

## 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

### STEP에 의한 조선 통합 생산 시스템(CIMS) 구현 방법

유상봉\*, 이재원\*, 정용문\*\*, 윤덕영\*\*\*, 김훈주\*\*\*

#### The Implementation Method of CIMS for Ship Manufacturing using STEP

by

S.B.Yoo\*, J.W.Lee\*, Y.M.Jeong\*\*, D.Y.Yoon\*\*\* and H.J.Kim\*\*\*

#### 요 약

통합 생산 시스템의 역할은 제품의 전 생명주기에 걸쳐 필요한 다수의 응용 프로그램을 통합하는 것이다. STEP은 독립적으로 개발된 응용 프로그램의 통합을 쉽게 하기 위하여 정보모델과 인터페이스를 정의하는 국제 표준화 작업이다. STEP을 이용한 조선 통합생산시스템(CIMS)에서 정보모델은 EXPRESS로 정의되어 있으며, 이 정보모델은 데이터베이스의 스키마로 번역되어 진다. 논문에서는 조선공정계획 시스템에 포함된 두개의 응용 프로그램(즉, 블록분할 시스템과 탑재 시스템)을 이용하여 개발된 프로토타입의 운용을 설명한다. 예로서 본 논문에서는 관계형 데이터베이스(Oracle)에 저장되어 있는 실제의 데이터를 보여준다.

#### Abstract

The role of CIMS(Computer Integrated Manufacturing Systems) is to integrate various applications throughout a product's life cycle. STEP is an international effort to standardize information models and interfaces so that independently developed applications can be easily integrated. A prototype for the Ship CIMS is built using STEP. In this prototype, the information model defined by EXPRESS is translated into database schemas. In this paper, we explain the operation of this prototype using the examples from two application programs, i.e., the Block Division System and the Erection System which are used for the process planning of ship manufacturing. As an example, Real data stored in a relational database system (Oracle) is presented in this paper.

발표일자 : 1993년도 대한조선학회 추계연구발표회('93.11.13)

접수일자 : 1993년 11월 29일, 재접수일자: 1994년 4월 25일

\* 정회원, 인하대학교 자동화공학과

\*\*인하대학교 대학원 자동화공학과

\*\*\* 정회원, 대우조선공업(주)

## 1. 서론

조선산업에 있어서 통합 생산 시스템(CIMS)의 필요성은 이미 많은 연구를 통하여 강조되어 왔고, 현재 일본, 미국, 유럽, 그리고 국내의 연구소와 산업체에서 연구와 시스템 구축이 진행 중이다[1][2][3]. 이러한 통합 생산 시스템은 전 산업 분야의 생산정보 통합을 위해 ISO(국제 표준화 기구, International Standard Organization)에서 추진 중인 STEP(Standard for the Exchange of Product model data)에 포함된다. STEP의 목적은 "제품의 전 생명주기를 통한 물리적, 기능적인 특성을 명백하고 완전하게 Computer로 정의하기 위함" 이라고 정의되어있다[4]. STEP은 다양한 응용 시스템의 통합을 위해 Open System으로 개발 중이며, 향후 국제표준으로 제정되면 많은 CAD/CAM/CAE 프로그램이 STEP Interface를 통하여 데이터를 교환할 수 있게 된다. 이러한 정보 모델 뿐만 아니라 정보의 교환과 공유도 STEP에 포함되어 있다. 즉 정보모델 언어인 EXPRESS로 정의된 정보를 서로 다른 프로그램 간에 교환 또는 공유하는 방법에 관한 표준의 개발이다. 이러한 정보의 교환과 공유는 File, 메모리 상의 Working Form, 또는 데이터베이스 시스템을 통하여 이루어진다.

