The Journal of Korean
Institute of CALS/EC
Vol. 1, No. 2, Dec. 1996

STEP 데이터의 활용 방안에 대한 연구

예노경*, 박정선**

A Study on the Practical Usage of STEP data

Do-Keong Yhe, Jung-Sun Park

Abstract

It is accomplished on the practical usage for STEP in enterprise. But they has feeble grasp of STEP, therfore it is hard to apply. That is due to take a negative policy with the lack of advertising and understanding, most of studies are going on a studentlike attitude rather than the practical usage or implementation, so they don't know exactly what is STEP, and how can they apply to their own enterprise.

This paper aimed to present more detail understanding of STEP usage and practical usage of STEP data in enterprise. We describe EXPRESS that is description method of STEP, four ways of implementation method, and introduce to using ST-Developer and ST-Oracle by STEP Tools Inc. for more detail understand of STEP. Thereafter we propose practical usage of STEP with CAD systems and PDM systems on the knowledge of previous study, and propose total system implementation image on STEP data in enterprise, finally discuss conclusion and futher study issues.

* (주)LG-EDS시스템
** 명지대학교 산업공학과 교수

1. 서론

CALS에서 추구하는 공유 데이터베이스(shared database)로의 과도기적 방안으로 제시되고 있는 각종 표준들은 CALS를 구현하기 위한 여러 인프라들과 함께 매우 중요한 이슈이다. 그 중에서도 제조와 관련된 핵심표준이라 하면 단연 STEP을 들 수 있을 것이며, 이는 현재 SGML(Standard Generalized Markup Language), EDI (Electronic Data Interchange) 등과 함께 활발히 연구되고 있다[김천환 김규수, 95; 황한웅, 96; STEP연구회, 96].

기업내에서의 STEP에 대한 시각은 그 활용에 대한 것으로서, 사용자 측면에서의 접근을 통한 STEP 구현에 있다. 이를 위해, STEP에 대한 보다 실제적인 이해를 위하여 STEP의 개략적인 의미가 아닌 그 내부를 들여다 볼 필요가 있으며, STEP을 통한 올바른 자사의 정보표현 및 교환을 위해서 단순 사용자 수준이 아닌 개발된 AP(Application Protocol)의 도입에 있어 일부 변경 내지는 재편성(refining)이 가능한 수준에서의 접근이 필요하다. 이에 STEP의 표현방법과 구현방법에 대한 이해와 상용화된 개발 툴을 통해 쉽게 접근함이 필요하며, 구현 이미지와 도입효과에 대한 예측이 필요하다.

본 논문은 STEP의 실제적인 이해와 기업내에서의 STEP data 활용 방안을 제시하기 위해 우선, STEP의 표현방법인 EXPRESS언어와 구현방법에 대해 소개하고, 상용 Tool인 STEPTools사의 ST-Developer와 ST-Oracle의 활용방법을 소개한다. 그리고, 이들을 통해 습득한 지식을 바탕으로 보다 실제적인 STEPdata의 활용방안을 CAD시스템과 PDM시스템에 대하여 제시하며, 기업내 STEP data를 통한 전체 시스템 구축 이미지를 제안하였고, 끝으로 결론 및 추후연구과제에 대해 제시하였다.

2. 표현방법과 구현방법

2.1 표현방법

표현방법에 관련된 파트는 모두 4가지이며, 최근 EXPRESS-M(Mapping)과 EXPRESS-V (View)의 기능을 결합한 EXPRESS-X가 더불어 추진되고 있는 실정이다. 이들은 각각 다음과 같으며, 이들은 현재 ISO/TC184/SC4 아래의 WG11에서 추진되고 있다.

<표1> 표현방법 관련 파트

No.	Status	Title	Project Leader
1	I	Overview and fundamental principles	Howard Mason
11	I	The EXPRESS language reference manual	Phil Spiby
12	C	The EXPRESS I language reference manual	Peter Wilson
13	W	STEP Architecture & Development Methodology	Julian Fowler

(Legend Status :
1 = Publication Stg.
(Int'l Standard approved & published)
C - Committee Stg.
(CD circulation -> PDIS regis)
W - Preparatory Stg.
(Working Draft devel -> CD regis.))

이들 중 Part 1은 개관이고 Part 13은 개발측면에서 유효한 것이므로, EXPRESS와 그 시각적 표현기법인 EXPRESS-G, 그리고 EXPRESS I에 대해서만 간단히 살펴보고, 끝으로 최근 이슈가 되고있는 EXPRESS-X에 대해 소개한다.

2.1.1 EXPRESS

EXPRESS는 STEP Part들에서 필요한 정보들의 표현방법에 대한 모델링 언어로서, 일반화와 제약조건을 표현할 수 있으며 엔티티-속성-관계(entity-attribute-relationship) 모델에 기초한 정보 모델링 언어이다[13]. STEP에서 개발되는 모든 공통자원과 응용 프로토콜은 EXPRESS로 표현된 정보 모델을 반드시 포함하여야 한다.

EXPRESS는 7개의 선언 구조체를 가지고 있다[STEP Part 11, 92].

- 스키마(schema) : 대상 분야 정의
- 타입(type) : 데이터 형태 정의
- 엔티티(entity) : 데이터의 표현 및 관계 정의
- 상수(Constant) : 불변하는 값의 정의
- 함수(Function) : 결과값을 반환하는 알고리즘
- 프로시저(Procedure) : 결과값을 반환하지 않는 알고리즘
- 규칙(Rule) : 제약 조건 정의

2.1.2 EXPRESS-G

EXPRESS-G는 EXPRESS의 공식적인 그래픽 표현방법이다. 이것은 Part11의 공식적인 부분에 정의되어 있으며, 사용자간의 의사전달에 이용된다. EXPRESS-G의 구조체는 다음과 같다.

- 스키마와 스키마간의 연결
- 엔티티와 서브타입 또는 슈퍼타입
- 엔티티간의 관계와 차수
- 타입
- 다른페이지와의 연결

EXPRESS-G는 제약조건의 정의는 지원하지 않는다. EXPRESS-G의 표현방식과 예는 그림 1과 같다. 또한, 그림 2의 EXPRESS 스키마를 EXPRESS G로 표현하면 그림 3과 같다[Steven Ray, 96].

