

IFC의 객체기반 관계형 데이터베이스로의 매핑

Mapping IFC to Object-oriented Relational Database

김 선 우* 이 강**
Kim, Seon Woo Lee, Ghang

요 약

대부분의 상용 데이터베이스 관리시스템은 관계형 데이터베이스 (relational database) 기술에 기반하고 있다. 그러나, 객체지향언어인 EXPRESS로 정의된 IFC(Industry Foundation Classes)를 일반적으로 많이 사용되는 관계형 데이터베이스로 매핑하려면 서로 구조가 달라 매핑과정이 매우 복잡해진다. 한편 IFC를 객체기반 데이터베이스(object-oriented database)나 객체관계형 데이터베이스(object-oriented relational database)와 같은 객체기반의 데이터베이스로 매핑하게 되면 그 과정이 비교적 단순해지고, 많은 장점을 가질 수 있다. 본 연구에서는 장기적으로 IFC와 객체기술에 기반한 통합정보교환기술을 실용화하기 위하여 먼저 관계형, 객체지향형 데이터베이스의 개념에 대해 살펴보고, IFC를 객체관계형 데이터베이스로의 매핑 방법에 대하여 알아본다.

키워드: IFC (industry foundation classes), EXPRESS, Mapping, Object, Implementation, 관계형 데이터베이스(RDB), 객체기반 데이터베이스(OODB), 객체관계형 데이터 베이스(ORDB)

1. 서 론

1.1 연구의 배경

본 연구는 건물의 수명주기 동안 발생하는 자원 및 정보를 통합적으로 관리할 수 있는 건설정보관리시스템 (project lifecycle management system)의 첫 단계로, 국제 표준 건설정보모델인 IFC(Industry Foundation Classes)와 객체기반 관계형 데이터베이스 (object-oriented relational database)의 매핑방법을 제시하고자 한다.

IFC (Industry Foundation classes)는 AEC/FM (Architecture Engineering Construction/Facility Management) 전반에 걸쳐 프로젝트 참여자 간의 다른 포맷을 가진 데이터 호환을 가능하게 하기 위한 국제표준건설 정보모델이다. IFC는 1994년 결성된 IAI (International Alliance for Interoperability)에 의해 개발되고 있으며 우리나라도 1998년부터 한국챕터(chapter)를 결성하여 정식 회원국으로 활동하고 있다.

IFC는 현재 2x3까지 개발되어 있으며 2003년 IFC2x2모델의 개발이후 BIM(Building Information Modeling)에 대

한 적극적인 연구와 도입과 함께 통합시스템 개발을 위한 기본이 되고 있다. 하지만 이러한 통합정보교환기술이 실용화단계에 이르기 위해서는 여러 가지 해결해야 할 과제들이 있다. 그 중 첫 번째로 풀어야 할 문제는, IFC와 같은 개념모델(conceptual model)을 어떻게 상용화된 데이터베이스 관리시스템의 물리적 정보모델(physical model) (예: 관계형 데이터베이스 schema)로 구현(implementation)하느냐 하는 문제이다.

IFC와 같은 국제표준 프로덕트모델들은 국제표준기구 (ISO: ISO 10303[3])에서 정한 표준정보모델링 절차와 EXPRESS[4]라는 표준 프로덕트 모델링 언어를 사용하여 정의된다. 그런데 EXPRESS는 상속(inheritance)이나 인스턴스화(instantiation) 같은 객체지향언어의 속성을 가지고 있어서, 현재의 많은 상용 데이터베이스 관리시스템에서 사용하지만 객체개념을 지원하지 않는 관계형 데이터베이스 (relational database)구조로의 구현이 쉽지 않다. EXPRESS와 같은 객체지향 언어를 관계형 데이터 베이스에 매핑(mapping)하기 위해서 복잡한 절차가 필요한데 이를 위하여 여러 연구가 진행되었다. 먼저 2001년 일본 SECOM Co., Ltd과 핀란드 VTT가 주축이 되어 개발한 IFC Model Server와 비슷한 시기 미국 조지아텍 (Georgia Institute of Technology)에서 미국철골건설협회 AISC (American Institute of Steel Construction)의 용역을 받아 개발한 철골조 건물 표준정보모델 CIS/2를 관계형 데이터 베이스 시스템으로 개발한 EXPRESS2SQL가 있다. 이 밖

* 일반회원, 연세대학교 건축공학과 석사과정
seonwkim@gmail.com

** 종신회원, 연세대학교 건축공학과 조교수, Ph.D., 교신저자
glee@yonsei.ac.kr

본 연구는 연세대학교 연구비 지원에 의한 연구의 일부임. 과제번호 2006-1-0032

에 EuroSTEP, EPM 등에서 상용으로 EXPRESS언어 기반 전용 데이터베이스 관리시스템을 개발하였다. 1999년 국제 표준 데이터베이스 질의언어인 SQL(structured query language)의 새로운 버전인 "SQL3 (또는 SQL-99)"가 객체개념을 지원하면서 최근 많은 상용 데이터베이스가 객체 관계형 데이터베이스(object-oriented relational database) 형식으로 객체개념을 포함하거나 별개의 객체기반 데이터베이스(object-oriented database)들이 많이 나오고 있다. 그러나 아직 EXPRESS 기반 정보 모델, 또는 IFC를 객체 관계형 데이터베이스로 구현하려는 연구가 진행된 바 없다.