본 논문에서는 조선 통합 생산 시스템중 블록 분할과 탑재에 관련된 정보를 중심으로 개발한 Prototype을 설명한다. 대상 선박은 VLCC Type으로서, 선각의 Cargo Holder의 일부인 링의 중앙평행부에 대한 정보를 다루었다. 데이터베이스 시스템으로는 관계형 시스템인 Oracle을 사용하였다. 먼저 EXPRESS로 표현된 정보 모델은 스키마 번역기에 의하여 데이터베이스 스키마로 변환되어 필요한 테이블을 정의한다. 블록분할시스템에서 블록분할 수행 후에 제공되는 블록에 관한 정보는 데이터베이스에 저장되고, 이러한 정보는 다음 공정한 탑재 시스템이 탑재 Activity를 결정하는 데 인출되어 사용된다.

## 2. STEP과 EXPRESS

EXPRESS는 STEP Project에서 공식적인 Information Modeling 언어로 개발되었고, STEP의 모든 Part의 개발에 공통으로 적용되어 사용되고 있다. EXPRESS는 그 동안의 성공적인 표준화 작업을 통하여 ISO의 국제투표표 1993년에 통과하였고 곧 국제표준으로 제정될 예정이다. Formal Information 언어로서 EXPRESS의 사용은 STEP으로 국한되지 않

고 EXPRESS User's Group 93 [5]에서 발표되었듯이 보편이나 가구산업 등과 같은 여러 분야에서 응용되고 있다.

### 2.1 조선산업에서 STEP의 이용

STEP의 구성은 크게 Overview, Description, Implementation Forms, Conformance Testing Methodology, General Resources, Application Resources, Application Protocols로 이루어져 있다. 이중 조선산업 관련 Part로는 Part 102: Ship Structures [6]가 있으며 이와 별도로 다음과 같은 새로운 Part의 개발이 최근에 시작되었다.

- Part 215 : Ship Arrangement
- Part 216 : Ship Moulded Forms
- Part 217 : Ship Piping
- Part 218 : Ship Structures

이와 같은 새로운 표준이 개발되면 조선 산업에 통합 생산 시스템의 개발은 더욱 용이해질 것이다. 하지만 이미 Part 102: Ship Structure에서 본 바와 같이 STEP의 정의는 매우 일반적인 것, 즉 모든 Ship Structure에 공통으로 적용할 수 있는 부분만 포함하고 있기 때문에 특정 조선소에서 그대로 사용하기에는 미흡하다. 이러한 것을 보완한 것이 200번부터 시작하는 각종 Application Protocol들이지만, 이것은 역시 개별회사에서 보완없이 사용하기에는 적절하지 않을 것이다. STEP에서는 전 산업 또는 특정 분야의 회사들이 공통으로 사용할 수 있는 정보모델을 정의하고, 각 개별회사는 여기에 자기회사가 필요한 정보모델을 첨가하여 개별적인 통합 생산 시스템을 구축할 수 있다. 이러한 경우 산업 공통의 정보(예, 표준 CAD File)는 산업체간 교환이 가능하며, 개별회사에서 정의한 정보(예, 자체 개발한 전문가 System의 Input/Output)는 회사내에서만 공유한다.

### 2.2 EXPRESS의 언어요소

EXPRESS는 Programming Language는 아니지만 Object의 정의와 명백하게 정의한 Data의 Integrity Constraint로 구성되어 있다[7]. EXPRESS의 언어요소에 대하여 간단히 소개한다.

#### 2.2.1 데이터 타입

EXPRESS Data Type들은 다음과 같이 요약될 수 있다.

- (1) Simple Data : Number, Real, Integer, Logical, Boolean, String, Binary.
- (2) Aggregate : Array, List, Bag, Set.
- (3) Constructed : Enumeration, Select.
- (4) Entity Data Type : Entity Type은 Data Type으로도 사용 가능하다.
- (5) 사용자가 정의한 Data Type : Abstract Data Type은 정의가 가능하다.

### 2.2.2 엔티티

엔티티의 정의는 Type Hierachy, Attributes, Local Rules를 포함한다. Type Hierachy는 SUB-TYPE이나 SUPERTYPE으로 정의한다. Attributes에는 Explicit, Derived, Inverse의 세가지가 있다. Local Rule들은 Entity Type의 각 Instance에 적용하는 Constraint를 명시하며, UNIQUE절이나 WHERE절에 의하여 정의한다.