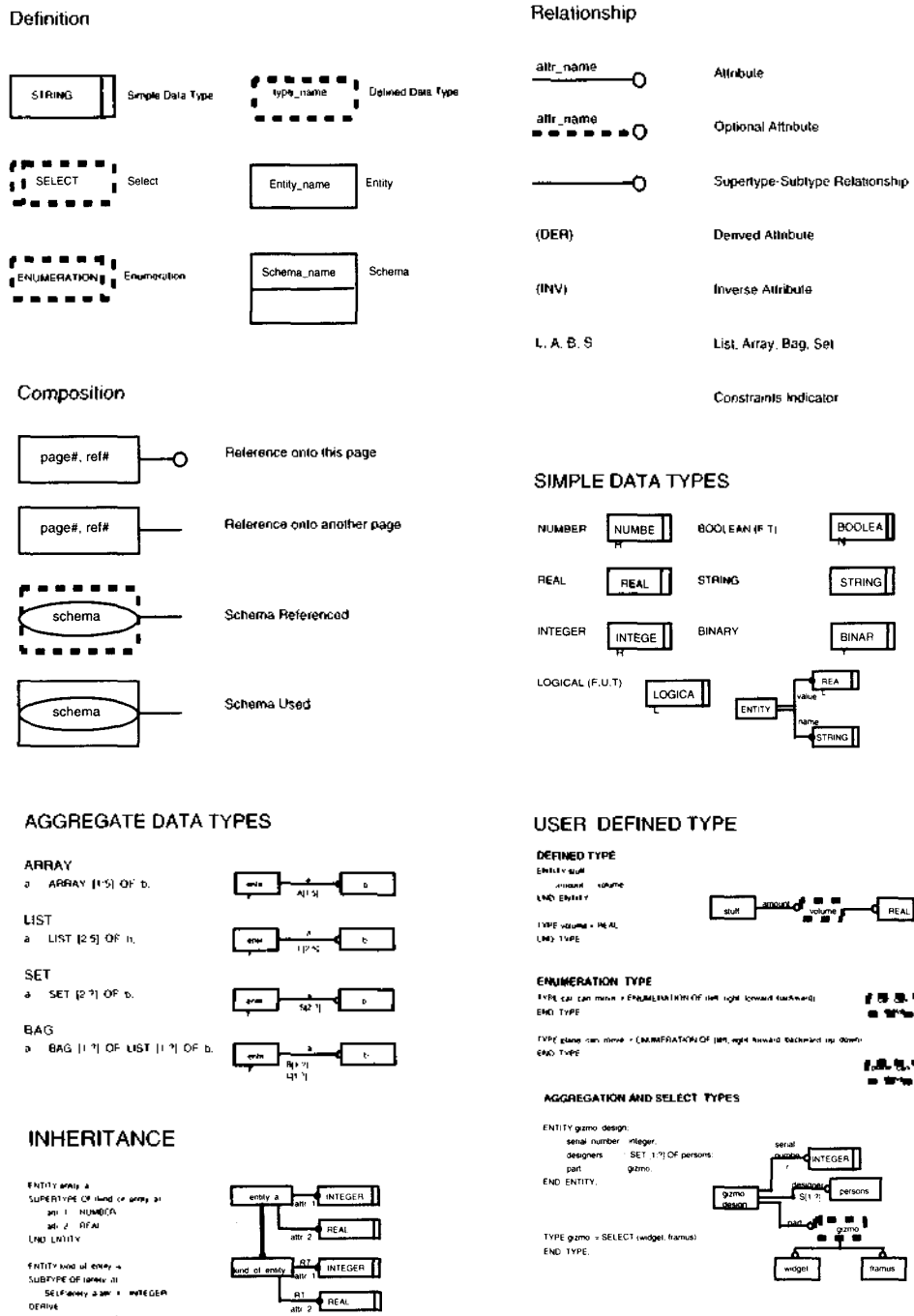
2.1.3 EXPRESS-I

EXPRESS-I는 인스턴스의 속성 값을 표현하는 언어로서, EXPRESS로 정의된 스키마의 전체 또는 일부를 적용한 인스턴스의 속성값을 표현한다[STEP 연구회, 96].

EXPRESS-I는 다음 2가지 경우에 주로 사용된다.

- 다음 각 수준에서의 인스턴스 표현
  - 엔티티
  - 스키마
  - 여러 스키마의 집합(즉, model)
- 적합성 테스트에서 사용될 테스트 데이터의 표현
  - 상황(context) : 파라메터와 기본값을 정의
  - 테스트 사례(test cases) : 한 개 이상의 상황들로 구성

EXPRESS 스키마 또는 데이터 타입과 EXPRESS-I 인스턴스간의 관계도 정



<그림 1> EXPRESS-G의 표현방식과 예제

```
SCHEMA organization;

ENTITY employee
SUPERTYPE of (manager);
 employee_name                    :                person_name;
 employee_number                  :                INTEGER;
 salary          :                amount_of_money;
 date_of_birth   :                date;
 citizen_of                       :                country_name;
DERIVE
 us_citizenship_status            :  BOOLEAN : = (SELF.citizen_of = 'USA');
UNIQUE
 UR1:            employee_number;
WHERE
 WR1:            SELF.salary.value > 0;
END_ENTITY;

TYPE amount_of_money: NUMBER;
END_TYPE;
TYPE date = STRING;
END_TYPE;

TYPE country_name = STRING;
END_TYPE;

ENTITY person_name;
 last_name       :                STRING;
 first_name                       :                STRING;
 middle_name     :                OPTIONAL STRING;
END_ENTITY;

ENTITY manager
SUBTYPE OF (employee);
 supervisor_to : SET [1:?] OF employee;
WHERE
 WR1: SIZEOF (QUERY (x <  SELF.supervisor_to |
    manager IN TYPEOF (x))) = 0;
END_ENTITY;

END_SCHEMA;
```

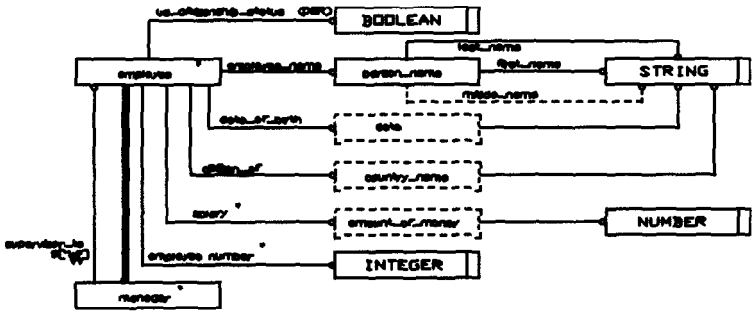
<그림 2> EXPRESS Schema 예제

의된다. EXPRESS-I는 구현방법이 아닌 정보의 표현방법으로 분류되어 Part12에 정의되어 있다.

2.1.4 EXPRESS-X

EXPRESS-X language는 ISO 10303의 official한 Part는 아니지만, EXPRESS-V language(ISO/TC184/SC4/WG5 N251)와 EXPRESS-M language(ISO/TC184/SC4/WG5 N243)의 조합으로서, 표현 방법에 관련된 분야로서 볼 수 있다.

STEP과 같은 제품 데이터에 대한 표현 및 교환 표준은 반드시 완전하고 모호하지 않아야 한다. 이런 이유로, STEP은 매우 방대하며, 많은 독립적 개별 시스템에

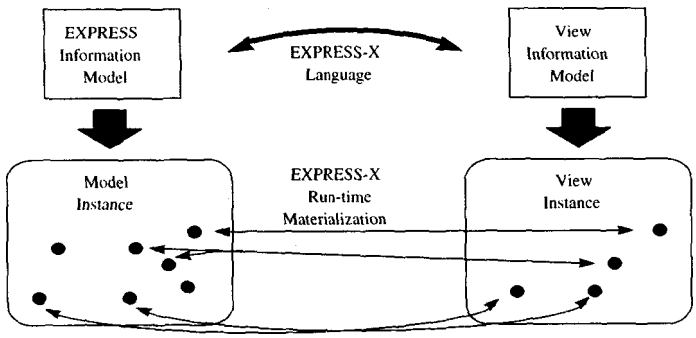


<그림 3> 그림 2의 EXPRESS-G 표현

서는 필요없을 자세한 내용들을 담고 있다. 다시 말해서, 이들 응용 시스템들의 요구의 합집합이라 정의 할 수 있다. 이에 모델의 불필요한 자세한 내용을 생략한 제품 모델의 단순화 된 view를 통해 많은 응용 시스템을 수용한다. 이와같이 단순화 된 view를 사용하는 것은, 이러한 view가 개념적으로 이해하기에 쉽고 응용 시스템내에서 프로세스되기 때문에 이들 응용 시스템에 대해 요구되어진다. 특히 legacy system에 대해서는 더욱 그러하다.

불행하게도, 최적으로 단순화된 하나의 제품모델의 view가, 심지어 두 시스템이 연관되어있다 하더라도, 다른 응용 시스템에서 반드시 최적은 아니다. 이러한 이유 때문에, 각 응용 시스템에 맞춘 제품 모델의 view를 쉽게 생성할 수 있도록 하려는 요구가 대두되었다.

STEP에서 제품 모델은 EXPRESS에 의해 정의되며, 이는 제품 모델의 view가 반드시 제품 모델의 EXPRESS 정의에 기초해야 함을 의미한다. 그러므로, STEP에 대해 language는 EXPRESS정보 모델 view의 정의와 밀접하게 요구된다. 이것이 EXPRESS-X의 목적이다. 이것은 EXPRESS 정보 모델의 단순화 된 view의 정의에 대한 구성을 포함한 EXPRESS의 확장된 형태이다. 결국, EXPRESS-X



<그림 4> EXPRESS-X Overview

language의 목적은, <그림 4>에서 보는 바와 같이 EXPRESS에 정의된 정보 모델들 사이의 매핑을 정의하기 위함이다.

EXPRESS-X는, 하나의 EXPRESS schema가 또다른 한가지 abstract view를 나타낼때, 한 쌍의 EXPRESS schema들 사이의 매핑을 정의하는데 주로 활용된다. 예를 들어, EXPRESS-X는 Application Protocol의 AIM으로부터 그것의 ARM으로의 엔티티 매핑을 구현하는데 사용되어진다. EXPRESS-X 언어는 생성되야하는 새로운 view하의condition을 명확화하는 것과 어떻게 새로운 view 엔티티에 대해 도출되어야하는지에 의해서 이들 매핑을 통제한다. EXPRESS-X language는 EXPRESS schema들 사이의 two-way mapping의 정의를 지원하기 위해서 확장된 것이다.

EXPRESS X에 대한 Document는 현재 Working Draft Status에 있고, 이외에도 EXPRESS-C(Conceptual)와 EXPRESS-P (Process) 등이 WG11에서 함께 제시되고 있다.

2.2 구현방법

STEP에서는 기존의 물리적 파일을 통한 구현이 응용분야의 요구사항과 물리적 파일을 이용한 구현방법상의 제약을 구분하지 못하여, 파일내에서 찾으려는 특정 정보의 위치나 인덱의 규칙을 알아내기 위해서도 그 파일을 읽어야만 하는 문제점을 해결하기 위하여, 물리적 파일 구조가 응용분야에 대한 지식과 분리되어 정의될 수 있도록 구현방법에 대한 클래스를 두었다.

