

---

 論 文
 

---

大韓造船學會論文集  
 第 31 卷 第 2 號 1994年 5月  
 Transactions of the Society of  
 Naval Architects of Korea  
 Vol. 31, No. 2, May 1994

## 수퍼 어플리케이션 개념에 의한 유연화된 개념설계시스템 개발에 관한 연구

이동곤\*, 이경호\*, 이순섭\*, 이규열\*, 한순홍\*\*

A Study on the Adaptable Conceptual Design System Based on  
 Super-Application Concept

by

Dong-kon Lee\*, Kyung-Ho Lee\*, Soon-Sub Lee\*,  
 Kyu-Yeul Lee\* and Soon-Hung Han\*\*

### 요 약

선박의 개념설계를 지원하는 전산시스템은 창조적인 설계를 지원할 수 있을뿐만 아니라, 작업환경의 변화에 유연하게 대처할 수 있는 기능이 필요하다. 본 연구에서는 기존의 개념설계시스템의 기능을 확장한 유연화된 개념설계시스템을 개발하였다. 시스템을 구성하는 각 모듈들은 독립적인 수행이 가능하도록 개발되었으며, 수퍼 어플리케이션 개념을 이용하여 상위레벨에서 각 모듈들을 제어하는 기능을 추가하였다. 수퍼 어플리케이션 개념의 제어 프로그램을 SUN SPARC Workstation의 UNIX C Shell 환경하에서 개발하였다.

### Abstract

A computer system supporting conceptual design is required to have not only a design model for creative works but also flexibility to cope with changing environments. In this research, a new conceptual design system was developed by expanding functions of existing systems. The modules of the system were maintained to have independency so that each unit module could be executed alone. The adaptable conceptual design system based on the super-application concept which control each module in high level was developed using UNIX C Shell of SUN SPARC workstation.

---

발표일자 : 1993년도 대한조선학회 추계연구발표회('94.11.13)

접수일자 : 1993년 12월 23일, 재접수일자: 1994년 4월 25일

\* 정희원, 선박해양공학연구원

\*\* 정희원, 한국과학기술원

## 1. 서 론

선박설계에 있어서 개념설계는 제한된 시간과 자료를 이용하여 선주요구 조건에 맞는 선박의 사양을 결정하는 단계이다. 개념설계는 설계의 시작 단계이므로 설계의 자유도(Degrees of Freedom)는 높은 반면에, 실험이나 해석으로부터 얻어진 이용가능한 자료는 제한되어 있다[1]. 따라서 설계자는 자신의 창조력과 과거의 설계경험을 총동원하여 설계를 하게 된다. 이러한 개념설계를 지원하는 전산시스템은, 창조적인 설계작업이 가능하도록 설계모델을 구성하여야 함은 물론, 상황변화에 원활하게 대응할 수 있는 유연성을 가져야 한다. 지금까지의 개념설계시스템 개발경향을 살펴보면, Batch Version 으로 개발된 시스템[2]을 거쳐 최근에는 사용자의 입장을 고려한 대화형 개념설계 모델 [3]이 발표되었다. 이들 시스템들은, 시스템설계라는 목표가 강조되어 통합된 형태로 개발되어 왔다. 그러나, 시스템을 구성하고 있는 여러가지 기능의 단위 모듈들이, 어떤 특정 부분의 업무만을 위해서는 단독으로도 사용가능하도록 보다 유연하게 구성되어야 할 필요가 있다.

본 논문에서는, 기존의 개념설계시스템이 가지고 있는 문제점을 해결한 새로운 개념의 개념설계시스템을 개발하였다. 개념설계시스템을 구성하는 단위 모듈들을 단독으로 사용 가능하도록 독립성을 유지함과 동시에 시스템 차원의 상위레벨에서 각각의 모듈들을 통합 및 제어하는 슈퍼 어플리케이션(Super-Application) 개념을 이용하여 유연화된 대화형 개념설계시스템을 개발하였으며, 그 주요내용은 다음과 같다.