### 2.2.3 오퍼레이터

Data Type이나 Entity Type을 다루는데 다양한 Operator가 제공된다. Operator는 Arithmetic, Relation, Binary, Logical, String, Aggregate, Entity등으로 구분된다.

### 2.2.4 실행문

EXPRESS는 Case, Escape, If-then, Repeat, While, Until, Return과 같은 실행문을 가지고 있으며 일반적인 알고리즘을 표현하는데 사용된다.

### 2.2.5 Function

EXPRESS 언어는 Abs, ASin, ACos, ATan, Cos, Exists, Exp, Log, Sqrt와 같은 Built-in Function을 가지고 있다. 또한 User들은 실행문을 이용하여 자신의 고유한 함수를 정의할 수 있다.

### 2.2.6 Global Rule

Global Rule은 하나 혹은 여러개의 Instance의 Set들 간의 Constraint를 명시하는 Reserved Word인 RULE로 정의할 수 있다. RULE은 실행문과 그 결과에 의해 적합성을 결정하는 WHERE절로 이루어진다.

### 2.2.7 EXPRESS의 간단한 예

```
ENTITY person
  first_name : STRING;
```

```
  last_name : STRING;
  birth_date : date;
  children : SET[0:?] OF person;
DERIVE ...
  age : INTEGER := years(birth_date);
INVERSE
  parents :SET[0:2] OF person FOR children;
END_ENTITY;
ENTITY female
  SUBTYPE OF (person);
  husband : OPTIONAL male;
  maiden_name : OPTIONAL STRING;
WHERE
  w1 : (EXISTS(maiden_name) AND
        EXISTS(husband)) XOR
        NOT EXISTS(maiden_name);
END_ENTITY;
```

이 예에서 Person과 Female의 2가지 Entity Type을 정의하였다. Entity Person은 First\_name, Last\_name, Birth\_date, 그리고 Children을 Explicit Attribute로 가지고 있다. Attribute인 Age는 Birth\_date에서 Derived된 것이다. 함수 Years()는 여기에서 정의는 생략되었지만, 주어진 Birth\_date로부터 Age를 계산한다. Entity Female은 Person의 Subtype이다. Subtype은 Supertype의 모든 Attribute를 상속받는다. Female의 Constraint W1은 Female이 Maiden\_name을 가지고 있다면 Husband가 있어야 한다는 제약조건을 명시한 것이다.

## 3. 데이터베이스 환경

인하대학교에서는 통합 생산 시스템의 구현을 위하여 Fig.1과 같은 환경을 구축하고 있다. 이 시스템은 미국의 표준기술국인 NIST에서 개발중인 STEP Tool들을 이용하여 개발되고 있다. 이중 Fedex는 EXPRESS의 컴파일러로서 EXPRESS로 표현된 문법을 검사하고 Schema 내용을 내부 데이터 구조(Working Form)로 변환하여 다른 프로그램이 이를 이용할 수 있도록 한다[8]. 이러한 Fedex의 결과를 이용하여 관계형 데이터베이스 시스템(Oracle)과 객체지향형 데이터베이스 시스템(Obejectivity)에 필요한 테이블과 Class를 각각 정의한다. RDB Interface와 OODB Interface는 이렇게 정의된 테이블과 Class에 데이터의 저장과 인출을 수행한다. Fig.1에 표시된 각 모듈

의 기능은 다음과 같다.

- (1) RDB Interface : 관계형 데이터베이스내의 테이블정의와 데이터입력을 수행하고, 저장된 데이터를 SQL 프로그램을 통하여 인출한다. 이 방법은 사용자가 정의된 테이블의 구조를 잘 알아야하기 때문에 불편하며, 이러한 단점은 STEP의 SDAI(Step Data Access Interface)를 통하여 보완될 수 있다.
- (2) OODB Interface : OODB Interface 역시 사용자가 내부 구현을 모르고도 쓸수있는 SDAI를 제공한다. 이 시스템에서는 RDB와 OODB를 모두 연결할 수 있기 때문에 기존의 RDB를 사용한 응용프로그램을 그대로 연결할 수 있고, OODB의 장점을 이용한 응용 프로그램의 개발도 가능하다.
- (3) Data Browser : 이 모듈은 NIST에서 개발하여 일반에 공개한 프로그램(dp)를 모체로하여 개발되었다. 기존 프로그램인 dp는 text 형태의 데이터만 처리한데 비하여 Data Browser는 CAD System과 연결하여 CAD 화면에서 직접 Graphical STEP Data를 Access할 수 있게한다.
- (4) Information System Manager : 이 Module은 전체 정보시스템에 포함된 데이터의 Integrity를 검사하는 역할을 한다. 또한 Security, Transaction Management, Query Optimization 등의 기능을 제공한다.