구현방법에 관련된 파트는 현재 모두 5가지이며, 파트 25의 Late FORTRAN은 개발이 중단된 상태이다. 이들은 아래와 같은 4가지 데이터 교환방식에 맞춰 제시된 것이다[STEP Part 21, 94].

- 물리적 파일, 즉 텍스트 파일
- 능동적 파일 교환 API(Application Programmers Interface)
- 공유 데이터베이스 관계형,망형, 객체지향DBMS기술 적용
- 지능형 지식 기반 시스템

<표2> 구현방법 관련 파트

No.	Status	Title	Project Leader
21	I	Physical File, Exchange Structure Working Format, Active Transfer	Jan van Maanen
22	C	Standard Data Access Interface	Jan van Maanen
23	C	Early C++ (binding for #22)	Tom Rando
24	C	Late C (binding for #22)	Dave Price
26	W	IDL (binding for #22)	Tom Rando

제시된 4가지의 방법에 대하여 간단히 설명하기로 한다.

2.2.1 물리적인 파일에 의한 방법

물리적인 파일에 의한 방법은 수동적 파일 교환(passive file exchange)을 통하여 구현될 수 있으며, 이는 한 Application에서 생성된 데이터가 STEP Translator를 통해 STEP file format으로 변환되는 후처리과정과 STEP file format을 또 다른

Application의 STEP Trans - lator로 변환하는 전처리과정을 통해 이루어진다. 이와 관계된 표준은 Part 21 : Clear text encoding or the exchange structure에 해당한다[STEP Par 21, 94].

2.2.2 능동적 파일 교환 API(Application Programmers Interface)

이는 특정 프로그래밍 언어와의 바인딩 규약에 따라 엔터티의 생성, 질의, 삭제, 수정하는 함수들의 집합으로 이루어진다. 따라서, 응용 시스템에서 정의한 정보구조에 따라 인스턴스를 저장하는 데이터 저장소(여기서의 저장소가 반드시 지속적이어야 하는 것은 아니다. 즉, working form형태가 가능하다.)가 있다고 가정하면, 사용자는 이 API를 통해 데이터 저장소에 접근한다. SDAI도 하나의 API로서 볼 수 있으며, 파트 22에서 SDAI에 대하여, 그리고 파트 23, 24에서 C++과 C를 이용한 바인딩에 대해 각각 기술하고 있다[STEP Part 22, 94; Steven Ray, 96].

2.2.3 공유 데이터베이스

- 관계형, 망형, 객체지향형 DBMS

이 방법은 데이터 저장소(persistent data repository)에 대한 동시 접근을 가능하게 함으로써 데이터를 공유하게 하는 것으로서 여러형태의 DBMS와 분산 데이터베이스 형태도 가능하다. STEP 스키마와 인스턴스들을 각각의 DB 스키마에 맞게 매핑시켜주는 작업이 필요하다. 물론 여기서도 SDAI와 C++, C를 이용한 바인딩이 참조된다[STEP Part 21, 94; STEP Part 22, 94].

2.2.4 이종 분산환경 지원 시스템

둘 이상의 상이한 프로세스들간의 커뮤니케이션에 있어서 persistent 저장소가 제품 데이터와 이를 사용하기위한 프로세스와 관련된 지식 둘 다를 지원하기위한 접근 메카니즘 상에서 데이터가 교환된다. 이는 Part 26에서 CORBA의 표준 정의언어인 IDL을 통해 제안되고 있는 단계로 분산환경과 연계되어 연구되고 있다[STEP Part 21, 94; STEP Part 26, 95].

3. 상용 STEP Tools의 고찰

STEP을 위한 상용 Tool들은 현재 수십여종이 나와 있으며, <그림5>와 같다.

Bell Atlantic/CERC
- expressToLaser
- laserToExpress
- pffToLaser

CADDETC
- DECexpress Starterkit
- EXPRESSmaster
- Code Generators
- EXPRESS-G Generator

Cadlab - Paderborn University
Canonical Systems, Inc
CSTB
Data Control & Management
Digital Equipment Corporation
- DECexpress Starterkit
- DECexpress Graphical Editor

EPM Consultants a.s
- EDMD
- EDMI
- EDMS
- EDM system platforms

Frauhofer IGD
General Electric
GIDA
- EXEP Tool Kit
- EFE

GOSET
- EXPRESS Compiler
- Model Data Base Manager
- Short to Long Form Translator
- Application Protocol Analyser

GOPAS
- ExpressLab
- Interface Modules

Honeywell
ITI
University of Kaiser lautern
Loughborough University
Manchester University
Newcaster University

NIST
- EXPRESS Toolkit
- Pretty Printing
- Part 21 File Exchange
- Class Library
- Shtolo
- Data Probe
- Transformr
- SEXE
- October 1993 Release
  - Environment
  - How to Obtain the Toolkit
  - Questions, Problems, and Support

National Research Council
PDES Inc
Product Data Integration Technologie, Inc.
- FirstSTEP XG
- FirstSTEP Modeler's Toolkit

Rensselaer Polytechnic Toolkit
Rutherford Appleton Laboratory
- EXPRESS Compiler
- Physical File Reader and Checker

SDRC

Senter for Industiforskning
- Coversion Tools Library

Southbank University
STEP Center
STEP Tools, Inc
- EXPRESS Modeling Tool Kit
- ST-EXPRESS Developers Tool Kit
  - Information Model Dev.
  - Application Development
  - System Integration
  - STEP Conformance Testing
- STEP ACIS Husk
- STEP Visualizer
- STEP Database Adapter
- STEP Tools AP203 Database Manager

TNO
- EXPRESS Compiler : xpress
- STEP Physical File Package
- STEP Physical File Reader
- STEP Physical File Writer
- EXPRESS Comment Stripper
- Public Release
- CCLib and CCgen

University of Trondheim
Versant Object Technology
ZGDV Rostock

<그림5> STEP Tool 취급사 및 상품명 일람

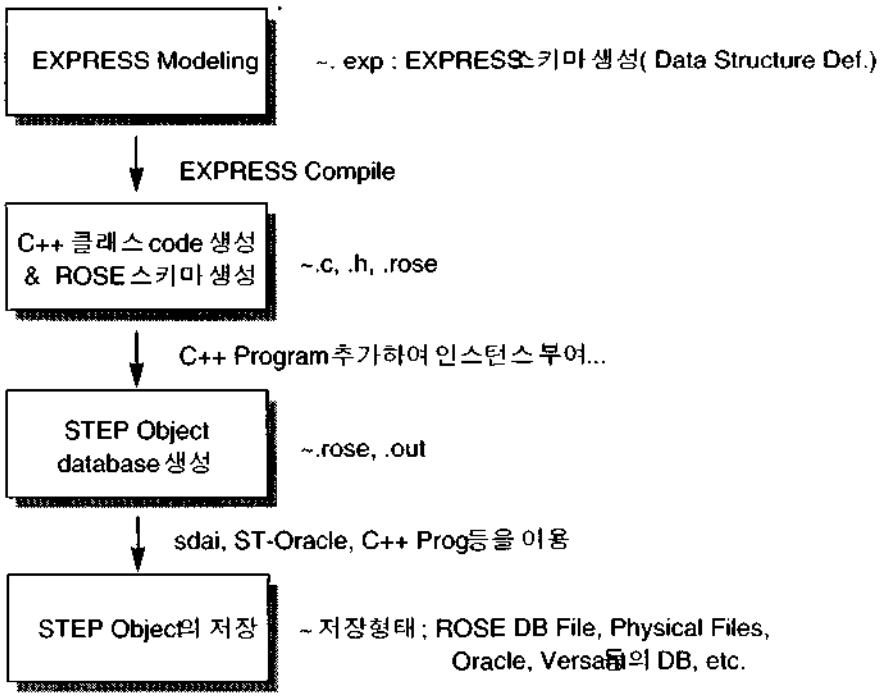
본 연구에서는 상용 STEP Tool들 중 가장 널리 쓰이고 있는 STEP Tools사의 제품을 통해 그 활용방법을 살펴보기로 하고, 그들 중에서도 STEP data의 활용에 있어 핵심적인 ST-Developer와 ST-Oracle에 대해 살펴봄으로써 상용Tool의 활용방안을 모색한다.

3.1 ST-Developer

ST-Developer는 EXPRESS의 어플리케이션을 생성하는 CASE 도구로서, EXPRESS 컴파일러, EXPRESS 해석기, 프로그래밍 라이브러리 및 STEP 물리적 파일 지원 등의 기능들로 구성되어 있다[STEP Tools Inc, 96].

ST-Developer는 통합된 도구로서 정보 모델링 작업자와 시스템 설계자가 EXPRESS기반의 정보모델을 개발하고 이러한 EXPRESS기반의 정보모델을 구현하는데 있어 지원한다. EXPRESS기반의 정보모델 개발지원 도구에는 EXPRESS 컴파일러와 EXPRESS-G 관련 도구들이 있으며, 구현을 위한 프로그래밍 지원 도구에서는 EXPRESS정의 인스턴스의 조작기능, C++내의 바인딩 기능, Physical File 및 저장고로서의 binary working형태 지원기능, SDAI지원 기능 등을 제공한다.

