 객체로 정의한 후, 그 각각의 객체 이름을 dividedIntoPlateParts 멤버 데이타에 지정한다. 이때 각 PlatePart 객체의 이름은 프로그램 내부에서 결정된다.

4. 선체모델링을 위한 내부처리 기법

4.1 개요

선체 CAD 시스템과 같이 복잡도가 높은 시스템의 구현을 위해서는 자료구조의 상세화 과정과 GUI 설계 과정에 병행해서 대상정보의 동적인 변화 과정을 전산기 내에서 알고리즘 형태로 어떻게 표현하여야 하는가를 사전에 정확히 검증해 보는 것이 중요하다.

본 연구에서는 시스템 개발 방법론으로서 객체지향 시스템을 추구하고 있으므로, 대상응용 분야인 선체모델에 대한 정적인 정보의 모델링을 OMT(Object Modeling Technique) 방법론에 따라 수행하였다[1]. 다음으로 동적인 정보의 모델링은 내부처리 기법이라는 알고리즘 형태로 정의하였다.

선체 CAD 시스템은 유닛이라는 개념을 도입하여 파라메트릭 설계가 가능한 시스템을 목표를 하고 있으므로, 시스템 내부에서는 암묵적으로 혹은 자동적으로 생성되는 정보를 효율적으로 처리하여야 한다.

본 장에서는 선체 CAD 시스템 전반에 걸쳐서 공통적으로 사용되는 알고리즘 중에서 다음과 같은 사항에 대해서 설명한다.

- 대상 객체의 Naming Conventions
- Plate 정의에 따른 내부처리 기법
- 구획 정의에 따른 내부처리 기법

4.2 대상 객체의 Naming Conventions

자료구조의 상세화 과정에서 도출된 대상객체(혹은 클래스)들의 멤버데이타중에서 가장 기본적인 것은 이들을 자료구조내에서 유일하게 식별할 수 있도록 해주는 객체이름이다. 객체이름의 예를 들면, Ship 클래스의 ShipID,DCU의 DCUID,Tank의 TankID, Plate의 PlateID등을 들 수 있으며 이들 이름은 GUI를 통해서 입력된 정보에 의해서 대부분이 자동적으로 생성되어야 한다. 이렇게 생성된 객체이름은 사용자의 필요에 따라서 사용되어 해당 객체를 지정하는데 이용된다. 이와같은 작업이 일관성 있게 수행되기 위해서 선체 CAD 시스템에서는 다음과 같은 Naming Conventions 메카니즘을 내부적인 알고리즘으로 처리하도록 하였다.

1) 부재에 대한 Naming Conventions

Name = (소속)+(부재명칭)+(위치)+
(Side Indicator)+(Serial Number)

여기서, (소속)이란 해당부재가 정의된 (혹은 설계된) 상위계층의 구획 (Compartment/Tank/Hold)을 말하며, 이를 통해서 이 부재의 기하학적인 위치를 간접적으로 알 수 있다. 따라서 기하학적인 계산에 의하지 않고도 어떠한 부재의 기하학적인 소속관계를 알 수 있고, 기하학적인 범위를 지정할 경우에도 기하학적 계산에 의하지 않고 단지 소속 이름을 지정하므로써 내부처리의 속도를 대폭 향상 시킬 수 있는 부수적인 효과도 있다.

(부재명칭)은 설계자의 입장에서 친숙한 설계용어를 사용하여 선체구조를 구성하는 부재(혹은 부품)의 이름을 식별하도록 하였다. 한가지 예로써 DECK라 하면 시스템 내부적으로 DECK가 가져야 할 기본적인 특성인 위상관계, 기하학적 관계를 이미 인식하고 내부적으로 처리된다. 이 경우에 DECK는 반드시 Shell에 의해서 경계지워지고, XY-Plan 상에서 정의되었으며, 종통부재인 DeckLongi를 갖고 있다는 사실을 시스템이 인식하도록 하였다.