- 1) 모듈들의 통합/제어 기능 개발 : 독립된 모듈들의 통합, 임의 모듈 수행을 위한 모듈간 작업 순서의 제어 및 모듈들의 실행 상태 감시를 위한 Monitoring 기능
- 2) Back Tracking 기능 개발 : 반복설계 과정과 대화형의 작업 환경을 효율적으로 지원할 수 있도록, 작업 순서의 변경을 위한 Back Tracking 기능
- 3) 작업 History의 보관 기능 개발 : 사용자가 입력한 데이터와 모듈에서 생성된 데이터를 ShipID를 이용하여 식별하는 작업 History 보관 기능

이상의 주요기능들을 SUN SPARC Workstation 의 UNIX C Shell 환경하에서 구현하였다.

## 2. 기존 시스템의 기능개선

### 2.1 개선사항 도출

비교적 최근에 개발된 사용자 지향 대화형 개념설계 시스템인 BASCON-II[3]의 사용경험[4]을 근간으로 시스템 개선사항을 도출하였다. BASCON-II는 그래픽 사용자 인터페이스(Graphical User Interface)를 도입하고 대화형으로 개발되었기 때문에, 기존 시스템에 비하여 기능이 많이 향상되었으나 현업에 적용해 본 결과 다음과 같은 불편한 점들이 발견되었다.

- 1) 시스템 설계라는 목표가 강조되어 설계모듈들이 하나의 덩어리로 개발되어 졌기 때문에, 필요한 모듈만을 실행할 수 없고 항상 처음부터 시스템 전체를 수행하여야 한다.
- 2) 설계모듈들이 하나의 덩어리로 결합되어 있으므로 모듈의 대체가 용이하지 않다.
- 3) 작업중에 시스템을 종료하였을 경우, 지금까지 작업한 내용을 다음 작업에 이용하지 못하고 처음부터 작업을 다시 수행하여야 한다.

이상과 같은 불편한 점을 개선하기 위해서 첫째, 사용자가 원하는 모듈만을 수행할 수 있도록 각 기능별 주요 모듈들을 분리하고 분리된 모듈들이 독립적으로 수행될 수 있도록 한다. 분리된 모듈들 간의 입출력 데이터를 분석하여 각 모듈간의 연관성을 파악하고, 이를 토대로 상위 레벨에서 전체 모듈을 제어하는 프로그램을 개발하여 통합한다. 둘째, 분석된 모듈간의 입출력 데이터를 근간으로 모듈의 대체를 보다 용이하게 한다. 셋째, 작업 History 를 보관하여 사용자로 하여금 보다 편리하게 사용할 수 있도록, 사용자가 입력한 각종 데이터를 ShipID를 이용하여 데이터 File에 보관하고, 보관된 데이터 File 은 추후 작업시에 ShipID를 이용하여 재 사용할 수 있도록 한다.

### 2.2 모듈분리

개념설계 프로그램인 BASCON-II를 각 기능별로 분류하여, 모두 9개의 모듈로 분리하였으며 분리된 모듈의 이름과 기능은 다음과 같다.

- 1) PPDB : 실적선 데이터베이스를 검색하여, 적절한 Mother Ship을 선정하고 설계선의 주요치수를 결정
- 2) ENGINE : 선박의 저항추진 성능을 추정하고, 이를 바탕으로 주기관 데이터베이스에서 설계선

- 에 적합한 주기관을 선정
- 3) LONGIDIV : 길이방향의 구획분할을 수행
  - 4) WEIGHT : 실적선 자료를 토대로 설계선의 경하중량을 추정
  - 5) HULLFORM : 실적선 선형 데이터베이스에 저장되어 있는 선형자료를 이용하여, 설계선의 특성에 맞는 선형을 선형변환기법을 통하여 도출
  - 6) MIDSHIP :선박의 중앙 횡단면 형상 결정
  - 7) HOLDSECTION : 화물창부의 화물창 Knuckle Point 위치 결정
  - 8) VOLUME : 화물창부를 비롯한 각종 구획의 용적계산 수행
  - 9) FREEBOARD : 건현 계산 수행

2.3 모듈간 정보교환

분리된 모듈간의 정보교환과 상위레벨에서의 통합을 위하여, 각 모듈에서의 입력 자료와 출력 자료를 분석하였다. 독립된 모듈간의 정보교환 방법으로는 다음의 2가지 방안이 검토되었다 (Fig. 1 참조).