ST-Developer를 이용한 스키마의 생성에서부터 저장까지의 활용에 대한 개략적인 순서는 <그림6>과 같다[STEP Tools Inc, 96].



<그림6> ST Developer를 이용한 database의 생성 및 활용

여기서의 ST-Developer의 가장 큰 기능은 라이브러리 제공으로 볼 수 있겠으며, 이를 이용한 C++이나 C, 혹은 각종 프로그래밍들로써 저장형태의 변환 및 저장소의 접근을 통한 조작이 가능하다 할 수 있다. 예를 들어, EXPRESS Model의 Compile을 통한 code 및 스키마 생성 이후의 C++ 프로그래밍을 통한 인스턴스부여는 실제적으로는 CAD에서 생성된 데이터를 가지고 일어나야 하며, 또한 이것이 각 CAD 벤디들에 맞게 구성되어 제품으로 제공된 것이 STEP Translator의 형태로 구현된 것이라 할 수 있다.

이외에도 ST-Developer는 각종 유틸리티들을 제공하여 활용하기 쉽게 제공하고 있다. 위 그림에서는 EXPRESS Compile에

% express2c++

이라는 명령어의 수행을 통해 EXPRESS언어의 C++ code로의 전환이 가능하다.

또한 마지막에서의 Physical file 생성에 있어서도,

% rose format step

이라는 유틸리티를 통해 쉽게 접근할 수 있다.

ST-Developer에서 제공하는 각종 툴들과 유틸리티들을 정리해 보면 다음과 같다[STEP Tools Inc, 96].

- EXPRESS-G Tools

  # expgedit : Interactive한 display와 EXPRESS G diagrams의 edit 기능
  # express2expg : EXPRESS -> EXPRESS-G
  # expgupdate : 기존의 express 변경시 EXPRESS-G diagram의 layout을 유지시킨다.
  # expgdiff : express file의 다른 버전들간의 유사점/차이점을 하이라이트 시킨다.
  # expg2ps : printing하기위한 PostScript file로의 변환
  # expg2hp : plotting하기위한 HP GL file로의 변환
  # expgsetopts : page size, font name, size 그리고 orientation 등을 설정한다.
  (이전의 EXPRESS-G 문서에 유용함)

- EXPRESS Compiler

  # expfront [options] expfile1 [expfile2...]
  # express2c++ [options] expfile1 [expfile 2...] : EXPRESS -> C++ code
  # express2sdai [options] expfile1 [expfile 2...] : EXPRESS -> SDAI

cf.) 여기서 EXPRESS가 C++로 변환되는 과정에서의 type전환은 <표3>과 같다.

<표3> EXPRESS vs. C++

EXPRESS	C++
INTEGER	int
NUMBER REAL	float (default) double (ten or more digits of precision)
STRING	STR (typedef char *)
BINARY	BINARY (C++ class)
BOOLEAN	BOOL (typedef char) (values ROSE_TRUE, ROSE_FALSE)
LOGICAL	LOGICAL (typedef char) (values ROSE_TRUE, ROSE_FALSE, ROSE_UNKNOWN)

- STEP Conformance Editor
  # stepedit

- STEP Conformance Checker
  # apconform [options] data_file
  # xapconform (; X window version)

- IGES To STEP Converter
  # iges2step ; IGES -> STEP
  # step2iges ; STEP -> IGES

- DXF To STEP Converter
  # dxf2step ; DXF -> STEP
  # step2dxf ; STEP -> DXF

- ROSE Database Utilities
  # rose <command> <options>

(commands)

cat ; 디자인 파일의 내용을 출력
conflict ; 디자인 파일들 사이의 갱신 충돌을 검색
create ; 새로운 객체 생성
delete ; 기존의 객체 소멸
diff ; 두 디자인 파일의 비교
format ; 파일 형식의 변경 및 데이터 집합의 병합
help ; 툴 선택사항 정보의 출력
index ; 객체 명명
ls ; 검색 path의 디자인 파일 출력
migrate ; 신규 스키마로의 이주
mutate ; 신규 데이터 타입으로의 변경
reference ; 파일간의 참조에 대한 유효성 검사
sed ; diff의 결과를 이용하여 디자인 파일 갱신
set ; 객체내의 속성 값 설정

C++ Development Tools

extclass ; 클래스를 안전하게 확장시킬 수 있도록, 생성된 C++클래스에 hook를 배치
mkmakefile ; 생성된 클래스들을 위한 makefile을 생성
stepmunch ; C++ static constructor에 대한 호출을 발생한다. 이것은 C++ 클래스와 C, FORTRAN, PL/1 등과 같은 언어로 작성된 코드를 링크할 때 도움이 된다.

3.2 ST-Oracle

ST-Oracle Interface는 RDBMS에서 STEP data를 archive하거나 maintain하기위한 툴이다. 이 Interface는 DDL statement와 RDBMS로부터 STEP file을 load, unload하기위한 code를 생성한다.

이 Interface는 두 가지 유틸리티들의 집합으로 구분지어 볼 수 있다[STEP Tools Inc, 96].

· EXPRESS to RDB Converter ; RDBMS로부터 data를 load, unload하기 위한 툴. 이 툴은 code generator이다. 이는 아래와 같은 code를 생성한다.

- STEP schema를 위한 DDL statement
- RDBMS로 design data를 load하기위한 Code
- RDBMS로부터 design data를 unload 하기위한 Code

이 code는 loading과 unloading을 최적화하기위해 동적 SQL을 사용치 않고 생성된다. 이 툴은 entities, selects, aggregates(List, Set, Bag, Array), 그리고 모든 primitives를 지원한다.

· RDB Tool Kit - RDB Tool은 RDBMS에서 Express2rdb tool에 의해 load된 STEP data를 조작하거나 유지/보수하기위한 유틸리티들의 집합이다. 이 유틸리티는 사용자가 data를 조회하고 조작하기위한 design data의 form내에서, relational database내에 일반화 된 data를 보는 것을 가능케 해준다. 이 유틸리티는 source form내에 분산되어 있기 때문에, 그들은 확장되어질 수 있고 사용자의 특별한 요구에 맞춰 customizing될 수 있다.

3.2.1 EXPRESS TO RDB CONVERTER

EXPRESS to RDB Converter (express2rdb)는 DDL statements와 RDBMS로부터 STEP design을 load /unload하기위한 code를 생성한다. 이 툴은 아래와 같이 사용된다 ;

```
# express2rdb [-h] [-u<usr/pwd>]
  [-n<prog_prefix>] <schema1>...
```

-h	Help message를 보여줌.
-n<prefix>	file name을 생성함에 있어 제공된 prefix를 사용함. Default는 첫 번째로 정의된 schema.

이 툴은 EXPRESS Compiler에 의해 이미 compile된 schema상에서 작동한다 (즉, compile된 ".rose" files 상에서 작동함).

툴을 작동하면, 아래와 같은 4개의 파일을 볼 수 있다.

<schema>.sql
- EXPRESS schema의 DDL.

<schema>_drop.sql
- database로부터 schema table을 drop하기위한 SQL statement.

<schema>_step2rdb.pc
- STEP design을 RDBMS로 loading 하기 위해 생성된 program.

<schema>_rdb2step.pc
- RDBMS로부터 STEP design으로 unloading하기위해 생성된 program.

예를 들어, 아래와 같이 실행시키면,
% express2rdb picture
아래와 같은 파일들이 생성된다.
picture.sql
picture_drop.sql
picture_step2rdb.pc
picture_rdb2step.pc

이 file <schema>.sql은 design에 의해 사용될 RDB DDL schema를 포함한다. 이는 반드시 sqlplus하의 command line상에서 "@<schema>"라고 typing함에 의해 실행된 것으로부터 생성된 것 내에 loaded되어야 한다.

생성된 *.pc files은 $ROSE/demos/oracle 내에서 찾아볼 수 있는 sample Makefile을 사용하여 compil 될 수 있다.

% make SCHEMA=picture step2rdb
% make SCHEMA=picture rdb2step

compile이후에 자신의 디렉토리 밑에서 아래와 같은 것들을 실행시켜야 한다.

picture_step2rdb
: RDBMS내의 picture schema에 의해 정의된 data의 upload를 위한 program
picture_rdb2step
: RDBMS내의 picture schema에 의해 정의된 data를 STEP file로 download를 위한 program

아래와 같이 typing함으로써 upload 혹은 download할 수 있다.

<schema>_step2rdb <design> <usr/pwd>
<schema>_rdb2step <design> <usr/pwd>

RDBMS내에 저장된 data는 sqlplus하의 SQL statement를 이용하여 보여질 수 있다.