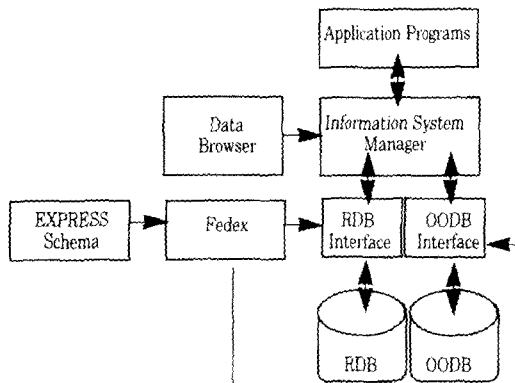


Fig. 1 Software architecture of CIMS

4. 조선공정계획을 위한 통합생산 시스템의 구현

선박의 건조 방법으로는 분할 건조법이 활용되고 있

고 이것에 바탕을 두어 선박의 공정계획은 블럭분할, 탑재, WS(Working Sequence)의 공정으로 크게 분류할 수 있다. 각각의 공정을 위해서 개발된 응용 System이 서로 정보를 공유하기 위해서는 정보에 대한 정확한 표현과 효율적인 검색, 수정, 새로운 정보의 생성이 요구된다. 이 글에서 정의한 통합 생산 시스템은 블럭분할공정과 탑재공정을 고려하여 블럭을 중심으로 정의하였다.

4.1 블럭분할 공정

블럭분할은 초기 설계단계에서 구현된 선각에 대하여 시행한다. 블럭분할 업무는 분할 전문가들의 축적된 지식과 오랜 경험을 바탕으로 과거 실적선의 분할 사례를 바탕으로 초기 분할선을 도출하고 이를 평가하고 수정하여 분할선을 재 도출한다. 작업량, 작업조건, 작업기간, 크레인의 적재효율과 Dock의 용량 등의 제반조건들은 분할선을 결정하는 요소이다. 분할선이 결정되면 블럭분할 시스템은 분할선을 입력받아서 블럭분할을 수행하고 블럭을 생성한다. 분할된 블럭에 대한 정보들은 이들 각각에 대하여 그들 간의 관계를 계층적으로 나타내며 각 블럭별로 형상, 크기, 중량, 무게 중심등의 물리적 특성과 접합관계, 접합길이에 관한 정보를 추출할 수 있다. 이러한 블럭에 대한 정보는 다음에 수행되는 공정인 탑재공정에 중요한 역할을 한다. 블럭 분할 System에서 제공되는 블럭에 관한 정보들은 STEP File로의 Data 변환 과정을 통하여 데이터베이스 시스템에 저장된다.

다음에 도시한 Fig.은 VLCC Type 선박의 일부인 Cargo Holder 중앙평행부 링의 Section을 분할작업을 수행한 예이다.