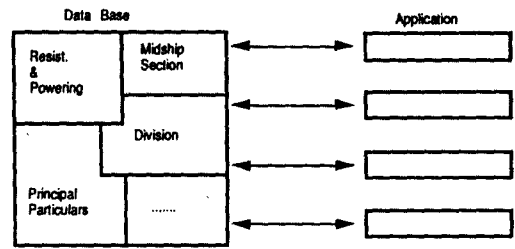
2.3.1 Data Base를 통한 정보교환

시스템 차원에서 데이터를 분석하고 이를 토대로 데이터베이스를 설계, 구현한 후, 데이터베이스를 통하여 모듈간 정보를 교환하는 방법이다. 이 방법은 예상되는 모든 데이터를 데이터베이스에 포함시키고, 데이터베이스가 잘 설계된다면 효과적인 방법이라 할 수 있으며, 규모가 큰 시스템에서는 필수적이다. 그러나, 시스템이 개발단계에 있을 경우에는 데이터의 양이 점차 증가하게 되므로 지속적인 데이터베이스의 수정과 보완이 필요하게 되는 단점이 있다.

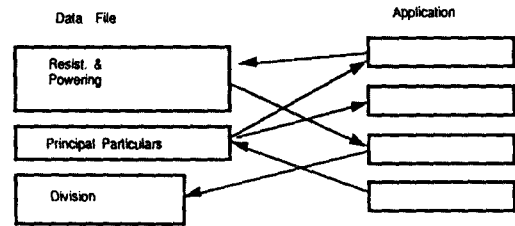
2.3.2 Data File을 통한 정보교환

선행 모듈에서 생성한 데이터를 데이터 File에 기록하면, 후행 모듈에서는 선행 모듈에서 생성한 데이터 File을 통하여 필요한 데이터를 읽어가는 방법으로서, 여러개의 데이터 File을 만들어야 하지만, 시스템의 기능을 점진적으로 개발하고 확장하는 단계에서는 보다 간편한 방법이다.

본 논문에서는, 시스템이 개발단계에 있음을 고려하여, 즉 단위 모듈들의 대체를 보다 효과적으로 할 수 있도록 두번째의 방법을 사용하였다. 데이터 File들은 사용자가 입력하는 ShipID를 이용하여 보존되며, 나중에 다시 작업을 수행할 경우 ShipID 를 참조하여 관련 데이터를 불러올 수 있다.



a) Data Base



b) Data File

Fig. 1 Information exchange methods between individual modules

2.4 모듈간 연관성 분석

분리된 모듈들과 각 모듈의 수행에 필요한 데이터의 분석을 통하여, 임의의 모듈을 수행하는데 필요한 선행 모듈의 수행조건을 도출하였으며, 상세한 내용을 Table 1 에 나타내었다. 예를 들면, 선박의 경하중량 추정을 위한 중량추정 모듈인 WEIGHT 를 수행하기 위하여서는, 설계선의 주요치수와 실적선 데이터 및 선주의 요구조건 자료가 필요하므로 이들 자료를 생성하는 주요치수 결정 및 실적선 자료에 관련된 PPDB 모듈이 먼저 수행되어야 한다. 또한, 주기관의 중량자료가 필요하므로 주기관의 중량을 계산하는 ENGINE 모듈도 수행되어야 한다. 이렇게, 모듈간 연관성 분석을 통하여 얻어진 정보를 이용하여 임의의 모듈의 수행을 가능하게 할 수 있다.

3. 수퍼 어플리케이션 개념에 의한 제어 모듈 개발

하나로 얽혀 있던 프로그램(BASCON-II)을 각 기능별로 분리하여 독립모듈화하였다. 따라서, 어떤 모듈을 수행하기 위해서는 먼저 수행하여야 할 모듈을 선정해야하고, 이들 모듈의 수행 순서를 결정하며, 기 생성된 관련 데이터 File이 있는지도 조사되어

Table 1 Relationship between each modules

Module Name	Related Module Name
PPDB	-
ENGINE	PPDB
LONGIDIV	PPDB ENGINE
WEIGHT	PPDB ENGINE
HULLFORM	PPDB ENGINE LONGIDIV
MIDSHIP	PPDB ENGINE LONGIDIV
HOLDSECTION	PPDB ENGINE LONGIDIV HULLFORM MIDSHIP
VOLUME	PPDB ENGINE LONGIDIV HULLFORM MIDSHIP
FREEBOARD	HOLDSECTION PPDB

야 한다. 이러한 기능을 갖는 제어 프로그램을 슈퍼 어플리케이션(Super-Application) 개념에 의한 "UNIX Shell Programming" 기법[5,6,7]을 이용하여 개발하였다. 제어 프로그램을 SUN SPARC의 C Shell 환경에서 구현하였다.