3.2.2 THE RELATIONAL DB UTILITIES

RDB툴은 RDBMS에 저장된 STEP designs을 보고 조작하기위한 interface를 제공하는 유틸리티들의 집합이다. 이들 유틸리티들은 source form에서 제공되기 때문에, 개별적인 필요에 맞추기 위해 modify할 수 있다. 툴의 source는 $ROSE/demos/oracle/rdb_tool 밑에서 찾아볼 수 있다.

이 툴은 아래와 같은 syntax로 호출된다.

% rdb [-u <usr/pwd>] <command> [<options>] [<args>]

user-id는 -u option을 사용함으로써 explicitly하게 제공된다. 만일 system이 OS의 user-id를 사용하지 않는다면, 툴을 사용하기 위해서 반드시 유효한 RDBMS user-id를 가져야 한다. 현재 commands는

아래와 같이 표현된다.

· CAT

arguments:: design

ROSE와 같은 format에서 정의된 design을 Display시킨다.

display를 위한 가정은 아래와 같다.

- Strings는 " "에 의해 경계지어진다. Reference 는 < >에 의해 경계지어진다.

· DELETE

argument:: design

database로부터 부분적인 design을 삭제한다.

둘은 reculsively하게 design에 있어서의 모든 object를 삭제한다.

NOTE : 외부참조는 이 버전에서는 다루지 않는다.

· HELP

argument:: command

툴에 있어서의 commands에 대한 help.

· LS

argument:: None

RDBMS내의 모든 design list가 생성된 날짜와 함께 사용자로부터 지장된다.

· RENAME

argument:: design1 design2

RDBMS내에서 design1으로부터 design2로 이름을 바꾼다.

3.2.3 기술적 검토사항

· Identifier Names

대부분의 RDBMS에서는 모든 identifier 이름(tables, columns, etc.)의 길이를 30 characterers로 제한한다. STEP은 그러한 한계가 없고, 다양한 parts내의 많은 entity definitions들이 이 길이를 초과한다. 또한, 많은 RDBMS keywords는 STEP에서의 keyword가 아니다. 위의 두 가지 경우에, relational system에의 저장을 위한 STEP identifier의 이름을 modify할 필요가 생긴다.

아래와 같은 프로시저를 따라 이름을 합리화 시킨다.

- 만일 name이 RDBMS에서의 keyword라면, 이름 앞에 "R "를 붙인다.
- 만일 name이 30 character를 넘어가면 name이 30 character보다 적어질 때까지 마지막 모음이나 반복되는 character를 지운다.
- 만일 name이 여전히 30 character를 초과할 때, name을 잘라낸다.

· Schema of the Relational Data

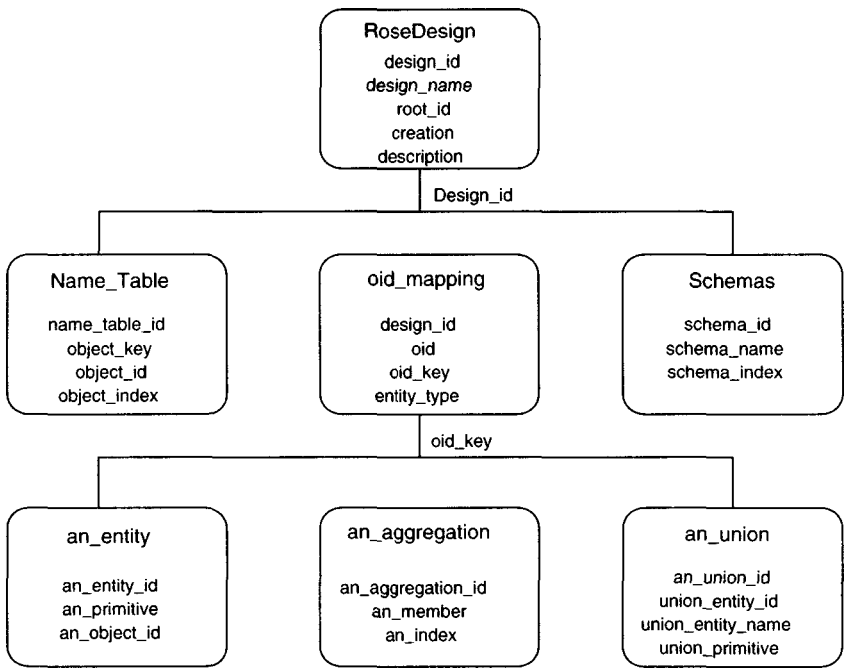
STEP data는 relations안에 저장되기 위해 일반화된다. <그림7>은 STEP relational interface의 schema definition이다.

· DESCRIPTIONS :

RoseDesign

RoseDesign은 hierarchy의 맨 위에 표현되어 있다. 이것은 RDBMS내에 저장된 각각의 design에 대한 단일 entry를 포함한다. RoseDesign에서의 colums는

· Design_id -- Design_id는 system의 생성된 identifier이다. 이는 RoseDesign relation의 primary key이다. Design_id는 design의 다양한 components에서 links maintain하기 위한 foreign key로서 사용된다. 즉, schemas, objects and name_table.
· Design_name -- Design_name은 design의 이름을 포함한다.
· root_id -- 하나의 STEP design은 하나의 root object를 정의할 수 있다. 만일 정의되어 있다면, 이 column이 object의 RDB id를 저장한다.
· creation -- design상의 날짜는 database로 loaded된다. 만일 design이 loaded되지 않았다면, NULL로서 저장된 column인 design을 외부참조로서 표현한다.



<그림7> Schema of the Relational Data

· Description 사용자가 design에 대해 포함시키고자 하는 모든 description.

Name_Table

Name_Table은 design의 contents에 대한 name index로서의 ROSE에 의해 사용되어진 내부객체 디렉토리이다. 이 relation은 RDBMS에서의 index를 저장하기위해 사용되어진다. relation system은 어떤 경우에도 이 index를 사용하지 않는다. 이것은 ROSE와 연계되어 maintain하기위해 저장되어진다.

· name_table_id -- 이 column은 저장되는 nametable design의 design_id를 저장한다.
·object_key 이 key는 ROSE dictionary 내에서 정의된다.
· object_id -- name_table내 object의 RDB id
· object_index -- order를 maintain하기위한 내부 identifier. 따라서, relation은 order의 concept을 갖지 않는 것들의 집합이다.

Schemas

relation은 개별적인 design에서 사용되는 schema들의 name을 포함한다. RDBMS에서 relational interface는 metadata를 저장하지 않는다. ROSE에서 schema를 maintain하기위한 design decision은 아래의 이유들로 결정되어진다.

여러 사용자들로부터 동시에 modify되어질 수 없다. STEP에서의 schema 정보는 대부분 매우 복잡하며, RDBMS에서의 정보로 저장되는데 있어 유용하지 않다.

이러한 이유에서 interface는 그 자신의 operation을 위한 ROSE schema에 있어서 읽기 위해 요구된다. 이들 columns는

· schemas_id -- RoseDesign에서의 design_id와 일치한다.
· Schema_name -- schema의 name
· Schema_index -- order를 maintain하기 위한 column

OID_Mapping

oid_mapping은 database내에 저장된 각각의 object에 대한 tuple을 포함한다.

이것의 column은 :

·design_id -- RoseDesign에서의 design_id와 일치한다.
· oid oid는 ROSE에서 정의된 full OID(Object Identifier)를 포함한다. OID는 시간과 공간을 초월하여 ROSE object를 유일하게 식별한다.
· oid_key -- oid_key는 RDB에서 사용되어지는 internal key이다. oid_key는 oid mapping relation의 primary key이다. Unique한 oid_key는 RDBMS에서 저장된 각각의 oid에 대해 생성된다.

Entity_Type

database에서의 object의 type. 이는 object의 immediate type과 일치한다. name은 relation name과는 다르다, 왜냐하면 relation name는 legalized되었기 때문이다.

· Mapping of EXPRESS Primitives

RDBMS의 data type은 STEP에서 정의된 것들과 같이 광범위하지는 않다. 그런고로 <표4>와같이 가장 가까운 타입을 근사시켜 사용한다.

<표4> EXPRESS vs. SQL type

EXPRESS	SQL
INTEGER	Integer
BOOLEAN	Integer
LOGICAL	Integer
REAL (float, double)	float, double
ENUMERATION	varchar
STRING	varchar
BINARY	number

· Mapping of ENTITY Type

STEP entity type은 생성된 단일 relation으로부터 mapped된다. EXPRESS primitives는 objects가 foreign key를 사용하여 mapped되는 동안 가장 가까운 가능한 RDB mapping을 이용하여 mapped된다.