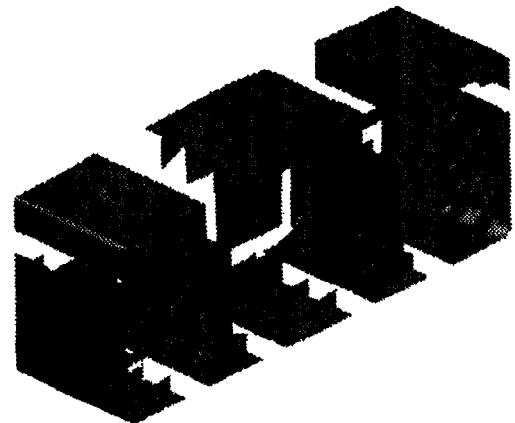


Fig. 2 Ring section of mid ship

## 4.2 탑재 공정

탑재 공정은 블럭 분할 시스템에서 제공된 블럭에 관한 정보를 바탕으로 그 위상관계와 블럭의 물리량에 관한 분석을 통해 탑재 작업의 기술적인 제약 사항을 고려하여 탑재 Sequence를 결정하고 선공정의 부하 분포관점에서 이를 평가하여 최적의 탑재 공정계획을 수립한다. 블럭들간의 위상 정보 즉 크기, 위치값들을 이용하여 탑재 우선 순위를 규정하는 몇가지 조건들을 고려하여 탑재 Activity를 결정하고 블럭의 물리량 즉 무게중심, 중량, 블럭의 접속관계, 블럭의 안정조 특성에 관한 정보를 이용하여 피치타임과 탑재순서의 제약 조건을 생성하고 이 조건을 고려하여 불필요한 Activity를 제거한다. 블럭분할 시스템에서 각각의 블럭의 물리량이 제공되기 때문에 탑재 Activity가 생성되면 선공정의 부하분포를 분석하여 최적의 부하분포를 유도하고 부하변동에 능동적으로 의사결정을 할 수 있다. 최종적인 탑재 Sequence를 결정할 수 있다.

## 5. 정보 모델의 EXPRESS표현

블럭 분할시스템과 탑재 시스템이 공유하는 정보들을 표현하기 위하여 구성한 Ship Schema에서 중요한 의미를 갖는 몇가지 객체에 대하여 설명하면 다음과 같다.

### (1) Block

블럭분할 후에 생성된 블럭을 표현한 객체로 Block은 고유한 Id, Block의 물리적 특성들 그리고, 6방향(상, 하, 좌, 우, 선수, 선미)으로 접한 Block들, 접합하는 Block간의 용접 등을 정의한다.

### (2) Weld

Weld는 블럭 간의 접합관계에서 서로 접합하는 두개의 블럭간의 용접 관한 정보 즉, 용접 길이와 용접 형식과 접합되는 두개 블럭을 정의하는 객체이다.

### (3) Weld\_length

블럭 간의 용접 길이를 정의하는 객체이다. 용접 길이는 작업성을 고려하여 직선과 곡선으로 구분하였고 각 선의 형식 별로 용접자세를 고려하여 상, 하, 그리고 수직으로 세분화하였다.

### (4) Physical Property

Physical Property는 각각의 블럭이 갖는 물리적 특성에 관한 정보를 나타내는 객체이다. 물리적 특성은 여기서 Block의 크기, 위치, 중량과 무게 중심점을 정의하였다.

### (5) Activity

Activity는 탑재작업을 나타내며 Activity 번호, Pitch Time, 선탭블럭, 후탑블럭으로 구성된다.

### (6) Activity Sequence

Activity Sequence는 탑재 순서를 나타내며 Activity의 Set으로 구성한다.

위에서 설명한 공정계획을 위한 정보 모델을 표현하기 위하여 EXPRESS 언어로 하나의 Schema를 구성하였다.

```

SCHEMA ship_schema:
ENTITY point:
  x,y,z : REAL;
END_ENTITY;
TYPE super_block_form = ENUMERATION
  OF(after_peak_form,engine_room_form,
  fore_peak_form, cargo_area_form,
  accommodation_form);
END_TYPE;
ENTITY block:
  block_id : STRING;
  physic_property : physical_property;
  (* 블럭의 물리량 *)
  weld_list : LIST[1:?] OF weld;
  (* 용접관계 *)
  upper_block, lower_block,
  left_block, right_block,
  head_block, tail_block
: OPTIONAL SET[0:?] OF block :
  (* 6방향의 인접 블럭 *)
  sub_blocks : OPTIONAL LIST[0:?]
  OF block: (* 포함된 블럭 *)
  block_form : super_block_form;
  (* 블럭의 형태상의 분류 *)
END_ENTITY;
ENTITY weld_length:
  upper.lower.vertical : OPTIONAL REAL;
  (* 상, 하, 수직의 직선 용접길이 *)
  c_upper, c_lower, c_vertical
: OPTIONAL REAL: (* 곡선 용접길이 *)
  length_unit : length_units;
END_ENTITY;
ENTITY weld:
  weld_form : weld_type: (* 용접형식 *),
  weld_lengths : weld_length: (* 용접길이 *)