(위치)는 유사한 부재들을 식별하기 위한 방법으로 기하학적인 위치를 나타내는 기호로써 사용될 뿐이고, 부재의 기하학적 위치를 절대적으로 표현하지는 않는다. TransBulkhead 경우에는 Frame번호나 X좌표로써 표현되고, Hopper Slant Plate와 같이 Non-Vertical Plan에 대해서는 -1로 표시한다.

(Side Indicator)는 배의 Port Side, Starboard Side를 가리킨다. 선체 구조상에는 좌우대칭으로 설계되는 부재가 많으므로 대칭인 부재는 B(Both)로 표기하고 시스템 내부적으로는 P(Port),S(Starboard)의 두개의 대칭인 부재를 자동생성 하도록 알고리즘을 정의하였다. 또한 배의 CenterLine상에 정의된 부재는 C(Center)로 표현한다.

(Serial Number)는 앞의 4개의 부분으로는 유일한 식별이 곤란할 때 시스템에서 자동적으로 부여하는 번호이다. 하나의 부재가 다른 부재에 의해서 절단 되었을때 Parent 부재는 Serial Number가 0이고 Child 부재는 1부터 순서적으로 번호를 부여 받는다.

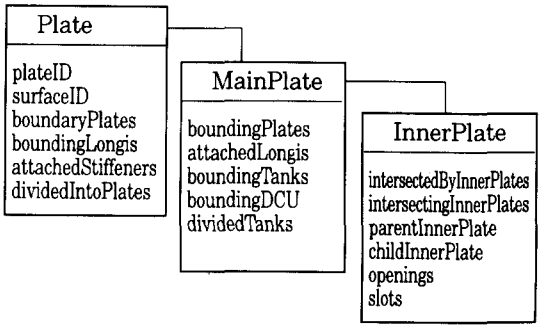

Fig. 4 Class inheritance of innerplate class

2)Tank와 Hold에 대한 Naming Conventions

Name=(소속)+(Tank/Hold명칭)+(SideLndIcator)

여기서,(소속)은 상위계층의 구획이나 Tank,Hold를 가리키므로써 Space의 계층구조를 표현한다. (Tank/Hold 명칭)은 기능상의 명칭이거나 번호로 부여된다.

4.3 Plate 정의에 따른 내부처리 기법

1) InnerPlate의 정의

InnerPlate는 클래스의 계층구조상 Plate 클래스와 MainPlate 클래스의 두개의 상위 클래스로 부터 계승을 받고 있다. 이들 클래스의 멤버데이타를 도식화 하면 Fig. 4와 같다. 즉, InnerPlate의 상위클래스인 Plate클래스와 MainPlate 클래스의 멤버데이타가 정의되므로써 InnerPlate객체는 완전한 인스탄싱이 됨을 알 수 있다. 선체모델의 객체모형도에 의하면 Inner_Plate 클래스는 PlanarSurface 클래스와 has geometry관계를 갖게 되므로써 InnerPlate의 형상이 구체화 된다. PlanarSurface는 Mould Curve와의 관계에 의해서 SurfaceEdge, BoundaryCurve, Trace-Curve 등을 정의하게 된다.

이와같은 InnerPlate와 연관된 클래스들을 인스탄싱하고 해당 멤버데이타를 생성해 가는 알고리즘의 이

름을 "ip_func_n"이라 설정하고 이 알고리즘의 세부내용을 PDL(Program Description Language)로 정의하였다. 여기서 사용한 PDL은 범용성을 띤 표준화된 프로그램밍 언어는 아니고, 본 연구에서 객체지향 프로그램밍에 적합하도록 작성하였고, 엄격한 문법은 사용하지 않았으므로 쉽게 표현할 수 있다.