예를 들어, 아래 EXPRESS structure를 고려해보자.

```
ENTITY point;
 x : REAL;
 y : REAL;
END_ENTITY;
ENTITY circle;
 center : point;
 radius : REAL;
END_ENTITY;
```

이에 상응하는 SQL definition's는 아래와 같이 제시된다.

```
CREATE TABLE point (
 point_id integer,
 x float,
 y float
);

and

CREATE TABLE circle (
 circle_id integer,
 center_id integer,
 radius float
);
```

각각의 relation들이 그 도입부에 <table-name>_id key field를 가짐에 주의하라. 이 key는 oid_mapping table에서의 oid_key와 relation에서의 unique한 identify's의 object에 부합된다.

Circle이 center_id attribute를 갖는 것과 같이, EXPRESS entity는 non-primitive attribute를 갖는다. center-id는 center를 규정하는 point relation에서의

tuple과 상응한다.

· Mapping of Aggregate Type

STEP Aggregation type은 두가지 relation을 사용하여 mapped된다.

EXPRESS type을 고려해보자.

```
ENTITY polygon;
 vertices: LIST OF point;
END_ENTITY;
```

이것은 두가지 relation을 사용하여 SQL상에 mapped된다.

```
CREATE TABLE polygon (
 polygon_id integer,
 vertices_id integer
);

CREATE TABLE ListOfPoint (
 ListOfPoint_id integer,
 point_id integer,
 point_index integer
);
```

relation ListOfPoint의 unique key는 ListOfPoint_id와 point_index의 복합체이다.

point_index column은 order를 maintain하기위해 나타낸다.

· Mapping of SELECT Type

STEP union type은 각각의 union과 objects에 대한 identifier column 사용에서의 primitive를 위해 생성된 column에 의해서 mapped된다.

EXPRESS definition을 고려해보자.

```
ENTITY Drawing;
      item : SELECT (Circle, Line,
                Text, Polygon, REAL);
END_ENTITY;
```

부합되는 relational table은

```
CREATE TABLE Drawing (
 drawing_id integer,
 union_entity_id integer,
 union_entity_name varchar(80),
 union_float
);
```

만일 union에서의 value가 primitive라면 적절히 저장되고, 그렇지 않으면 union_entity _name은 값이 저장된 relation을 정의하고, union_entity_id는 oid_key를 정의한다.

· Enumeration

Enumerations는 관계형 데이타베이스에서 열거형 string을 저장함으로써 저장된다.

· Inheritance

각 엔티티는 인접한 attribute들에

해당하는columns에서 만들어 진다. 상속된 attribute들은 supertype속에 저장된다.

예를 들어, EXPRESS definition에서

```
ENTITY 3D_Point SUBTYPE
        OF(Point);
        z: REAL;
END_ENTITY;
```

RDBMS 정의는

```
CREATE TABLE 3d_point (
                3d_point integer,
                z float
);
```

이처럼 만일 하나의 Point가 3D_Point 라면, 그것은 두 개의 relations속에 저장된다.

x와 y좌표는 point속에 저장되고, z좌표는 3d_Point 속에 저장된다.

이외에도 STEP Tools사에서 제공하는 여러 가지 제품들이 있으며, 현재 제공하는 제품들의 내용은 <표5>와 같다 [STEP Tools Inc, 96].

<표5> STEP Tools사에서 제공하는 주요 제품들

주요제품	기능 및 관련 제품명
ST-Developer (Base)	· Data Modeling · Application Development · STEP Conformance Checking · IGES/DXF Utilities
ST-DB Adapter (DB Extentions)	· ST-ORACLE · ST-OpenDB · ST-ObjectStore · ST-Versant
ST CAD Converters (CAD Extentions)	· ST ACIS · ST-IGES · ST-DXF
ST-Visualizer (Visualization)	· Virtual trackball for 3D manipulation · Select and examine indivisual curves or surfaces · View or manipulate model as wireframe or solid · Use of multiple lights for realistic rendering
ST-EXPRESS (CASE Tool)	· EXPRESS information modeling (Editor, Layout, Upgrade, Comparison, Converter)
ST WebPublisher (Web)	· Web support · JAVA applets support · STML(STEP Template Markup Language) · GIF Generation

4. STEP의 활용방안

여기서는 STEP의 활용방안에 대해 이기종 CAD 시스템간의 활용과 PDM 시스템에서의 활용에 대해 살펴봄으로써 그 전체 이미지를 구상해 본다.

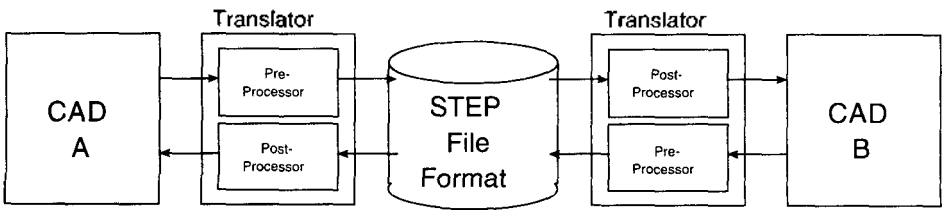
4.1 이기종 CAD 시스템간의 STEP data 활용

CAD시스템간의 데이터 교환은, Part21에서 정의하고 있는 물리적인 파일을 통한 수동적인 교환과 API를 통한 능동적 교환, 공유 DB를 통한 방법, 마지막으로 지능형 지식 기반 시스템을 통한 4가지의 형태가 있으며, 아직은 파일 단위의 교환 정도가 구현사례로써 제시되고 있는 단계이다. 이는 CAD시스템의 고유 포맷을 STEP의 포맷으로 바꿔주는 전 처리기와 반대로 STEP 파일의 포맷을 CAD시스템의 고유 포맷으로 바꿔주는 후 처리기로 구성되는 STEP Translator에 의해 처리되며, 이는 앞서 살펴보았던 ST-Developer를 이용하여 가능하다.

현재 CAD시스템을 제공하는 대부분의 벤더들은 STEP의 필요성을 인식하고 각자 STEP Translator를 제공하고 있으며, AP 214는 아직 C(Committee Stage)단계에 있으나 자동차 업계의 많은 수요로 인해 제정단계 이전에 이미 지원하고 있다 <표6>.

<표6> CAD 벤더들의 STEP 지원

구분	AP 201	AP 202	AP 203	AP 214
CV / CADDS5	*	-	*	*
Dassault Systems - Catia	-	-	*	*
EDS Unibraphics	*	*	*	*
NG - ParaSTEP	-	-	*	-
Hewlett Packard	-	-	*	*
Intergraph	*	*	*	*
MCS Aries	-	-	*	-
Pro/Engineer	-	-	*	*
SDRC	-	-	*	*
ITI	-	-	*	-
STEP Tools - ACIS	-	-	*	-



<그림8> 이기종 CAD 시스템간의 파일 단위 데이터 교환

4.2 PDM 시스템에서의 STEP data활용

4.2.1 PDM 시스템의 개요

PDM은 제품과 관계된 모든 정보를 기획단계에서부터 폐기에 이르기까지의 제품 전 수명주기(Product Life Cycle)범위 동안 일괄 및 통합관리하는 도구이다. 이러한 PDM이 관리하는 정보의 형태는 크게 3가지 정도로 나누어 볼 수 있으며, 이는 다음과 같다[최열현, 95].

- 제품 정의 정보
  (Product Definition Information)
  - 입력정보 : 규격 및 사양, 시장 정보, 고객 정보, 기능 요구 사항
  - 출력정보 : 도면, BOM, 가격, 시방, 기술자료, 부품 정보 등
- 조정 및 운영 정보
  (Control Information)
  - 시방 변경 정보, XX 검토 요청서, XX 결과서, XX 합의서
  - 일정 관리 정보 등
- 기술 정보 (Technical Information)
  - 시험/테스트 결과 정보, 검사 결과 정보 등

이들 정보는 STEP에서 지원하고자 하는 정보들과도 일치하며, 특히 제품모델정보 외의 여러 가지 부가정보까지도 포괄하고 있어 STEP을 PDM에 적극 활용함으로써 그 효과를 극대화 시킬 수 있다 하겠다.

특히, PDM 시스템에서의 STEP은, 4가지 제품 데이터의 요구와 부합되기 위한 최상의 해답으로서 제시되고 있다.

이 4가지 제품 데이터 요구 [CIMdata, 96]란,

- 이식성(Portability)
- 상호운용성(Interoperability)
- 지속성(Longevity)
- 확장성(Extensibility)

를 뜻하며, STEP은 이들 요구를 만족시킨다.

STEP은 표준으로서 이기종의 하드웨어 시스템이나 오퍼레이팅 시스템들간 뿐만아니라 서로다른 어플리케이션들 상호간의 데이터 이식을 가능케한다.

또한 CE를 위한 상호운용성 측면에서보면, 팀원이 서로다른 일을 하거나 동시병행적으로 하기위해 이기종 어플리케이션간에서 동일한 데이터의 공유가 가능해야 하며, 이에 동일한 포맷의 정보를 제공해 줌으로써 지원한다. STEP은 하나의 구조로서 독립적인 자원의 활용을 통해 AP 개발에 있어서도 상호운용성을 지원하며, 이는 곧 AP들간에도 상호데이터교환이 가능하여 용이하다.