```

```

INVERSE
weld_blocks : SET[2:2] OF block
FOR weld_list: (* 용접하는 두개의 블록 *)
END_ENTITY:
TYPE weld_type = STRING :
END_TYPE:
ENTITY physical_property:
max_size : size: (* 블록의 최대 크기 *)
position, center_of_gravity
: point: (* 블록의 위치, 무게중심 *)
weight : REAL:
weight_unit : weight_units:
END_ENTITY:
ENTITY size:
width,height,depth : REAL:
size_unit : length_units:
END_ENTITY:
TYPE length_units = ENUMERATION
OF(M, mm):
END_TYPE:
TYPE weight_units = ENUMERATION

```

```

OF(TON.kg):
END_TYPE:
ENTITY activity:
activity_no : INTEGER:
pitch_time : INTEGER:
start_block : block: (* 선탭 블록 *)
end_block : block: (* 후타 블록 *)
END_ENTITY:
ENTITY activity_sequence:
activities : LIST[1:?] OF activity:
(* 탑재순서 *)
END_ENTITY:
END_SCHEMA:

```

Fig.3은 정보 모델을 EXPRESS-G로 표현한 Diagram이다.

EXPRESS-G는 EXPRESS로 표현된 정보 모델을 간결한 그래픽으로 표시하는 방법이다.

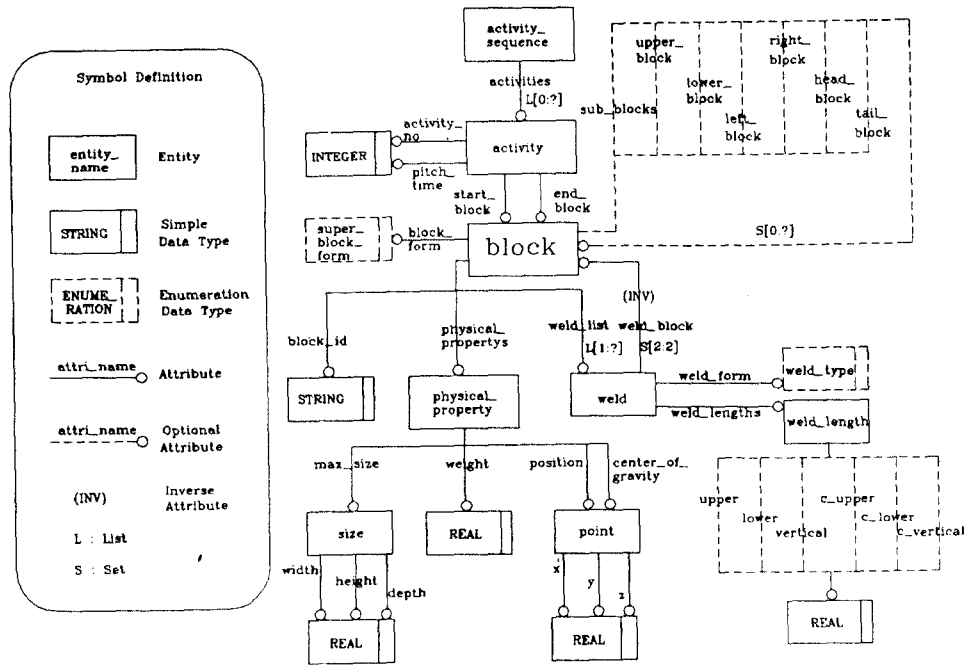


Fig.3 Express-g diagram

6. 시스템 통합의 예

블럭 분할시스템에서 블럭간의 위상정보와 블럭들의 물리적 특성치들에 관한 정보는 탑재시스템의 입력자료가 되어 탑재 시스템이 Activity를 결정하는데 중요한 자료가 된다. 본 절에서는 관계형 데이터베이스 시스템인 Oracle을 이용하여 블럭분할 시스템과 탑재 시스템의 정보교환의 예를 Fig. 2에서 도시한 VLCC Type 선박의 Cargo Holder 중앙평행부의 링의 데이터를 이용하여 설명한다. 이 두 시스템간의 정보를 공유하기 위하여 다음과 같은 일련의 과정을 거친다.