2)알고리즘 "ip_func_n"의 내부처리 절차

"ip_func_n"은 "ip_func_1", "ip_func_2", "ip_func_3", "ip_fune_4"등의 4개의 모듈로 구성되며, "ip_func_1"은 해당 Plate의 Surface 기본정보 만으로 (Knuckle 정보 포함) Plate를 생성하는 기능을 갖으며, "ip_func_2"는 Surface 기본정보와 다른 부재와의 Boundary 구속관계를 동시에 알 수 있을 때 사용하고, "ip_func_3"은 이미 정의된 Plate에 Boundary 관계를 부여하고, "ip_func_4"는 이미 정의된 Plate에 Intersection 관계를 지정하는 기능을 갖는다. 이중에서 "ip_func_2"에 대한 알고리즘을 PDL로 작성한 예를 다음에서 보여준다.

ip_func_2("HPDCU_TBHD_f1_O_O", surface의 기본정보, boundary 관계의 기본정보)

```
InnerPlate의 인스탄싱 ip1
    ip1.plateID = "HPDCU_TBHD_f1_0_0"
    ip1.surfaceID = "PS_TBHD_f1_0_0"
planarSurface의 인스탄싱 psl
    ps1.surfaceID = "PS_TBHD_f1_O_O"
    ps1.localOrigin = lorigin
    ps1.box3d = box3
    ps1.pcoef = nvector
    ps1.planeType = "YZ"
    ip1.boundaryPlates =
        ["SHIP_Shell", "SHIP_DECK" 추가]
    ps1.moldedCurves =
      ["BC_/PS_TBHD_f1_0_0/SS_Shell/",
       "BC_/PS_TBHD_f1_0_0/PS_Deck/"]
    ps1.surfaceEdges =
      ["SE/PS_TBHD_f1_0_0/SS_Shell/",
       "SE/PS_TBHD_f1_0_0/PS_Deck/"]
BoundaryCurve의 인스탄싱 bc1
    bc1.moldedCurveName=
      "BC_/PS_TBHD_f1_0_0/SS_Shell/"
    bc1.curveGeometryType = "SPLINE"
    bc1.boundarySurfaceName = "SS_Shell"
Spline의 인스탄싱
BoundaryCurve의 인스탄싱 bc2
    bc2.moldedCurveName =
      "BC_/PS_TBHD_f1_0_0/PS_Deck/"
    bc2.curveGeometryType = "POLYLINE"
    bc2.boundarySurfaceName = "PS_Deck"
Polyline의 인스탄싱
SurfaceEdge의 인스탄싱
"SHIP_Shell"에 해당되는 Shell의 인스탄스 sh1
  sh1.boundingPlates =
    ["HPDCU_TBHD_f1_0_0"추가]
"SHIP_Shell"에 해당되는 SculpturedSurface의
인스탄스 ss1
  ss1.moldedCurves =
 ["TC_/SS_Shell/PS_TBHD_f1_0_0/"추가]
TraceCurve의 인스탄싱 tc1
  tc1.moldedCurveName =
    "TC_/sS_Shell/PS_TBHD_f1_0_0/"
```

4.4 구획 정의에 따른 내부처리 기법

주요격벽이 정의되면 (다른 표현으로는 "Major Surface, Minor Surface가 정의되면") 시스템 내부적으로는 Space가 자동생성 된다. 여기서 Space란 구획, Tank, Hold등을 말하며, 이들간의 계층관계, 소속관계를 유지한다.