데이터는 소프트웨어나 하드웨어 시스템의 생명성에 관계없이 지속적이어야 하며, 중립화된 포맷의 표준을 통해 시스템에 독립적으로 적용될 뿐만아니라 버전 변경에 있어서도 지속성을 지니고 있다.

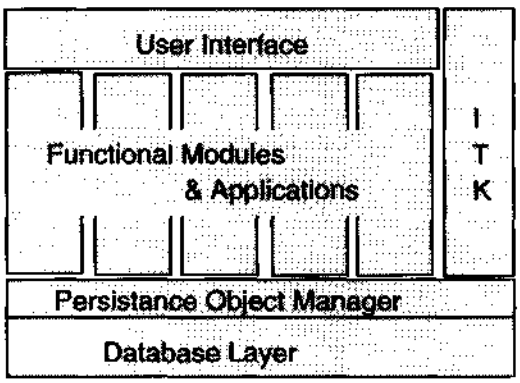
STEP data의 이용은 확장성 측면에서도 강력한 지원을 하며, PDM에서는 이를 통해 데이터의 연계관리 및 외부시스템들(MRP, CAx , etc.) 과의 통합에도 구조적으로 적용되므로 매우 유용하다 할 수 있다. 이는, 향후 CALS환경으로의 확장에

있어서도 핵심적인 요인으로 작용한다.

4.2.2 PDM 시스템에서의 STEP 활용

PDM에서의 STEP 활용 방안을 모색하기 위해 그 내부 구조와 데이터 관리 방식을 살펴볼 필요가 있다.

<그림 9>에서 보는 바와 같이 대부분의 PDM에서는 객체지향개념하의 개방형 시스템 구조를 가지고 있으며, DB는 아직도 안정적인 RDB를 대부분이 취하고 있다. 따라서, 그 사이에 API와 같은 layer를 두어 객체지향화하여 DB를 관리한다. 즉, 각각의 RDB Table과 Class를 매칭시켜주고 oid (object id)를 부여함으로써 ORDB의 형태와 유사한 형태를 취한다.



<그림9> 일반적인 PDM 시스템의 구조[EDS, 96]

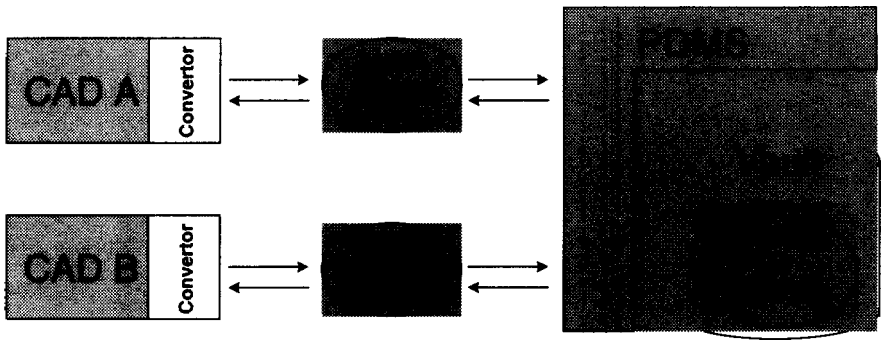
· 수준 1에서의 구현
: 물리적 파일 형태의 교환

PDM에서의 데이터의 관리는 크게 실제 데이터와 그것을 관리하기 위한 메타데이터(Metadata)로 나누어 관리되며, 각각은 PDM Vault(저장고)에 파일 단위로, RDB형태로 서로 다른 저장소에 저장되며 각각은 연계되어 관리된다. 따라서, STEP을 단순히 CAD데이터의 관리 측면에서 본다면 파일형태로 PDM Vault에 저장관리함으로써 가능하다. 이는 상용 PDM Tool인 EDS사의 IMAN의 경우를 예를 들어 설명하면 다음과 같이 간단히 구현할 수 있다 [EDS, 96; EDS, 96; EDS, 96].

먼저, STEP 파일의 생성은 CAD벤더들에 의존하여 Translator를 사용하고, 생성된 STEP 파일(Part21)을 PDM Vault에 저장하기 위해 Dataset type을 .step으로 지정해주고, 이 Dataset type을 사용할 Tool들을 지정하여 연계함으로써 PDM에서 .step형식의 파일을 불러올 수 있다. 물론, Tool을 불러올 때 미리 시행될 간단한 script문을 생성시켜 연계시켜줌으로써 STEP 파일을 불러올 때 백그라운드에서 STEP Translator를 자동시킴으로써 가능하다.

· 수준 2에서의 구현
: API를 통한 능동적 파일 교환

PDM에서의 STEP 활용은 단순 형상 모델 정보 이외에도 여러 가지 부가정보의 통합관리를 위해 보다 능동적이어야 할 필요가 있다. 다시 말해서, 물리적인 파일정보의 교환 이외에 SDAI와 같은 API를 통한 PDM DB로의 활용이 필요하다는 것이다. SDAI를 통한 구현 이미지는 <그림 11>과 같고, 여러 PDM벤더들은 97년 후반부터 이 기능을 선보일 예정에 있다 [CIMdata, 96].

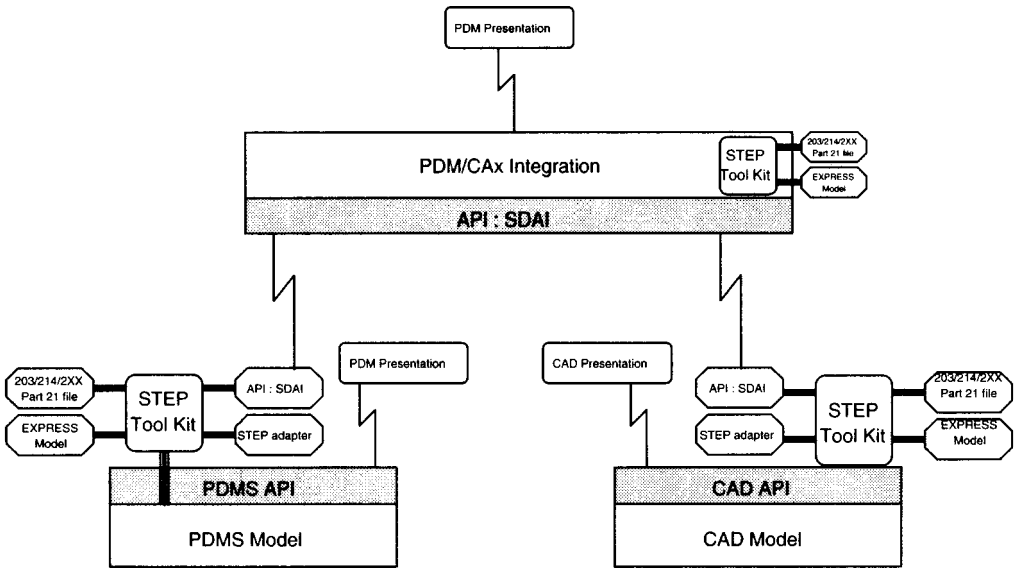


<그림10> 물리적 파일을 이용한 PDM에서의 구현

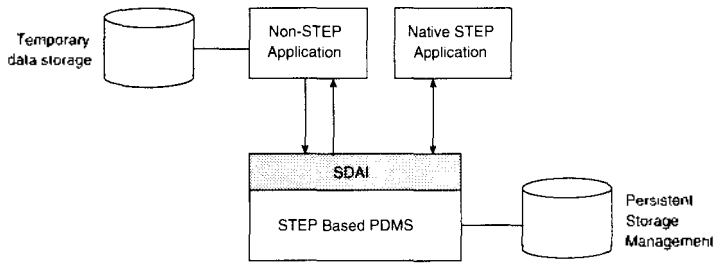
• 수준 3에서의 구현
: 공유 DB에 의한 방법

<그림11>에서의 구조도 아직은 DB에의 완전한 구현은 아니며, 이는 향후 PDMS에서의 자체 API가 SDAI화 되어야 함을 의미한다. 이는 다시 <그림12>과 같이 표현될 수 있으며, CAD를 포함한 STEP을 지원하는 어플리케이션과 지원하지않는 어플리케이션 둘 다를 고려하여 표현하였다.

또한, 이들은 PDM DB자체가 STEP의 구조를 가지고 있어 공유 DB형태의 구현으로 제시될 수 있다[Malcolm D. Spence, 95; Manfred Koethe, 94].