- (1) EXPRESS로 표시된 스키마를 SQL로 전환하여 필요한 테이블을 정의한다.
- (2) 블럭분할 시스템의 출력정보를 데이터베이스에 저장한다.
- (3) 탑재 시스템에 필요한 입력 정보를 데이터베이스에서 인출한다.
- (4) 탑재 시스템의 결과인 탑재 Activity와 탑재순서를 데이터베이스에 저장한다.

본 예에서 설명한 데이터의 흐름은 Fig. 4와 같고 데이터베이스에 저장된 테이블의 일부는 다음과 같다.

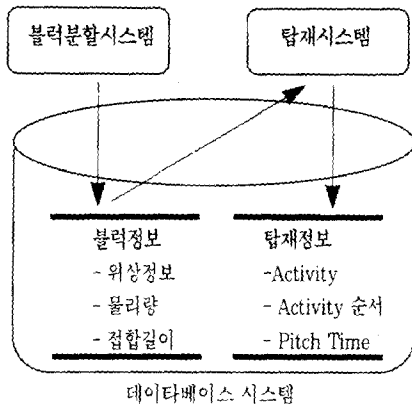


Fig. 4 Information flow using database

- (1) 블럭 테이블의 정의와 저장된 데이터

Name	Null?	Type
ENTITY_ID		NUMBER(38)
BLOCK_ID	NOT NULL	CHAR(24)
PHYSIC_PROPERTY	NOT NULL	NUMBER(38)
BLOCK_FORM	NOT NULL	NUMBER(38)

ENTITY_ID	BLOCK_ID	PHYSIC_PROPERTY	BLOCK_FORM
1	205	2	4
6	206	7	4
11	403	12	4
16	404	13	4
21	7e0	22	4
26	7f0	27	4
29	7g0	30	4
34	7h0	35	4
37	6e0	38	4
42	6d0	43	4
46	5e0	47	4
51	5f0	52	4

- (2) 블럭의 물리량 테이블의 정의와 저장된 데이터

Name	Null?	Type
ENTITY_ID		NUMBER(38)
MAX_SIZE	NOT NULL	NUMBER(38)
POSITION	NOT NULL	NUMBER(38)
CENTER_OF_GRAVITY	NOT NULL	NUMBER(38)
WEIGHT	NOT NULL	NUMBER
WEIGHT_UNIT	NOT NULL	NUMBER(38)

ENTITY_ID	MAX_SIZE	POSITION	CENTER_GRAVITY	WEIGHT	WEIGHT_UNIT
2	3	4	5	192	0
7	8	9	10	165	0
12	13	14	15	142	0
17	18	19	20	32	0
22	23	25	24	439	0
27	23	28	24	439	0
30	31	32	33	332	0
38	39	40	41	539	0
43	39	45	44	539	0
47	48	49	50	555	0
52	48	53	54	532	0

(3) 접합관계 테이블의 정의와 저장된 데이터

Name	Null?	Type
ENTITY_ID		NUMBER(38)
WELD_FORM	NOT NULL	CHAR(24)
WELD_LENGTHS	NOT NULL	NUMBER(38)

ENTITY_ID	WELD_FORM	WELD_LENGTHS
55	arc	56
57	arc	58
59	arc	60
91	arc	90
92	arc	93
94	arc	93

(4) 탑재의 Activity 테이블의 정의와 저장된 데이터

Name	Null?	Type
ENTITY_ID		NUMBER(38)
ACTIVITY_NO	NOT NULL	NUMBER(38)
PITCH_TIME	NOT NULL	NUMBER(38)
START_BLOCK	NOT NULL	NUMBER(38)
END_BLOCK	NOT NULL	NUMBER(38)