### 3.1 UNIX Shell 프로그래밍

Shell은 컴퓨터 운영체제의 일종인 UNIX 시스템의 유틸리티 프로그램으로서, 사용자로부터 명령을 받아 UNIX 시스템으로 하여금 사용자의 명령을 수행하도록 한다. 즉, Shell은 사용자와 UNIX 시스템 간을 연결시켜주는 일을 하며, 대화형의 작업환경을 제공한다. Fig. 2와 같이, Shell은 Kernel 부분을 감싸고 있다. Kernel은 UNIX 시스템의 핵심부분으로 Hardware와 직접적으로 관계된다[8]. Shell은 Kernel 부분을 통하여 사용자의 명령을 수행한다.

### 3.2 제어 프로그램의 개발

여러개의 Shell 명령을 하나의 File에 저장하여 프로그램과 같이 사용할 수 있도록 하는 기능, 즉 Shell Script를 이용하여 작업 History의 보관, 임의 모듈

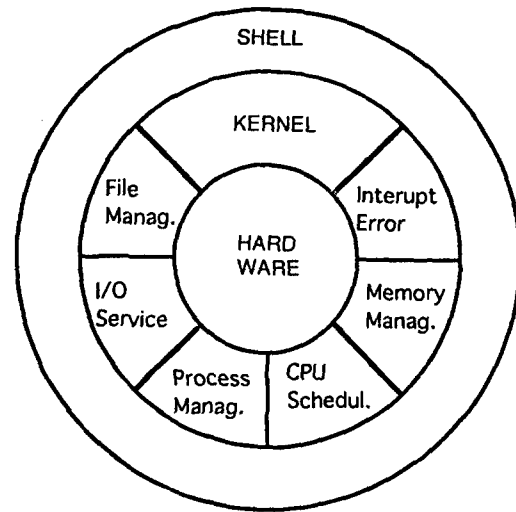


Fig. 2 Kernel of the UNIX

수행 기능, Back Tracking 기능 및 전체작업의 제어 등에 관한 기능을 프로그래밍 하였다.

#### 3.2.1 임의 모듈수행 기능

사용자가 일부 모듈만을 수행하기를 원하는 임의 모듈수행 기능은, 각 모듈간의 선행관계를 파악하여 구현하였다. 즉, 사용자가 수행하기 원하는 모듈들의 상관관계를 각각 파악하고, 이것을 종합하여 상위 레벨에서 수행순서에 따라 작업을 제어한다.

#### 3.2.2 작업 History의 보관

사용자가 작업을 수행하다가 중단하였거나, 기존에 작업한 데이터를 이용하여 새로운 작업을 하고자 하는 경우를 효과적으로 지원하기 위하여, 사용자가 시스템을 사용할때 입력하는 ShipID를 이용하여 작업 History의 보관 기능을 구현하였다. 먼저, 시스템은 사용자가 입력한 ShipID와 같은 이름을 갖는 데이터 File이 있는지를 검색한다. 같은 이름이 있다면, 이들 데이터 File을 프로그램이 사용할 수 있도록 이름을 바꾸어 주고, 관련 데이터 File을 생성하는 선행 모듈은 수행을 하지 않는다. 만약 데이터 File이 없다면, 시스템은 작업 순서에 따라 정상적인 수행과정을 따르게 된다. 작업이 모두 끝나게 되면, 프로그램에서 사용하던 데이터 File의 이름을 사용자가 입력한 ShipID를 갖는 이름으로 바꾸어 준다.

3.2.3 Back Tracking 기능

시스템 수행중에 이미 수행한 선행 모듈을 다시 수행하는 기능(Back Tracking)은, 모듈들이 하나의 덩어리로 결합되어 있을 경우에는 별 문제가 되지 않는다. 그러나 여기서는, 모듈들이 독립적으로 분리되어 있고 작업 History의 보관기능과 임의 모듈 수행기능이 결합되어 있기 때문에 상당히 복잡하다.