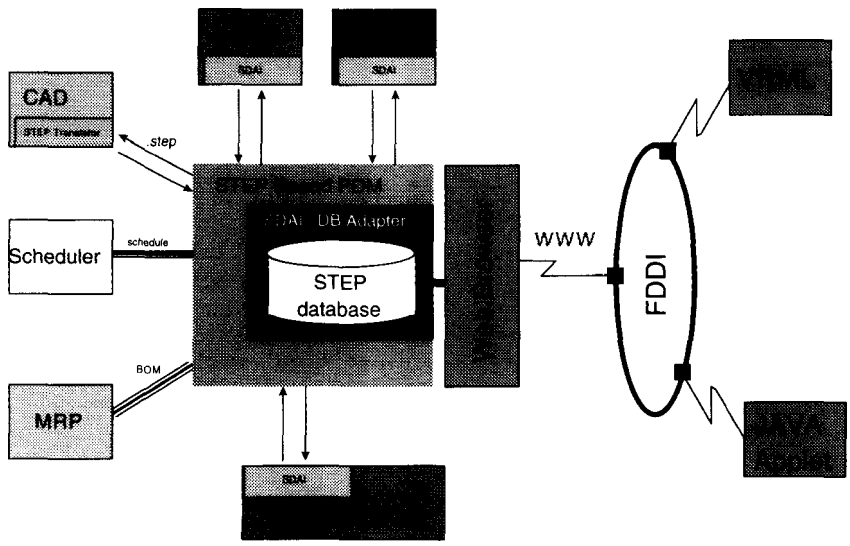
<그림11> SDAI를 이용한 PDMS와 CAD의 통합



<그림12> SDAI를 API로 갖는 STEP Based PDM 구조

4.3 STEP 활용에 대한 전체 구현 이미지

이제까지 STEP의 활용에 대해 그 주된 적용대상인 CAD와 PDM에 대해 살펴보았다. 현재의 상황에서 가능한 부분은 파일형태의 구현이지만, 그 파일의 크기가 기존 파일크기의 1.5배에서 심지어는 3배 정도까지도 커짐으로써, 파일의 크기면이나 속도면에서 많은 문제점을 안고 있다. 또한, DB의 경우에도 대부분이 RDB를 사용하고 그 위에 한 layer를 얹어 사용함으로 인해 SDAI를 통한 접근 속도가 현저하게 떨어짐에 따라 아직 STEP의 미래와 그 효과에 대해서는 명쾌하게 답변할 수는 없다. 하지만, 정보기술과 기계기술의 계속적인 발전과 함께 이러한 문제점들은 곧 해결될 것이며, 이에 SGML과 EDI까지도 포괄하는 더 큰 의미 STEP의 형태가 나타날 것이며, 그것이 곧 CALS의 종점이라고도 볼 수 있겠다. 이러한 조짐은 SWEDCALS나



<그림13> 전체 STEP 활용 이미지

PDES, ESPRIT 등의 많은 Project 연구자료들로 부터 찾을 수 있다[STEP 연구회, 96]. 현 상황에서 고려한 STEP data 의 활용에 대한 전체 이미지를 <그림13>과 같이 제시한다.

5. 결론

이상과 같이 ST- Developer와 ST-Oracle의 활용방법을 통해 보다 실제적이고 구체적인 이해를 위해 STEP의 활용방안을 제시하였다. STEP에 대해 아직도 많은 거부세력이 있는 것이 사실이고, CALS의 구현 및 다가올 정보화 사회에 대해 회의적인 주장이 있는 것 또한 사실이다. 그러나, 현 상황에서의 CALS와 STEP 표준은 미래 기업환경에서의 해답으로서 가장 유력한 대안이며, 그러한 이유로 학계와 업계에서 연구와 구현이 병행되어져야만 한다.

또한, 현재 선진국들 위주로 추진되고 있는 국제표준에 적극 동참함으로써 우리나라의 환경에 맞는 표준의 반영과 국제사회에서의 입지를 지킬 수 있어야 하는 이유에서도 STEP에 대한 관심은 반드시 필요하다. 올해 투표권을 가진 정식 멤버인 P-멤버로 등록은 되었으나, 아직도 STEP에 대한 홍보가 부족하고 또한 STEP에 대한 깊은 이해의 부족으로 업계에서는 아직도 용어조차도 모르고 있는 상황에서 구체적인 이해와 기업에서의 그 활용방안에 대한 연구는 꼭 필요하다 할 수 있으며, 본 논문이 그 역할을 조금이나마 할 수 있기를 바란다.

이후의 연구의 방향에 있어서도, 실제 구현에 더 비중을 두어 각 기업내의 정보를 표현해보고, 또한 그것이 다시 연구에 반영됨으로써 국내 상황에 맞게 추진이 되어야 할 것이며, PDM과 같은 시스템에서도 여러 외부 시스템과의 통합에 있어 적절히 적용됨으로써 그 형태를 확장시켜 나가야 할 것이다.

참고문헌

[김철환, 김규수, 95] 21세기 정보화 산업혁명 CALS·이론과 실제, 도서출판 문원, 1995.

[최열현, 95] IMAN시스템, 월간 CAD/CAM, 1995.

[한국능률협회컨설팅, 95] CALS & PDM Forum자료, 1995.

[황한웅, 96] CALS의 실상, 연학사, 1996.

[CIMdata, 94] "PDM Buyer's Guide," 1994

[CIMdata, 96] PDM Conference Proceedings, 1996.

[EDS, 96] Information Manager ITK Programmer's Guide, 1996.

[EDS, 96] Information Manager SystemAdministrator Guide, 1996.

[EDS, 96] Information Manager Users Guide, 1996.

[Jon Owen, 93] STEP An Introduction, Information Geometers Ltd, 1993.

[Malcolm D. Spence, 95] "An Assessment of STEP as an Architecture for Product Data Integration", AUTOFACT Asia CAD/CAM Symposium, 1995.

[Manfred Koethe, 94] "Product and Process Data Management Using STEP : An Advanced Integration Architecture", Digital Equipment Co., 1994.

[STEP 연구회, 96] 제품 모델 정보 교환을 위한 국제 표준 (ISO10303) STEP, 성안당, 1996.

[STEP 연구회, 96] STEP 기술강좌 자료, STEP연구회, 1996.

[STEP Part 1, 88] Overview and Fundamental Principles, ISO/TC184/SC4, 1988.

[STEP Part 11, 92] The EXPRESS Language Reference Manual, ISO/TC184/SC4, 1992.

[STEP Part 21, 94] Industrial automation systems and integration - Product data representation and exchange, ISO/TC184/SC4, 1994.

[STEP Part 22, 94] Standard Data Access Interface, ISO/TC184/SC4, 1994.

[STEP Part 23, 95] Product Data Representation and Exchange - Implementation Method : C++ Programming Language Binding to the SDAI Specification, ISO/TC184/SC4, 1995.

[STEP Part 26, 96] Product Data Representation and Exchange - Implementation Method : Interface Definition Language Binding to the SDAI Specification, ISO/TC184/SC4, 1995.

[STEP Tools Inc, 96] 제품 홍보 자료, 미니정보 시스템, 1996.

[STEP Tools Inc, 96] EXPRESS/IDL Compiler Notes, 1995.

[STEP Tools Inc, 96] ROSE Library Reference Manual, 1996.
[STEP Tools Inc, 92] ROSE Library Tutorial Manual, 1992.
[STEP Tools Inc, 93] SDAI Library Reference Manual, 1993.
[STEP Tools Inc, 94] Some Common STEP Tasks, 1994.
[STEP Tools Inc, 96] ST-Developer Release Notes, 1996.
[STEP Tools Inc, 96] ST-Oracle Reference Manual, 1996.
[STEP Tools Inc, 94] STEP Utilities Reference Manual, 1994.
[STEP Tools Inc, 95] TCL/SDAI Reference Manual, 1995.
[Steven Ray, 96] "STEP-Related Work at NIST" 강연자료, NIST, 1996.
http://www.admin.kth.se/SGML/Bibliotek/Litteratur/whitep/wp.html, 1996.

저자소개

박정선

1983년 서울대학교 산업공학 학사, 1985년 KAIST 경영과학 석사, 1993년 텍사스대학교(오스틴)에서 경영정보학 박사학위를 수여받았다. 한국 전산원 선임연구원을 역임하였으며, 현재는 명지대학교 산업공학과 조교수로 재직중이다.
관심분야 : CALS/EC, OODB, 전문가 시스템 등

예도경

1995년 명지대학교 산업공학 학사를 취득한 후, 동대학원에서 산업공학 석사를 취득하였으며, 현재 (주)LG-EDS시스템 CALS&CIM사업부 CE/PDM팀에서 근무하고 있다.
관심분야 : CALS/CIM, CE/PDM, OOA/D, STEP, SGML 등

연락처 :

경기도 용인시 남동 산 38-2번지 명지대학교 공과대학 산업공학과
(우) 449-728 TEL : (0335)-30-6453 FAX : (0335)-30-6445
jspark@wh.myongji.ac.kr, stmydka@lgeds.lg.co.kr