본절에서는 Space의 예로써 Hold내의 Port Side Hold("HOLD_SHOLD_P")에 대한 내부처리 알고리즘을 설명한다. 선체모델에서 Space의 형상은 일반적인 다면체의 형상을 취하는 것은 아니고, 몇가지의

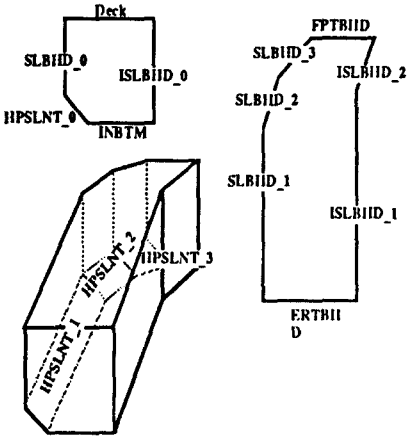

Fig. 5 View of "hold_Shold_P"

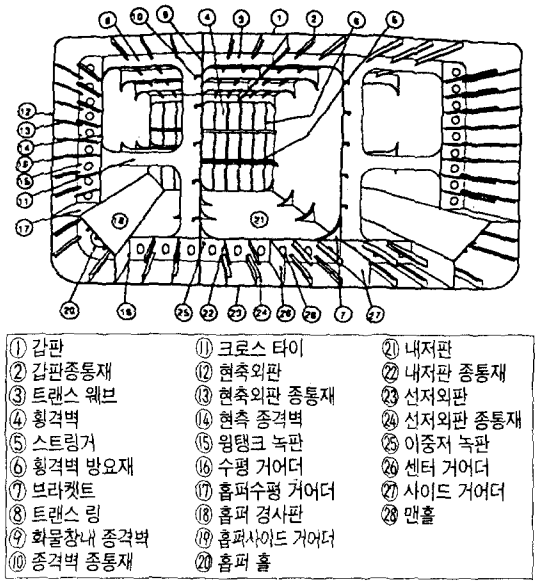

Fig. 6 Stuctural components of ship structural model

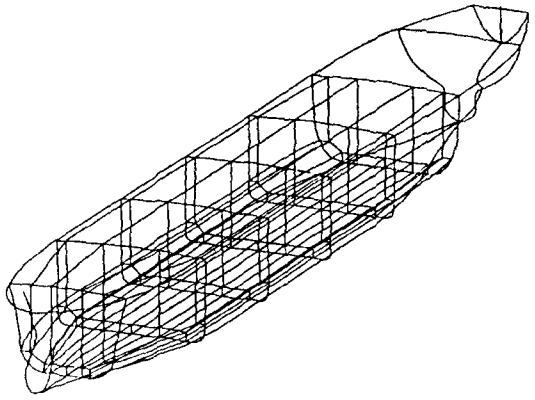

Fig.7 Arrangement of girders in ballast tank

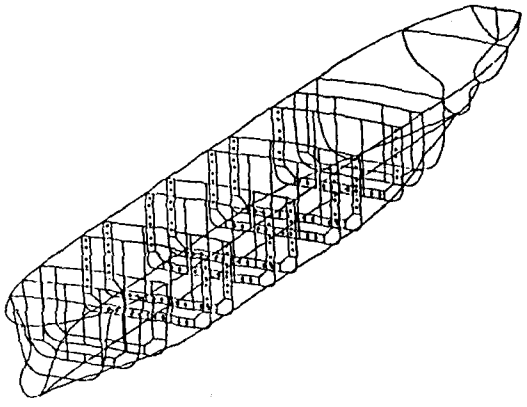

Fig.8 Arrangement offloors in ballast tank

유형으로 분류된 단면형상을 배길이 방향으로 Sweeping한 형상을 기본적으로 갖는다. 그러나 단순한 Sweeping의 결과는 아니고, 배길이 방향(X축 방향)의 Face Plate들은 Knuckled Plate로 구성되고, 선체 외판의 곡면부가 Face Plate로 구성되기도 한다. "HOLD_ SHOLD_ P"에 대한 Space 형상을 도식화하면 Fig. 5와 같다. 여기서, SLBHD(SideLongiBulkhead),ISLBHD(InsideLongiBulkhead),HPSLNT(HopperSlantplate)등의 Plate는 X-축 방향으로 진행하면서 Knuckled Plate들의 조합으로 이루어지므로 시스템내에서는 이들 Plate들의 부재명칭의 마지막 필드인 Serial-Number를 찾아서 해당 Hold의 Face가 되도록 Space를 생성한다.