ENTITY_ID	ACTIVITY_NO	PITCH_TIME	START_BLOCK	END_BLOCK
95	1	1	1	6
96	2	1	6	11
97	3	1	11	16
98	4	2	16	21
99	5	2	21	26
100	6	2	26	29
101	7	2	29	34
102	8	3	34	37
103	9	3	37	42
104	10	3	42	46
105	11	3	46	51

(5) 탑재 순서 테이블의 정의와 저장된 데이터

Name	Null?	Type
ENTITY_ID	NOT NULL	NUMBER(38)
ELELMENT_ID0	NOT NULL	NUMBER(38)
REF_ENTITY_ID	NOT NULL	NUMBER(38)

ENTITY_ID	ELELMENT_ID	REF_ENTITY_ID
106	0	95
106	1	96
106	2	97
106	3	98
106	4	99
106	5	100
106	6	101
106	7	102
106	8	103
106	9	104
106	10	105

7. 결론

본 연구에서는 조선산업의 공정계획중 불력분할 시스템과 탑재시스템의 정보교환을 STEP을 통하여 구현하였다. 이와같이 제한된 범위의 응용분야에 적용하였기 때문에 현재 STEP에서 개발중인 3차원 형상이나 Ship Structure의 정의는 포함되지 않았다. 이는 본 Prototype의 대상범위를 조선의 전 생산 과정으로 확장하고 STEP의 개발이 진전됨에 따라 포함되어야 할것이다. 데이터베이스 접속도 현재는 관계형 데이터베이스 시스템의 언어인 SQL을 사용하고 있으나 향후 STEP의 표준 접속 방법인 SDAI[9]의 구현이 필요하다.

본 논문에서 설명한 Prototype의 대상을 전 생산 과정으로 확장하기 위한 절차는 다음과 같이 요약된다.

- (1) 통합 생산 시스템에 포함될 대상 작업과 관련 응용 프로그램 선정한다.
- (2) 이들 응용 프로그램간에 서로 교환할 데이터를 EXPRESS로 표현한다. 이때, STEP에서 개발되었거나 개발중인 정보모델을 이용한다.
- (3) 사용할 데이터베이스 시스템을 선정한다.



- (4) 서로 관련된 응용 프로그램에 통합 데이터베이스 시스템과 Interface하는 기능을 추가한다.
- (5) 전체 시스템의 개발과 테스트를 수행한다.

#### 참 고 문 헌

- [1] 강원수, 서승완, 이규열, "객체지향 선박구획 정의 표현 방법론", 1992년도 추계연구발표 논문집, 대한조선학회, pp121-132.
- [2] 윤덕영, 서홍원, "선박의 설계 및 생산시스템 통합을 위한 제품모델 개발", 1993년도 춘계 연구발표 논문집, 대한조선학회, pp12-17.
- [3] 일본 SHIP & OCEAN 재단, "1992년도 CIMS FRAME MODEL 연구개발 보고서", 1993.3.
- [4] Howard Mason, "Part 1 : Overview and Fundamental Principles", NIST, 1992.9.
- [5] Peter Wilson, "Third Annual EXPRESS User's Group(EUG93)", 1993.10.
- [6] NIST, "ISO10303 Industrial Automation - Product Data Representation and Format Description - Part102 Ship Structures", 1989.
- [7] Philip Spidy, "EXPRESS Language Manual", ISOTC/184/WG5 N151, 1992.
- [8] Clark, S.N., "The NIST Working Form for STEP", NISTR4351, NIST, 1990.
- [9] James Fowler, "ISO Industrial Automation - Product Data Representation and Format Description - Part22 STEP Data Interface Specification", NIST, 1992.9.
- [10] Oracle Co., "SQL Designer's Reference Ver3.0", 1990.
- [11] OBjectivity Inc., "Objectivity/DB documents Ver2.0 - C++ Interface Reference", 1992.9.