여기서는, Back Tracking 시에 Back Tracking을 인식할 수 있도록 제어 변수를 할당하고, 사용자가 Back Tracking하기를 원하는 모듈의 수행 절차상의 위치를 인식하게 함으로서 Back Tracking 기능을 구현하였다. 즉, 사용자가 Back Tracking을 위한 기능을 선택하면, 프로그램은 임의 모듈의 수행을 위한 시작점의 위치로 되돌아 간다. 다음에, 사용자가 처음에 선택한 모듈과 Back Tracking 모듈을 결합하여 상위 레벨에서 전체적으로 수행순서를 재구성하고 작업을 수행한다. 이때, 사용자가 지정한 Back Tracking 모듈보다 선행되는 모듈은 수행되지 않으며, Back Tracking 모듈보다 수행순서가 뒤쪽에 있는 모듈들은 관련 데이터 File이 생성되어 있어도 무시하고 작업을 수행한다. 임의 모듈수행 기능, 작업 History 보관 기능 및 Back Tracking 기능의 구현을 위한 Pseudo Code를 Fig. 3에 나타내었다.

4. 유연화된 대화형 개념설계시스템 개발

임의 모듈수행 기능, 작업 History의 보관 기능 및 Back Tracking 기능을 결합하여 상위 레벨에서 작업을 제어하는, 수퍼 어플리케이션 개념의 제어모듈을 UNIX Shell 프로그래밍 기법을 이용하여 개발하였다. 개발된 제어모듈을 중심으로 개념설계시스템을 구성하는 9개의 분리된 모듈과 그래픽 사용자 인터페이스(Graphical User Interface)를 통합하여 유연화된 대화형 개념설계시스템(BASCON-III)을 개발하였고, Fig. 4에 그 구성도를 나타내었다. Fig. 4에서와 같이 모듈들이 서로 연결되어 있지 않으므로, 사용자가 수행하고자 하는 모듈만을 독립적으로 수행 가능하다. 한편 각 모듈들이 분리되어 있음에도 불구하고, 시스템 설계를 효율적으로 지원할 수 있는 임의 모듈수행기능과 Back Tracking 기능도 동시에 구현되었다.

이상과 같이 설계모듈들이 독립성을 유지하면서 통합된 형태로 개발되었기 때문에, 사용자가 보다 편리하게 사용할 수 있으며 시스템의 유지 및 보수가 쉽고 확장이 용이하다. 예를들어, 사용자의 설계환경을 고

```

read ShipID and Execution Modules No.
set Job_Status = normal
for (all file with ShipID)
    change name ShipID.name.dat to name.dat
end
foreach module (all module)
    sort $module from module list
end
set back_module
BackTrack:
If (Job_Status = normal) then
    set accumulation_module, runmodule
end
if (no. of runmodule != no. of accumulation_module) then
    while (no. of runmodule)
        if (runmodule[1] != accumulation_module[1]) then
            goto Next
        else
            shift runmodule
            shift accumulation_module
        endif
    else
        while (no. of runmodule)
            if (runmodule[1] != back_module) then
                shift runmodule
            else
                goto Next
            endif
        end
    while (no. of runmodule)
        if (exist datafile) execute module
        set back_module
        if (back_module != continue) goto BackTrack:
    shift runmodule
end
for (all data file)
    change name name.dat to ShipID.name.dat
end
    
```

Fig. 3 Pseudo code for control program

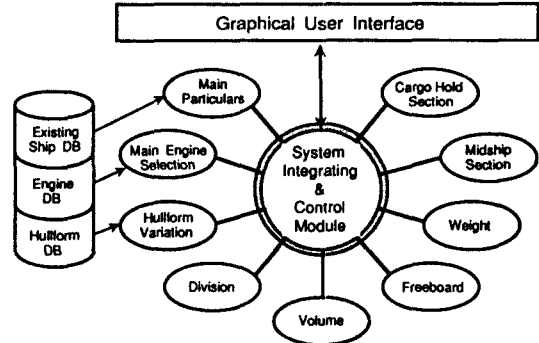


Fig. 4 System configuration

려하여 성능추정 모듈을 사용자 고유 모듈로 대체할 필요가 있을 때에는, 사용자 고유 모듈의 입출력 데이터를 파악하고, 대체하고자 하는 BASCON-III의 해당 모듈의 입출력 데이터와 비교한다. 만약, 사용자 모듈의 입력 데이터가 BASCON-III 모듈의 입력 데이터 보다 많은 입력을 필요로 하면 이들 데이터를 생성