4.5 내부처리 기법의 검증에 관한 예

본 논문에서 제시한 선체모델의 객체모형, 이를 상세화한 자료구조 및 내부처리 기법의 정합성을 검증하기 위해서, 대상선박을 이중선각 유조선의 화물창부로 Fig.6과 같이 설정하고, 유닛모델링 기법에 따라 선체구조의 각 부분에서 보여지는 특징적인 구조를 유닛으로 하여 선체구조 정보를 설계 변수화하여 파라메트릭 설계기법을 이용하여 선체모델을 컴퓨터에 정의 하였으며, 그 결과의 한 예로써 Fig.7은 Ballast Tank내에 설치되는 Girder의 배치를 나타내고, Fig. 8에서는 Floor의 배치를 확인할 수 있다.

5. 결 론

본 논문에서는 선체 CAD 시스템 개발을 위한 첫단계인 시스템의 요구분석 및 기능분석과 더불어 컴퓨터 내에서 시스템 구현을 위한 기반기술로써.다음과 같은 사항을 중점적으로 개발하였으며 이 결과를 바탕으로 하여 향후 선체 CAD 시스템의 본격개발을 추진할 계획이다.

(1) 선체모델 구축을 위한 객체의 도출:

선체모델을 구축하기 위해 필요한 세분화된 객체 및 관계를 모델링한 객체모델을 도출하여 데이타베이스로 구현하였다[3].

(2) 제품모델을 기초로한 자료구조의 정의:

선체설계 단계시 요구되는 광범위한 모델정보와 물량정보를 정확히 표현하기 위해서는, 제품모델(Product Model) 개념에 기초를 둔 선체모델을 구현해야 하며, 이를 구체화 하기 위한 자료구조를 확립하였다.

(3)선체모델링을 위한 내부처리기법의 개발:

선체모델 내에서 설계정보의 동적인 변화과정을 알고리즘 형태로 개발하여 구현하였다.

후 기

본 연구는 CSDP 사업과 관련하여 수행된 연구결과의 일부임을 밝혀 둔다.

참 고 문 헌

[1] Rumbaugh James, Blaha Michel,"Object oriented modeling and design, Prentice hall inc.,1991.

[2] 이규열, 서승완 외, "CSDP(IV)-종합시스템" 한국기계연구원 선박해양공학연구센터 연구보고서 ,UCN272-1706.D,1993

[3] 신동우 외, "CSDP(IV)-선박설계생산 데이타베이스관리시스템" 한국기계연구원 선박해양공학연구센터 연구보고서, UCN 272-1707.D,1993

[4] 김광욱,서승완,김원돈 외,"CSDP(IV)-선체CAD시스템" 한국기계연구원 선박해양공학연구센터 연구보고서, UCN272-1709.D,1993

[5] 장석호,장옥현 외,"CSDP(IV)-선체모델링기법" 현대중공업 CSDP 연구보고서,1993

[6] 윤덕영 외, "CSDP(IV)-초기공정 및 일정계획 정보처리 시스템" 대우조선 CSDP 연구보고서,1993

[7] 봉현수 외, "CSDP(IV)-선체상세설계지원 전문가시스템" 대우조선 CSDP 연구보고서, 1993

[8] 강원수, 서승완, 이규열, "객체지향 선박구획정의표현방법론", *대한조선학회논문집*, 제30권 제3호, 1993

[9] R. Bronsart."Design and management of product model", Schifftechnik, 1990

[10] Toshiba Nomoto, Kazuhiro Aoyama,"The product definition system for oil-tanker: Computer aided information acquisition system of design and manufacturing in shipbuilding (part3) ",*日本 朝鮮學會論文集*, 弟 169號, 1991

[11] STEP, "Part 102:Ship Structure", ISO TC 184/SC4/WG1 Document N411,1989