하는 루틴을 만들어 준다. 한편, 사용자 모듈이 BAS-CON-III의 후행 모듈에서 필요로 하는 입력 데이터를 생성하지 않으면, 이들 데이터를 생성하는 루틴이나 기능을 만들어 준다. 마지막으로 입출력 데이터에 문제가 없으면, 제어 프로그램에서 해당 모듈의 이름을 대체하거나, 사용자 고유의 모듈의 이름을 BAS-CON-III의 모듈 이름과 같게 한다. 또한 새로운 기능을 갖는 모듈을 추가 할 경우에는 입력 데이터만을 검토하고, 제어 프로그램에 모듈의 이름을 추가하면 된다.

## 5. 결 론

기존의 개념설계시스템들이 가지고 있는 문제점을 해결한 새로운 개념의 개념설계시스템을 개발하였다. 시스템을 구성하는 단위 모듈들을 단독으로 사용 가능하도록 독립성을 유지함과 동시에 시스템 차원의 상위 레벨에서 각각의 모듈들을 통합제어하는 슈퍼 어플리케이션 개념을 이용하여 유연화된 대화형 개념설계시스템을 개발하였으며, 그 주요 결과는 다음과 같다.

- 1) 모듈들의 통합/제어 기능 개발 : 독립된 모듈들의 통합, 임의 모듈 수행을 위한 모듈간 작업 순서의 제어 및 모듈들의 실행 상태 감시를 위한 Monitoring 기능을 개발하였다.
- 2) Back Tracking 기능 개발 : 반복설계 과정과 대화형의 작업 환경을 효율적으로 지원할 수 있도록, 작업 순서의 변경을 위한 Back Tracking 기능을 개발하였다.
- 3) 작업 History의 보관 기능 개발 : 사용자가 입력한 데이터와 모듈에서 생성된 데이터를 ShipID를 이용하여 식별하므로써 작업 History의 보관 기능을 개발하였다.
- 4) 상기의 제어모듈을 중심으로 개념설계시스템을 구성하는 9개의 분리된 모듈과 3개의 데이터베이스 및 그래픽 사용자 인터페이스를 통합하여 유연화된 대화형 개념설계시스템을 개발하였다.

이상과 같이 작업 History의 보관 기능, 임의 모듈 수행 기능, Back Tracking 기능등을 갖는 유연화된 대화형 개념설계시스템이 개발되므로써, 사용자는 보다 편리하게 시스템을 사용할 수 있게 되었다. 나아가서, 사용자(조선소) 고유의 각종 성능추정 모듈들을 효과적으로 대체할 수 있기 때문에 사용자의 특성을 고려한 시스템을 구축하는 것이 가능하게 되었다.

## 후 기

본 논문은 과학기술처의 특정연구사업으로 수행한 "선박설계생산 전산시스템(CSDP) - 초기설계시스템" 개발 과제 4차년도 연구결과의 일부분임을 밝힌다.

## 참 고 문 헌

- [1] F. Mistree, et al, "Decision - Based Design : A Contemporary Paradigm for Ship Design", *SNAME Transaction*, 1990
- [2] 이규열 외, "선박설계생산 전산시스템(CSDP) - 초기설계시스템 및 종합시스템 개발(II)", 해사기술연구보고서, 1990
- [3] 이동곤 외, "사용자 지향 대화형 개념설계 모델", 대한조선학회 논문집, 1992년 11월
- [4] "제4회 CSDP Progress Meeting 자료", 해사기술연구소 CSDP 사업단, 1993년 3월
- [5] "HP - UX Shell Programming for General Users", 삼성휴렛팩커드, 1993
- [6] "UNIX Shell Programming", 월드컴퓨터테크, 1991
- [7] 조형진, "UNIX C Shell", 크라운출판사, 1991
- [8] 김태진, "유닉스의 이해", 가남사, 1993