본 논문은 먼저 객체형데이터베이스의 개념과 관련 선행 연구과제를 살펴본 후, IFC를 중심으로 몇 가지 EXPRESS 핵심개념의 ORDB로의 매핑방법을 제시하고자 한다.

1.2 연구의 범위 및 방법

본 연구에서는 SQL-99(Structured Query Language 1999)¹⁾(ISO/IEC 1999)에서 제안된 객체지향 DBMS 표준을 따르며, 각 시스템마다 구현방법에 차이가 있기 때문에, 구현에 있어서 구체적인 부분은 본 연구에서 기반플랫폼으로 사용하는 Cubrid 구문을 따랐다. Cubrid는 국내에서 UniSQL이란 이름으로 처음 개발되어 안정성과 성능면에서 국내외적으로 검증이 된 상용 ORDBMS이다 (<http://www.cubrid.com/>).

2. 선행연구

통합정보교환기술을 실용화 하기위한 과정에서 생기는 문제들을 해결하기 위한 선행연구를 살펴보면 다음과 같다.

2.1 IMSVR과 SABLE

일본의 경우 2001년 일본SECOM Co., Ltd와 핀란드 VTT가 주축이 되어 모델서버의 프로토타입을 개발하고자 IFC Model Server Development Project를 수행하였다. 인터넷을 기반으로 하는 모델서버는 IFC 객체(object)모델을 데이터베이스에 저장하고, BLIS-XML²⁾과 STEP파일 형태로 정보를 import/export한다. 이러한 프로토타입은 IFC호환 소프트웨어 애플리케이션을 위한 모델서버의 기본이 된다. 이를 위해서는 스키마의 정의 방법과 변환된 XML스키마가 무엇인지, 데이터 액세스 레이어 컴포넌트(Data Access Layer Component)의 API 디자인, 그리고 웹기반에서의 SOAP(Simple Object Access/Architecture Protocol)³⁾ 기능 사용법 등이 해결되어야 하고, EXPRESS를 XML스키마로 바꾸는 컨버터(EXC)⁴⁾, IFC Object Model

DALC(Data Access Layer Component)⁵⁾ 및 웹 서버 레이어(WSL)⁶⁾의 소프트웨어 모듈이 이용되었다.

IMSVR프로젝트는 인터넷을 기반으로 IFC 호환 애플리케이션이 서로 정보를 교환하고 import/export하게 하였지만 클라이언트 없는 서버에 대한 질문이 제기 되었고 이것은 SABLE (Simple Access to the Building Lifecycle Exchange)프로젝트로 이어졌다.

SABLE은 유럽연합(EU)에서 IFC모델 서버로 정보를 교환하기 위한 심플한 AEC도메인 언어 개발과 이용을 가능하게 하기 위한 목적을 가지고 시작되었다. 이 인터페이스는 무료로서 AEC 도메인에 세부적인 표준 테크놀로지를 이용했다. 또한 모든 기술이 이 애플리케이션을 이용하여 모델서버에 접속 가능하였다. SABLE은 인터넷을 기반으로 데이터의 구조를 쉽게 추출(abstraction) 하고 뷰(view)를 보여주는 객체(object)모델의 장점을 가지며 인터페이스 상의 관계와 전체구조를 구체화 하였다. 단지 데이터 자체가 아닌 데이터 액세스를 정의하므로써 수정이 용이하고 UML (Universal Modeling Language)을 사용하여 개방성이 향상되었다. SABLE은 모델 서버간의 커뮤니케이션을 하나로 합치고 간단하게 하는 BLIS-XML, STEP 10303-P28 XML을 기반 프로젝트였다.

IMSVR과 SABLE의 경우 모두 STEP 10303-P28 (EXPRESS Schema와 XML 매핑 방식을 정의하는 국제표준)을 기반으로 하고 있는데, 이것은 XML 매핑을 통하여 IFC와 데이터베이스 관리시스템을 연결하기 위해서였다. 이 경우에도 역시 데이터베이스가 필요한데 관계형과 객체지향형의 타입이 문제가 된다.

2.2 EXPRESS2SQL

EXPRESS2SQL 프로젝트는 처음 CIS2SQL이라는 이름으로 2000년부터 미국 조지아텍 (Georgia Institute of Technology)에서 미국철골건설협회 AISC (American Institute of Steel Construction)의 용역을 받아 시작되었다. 처음에는 MS SQL을 기반으로 철골조 건물 표준정보모델 CIS/2를 관계형 데이터베이스 시스템으로 매핑하였으며, 이후 MS SQL을 Oracle로 대체하여 EXPRESS2SQL를 개발하였다. You, Yang, Eastman (2004)의 논문에 매핑방법이 자세하게 소개되어 있다.

2.3 EXPRESS기반 데이터베이스

EXPRESS와 같은 객체지향언어나 객체기반의 모델의 경우 관계형 모델로 전환할 경우 그 크기가 몇 배로 커지고 정보처리시간이 길어지게 된다. 따라서 핀란드의 EPM와 EuroSTEP 등과 같은 회사들에 의해 EXPRESS를 토대로 객체기반 데이터베이스를 개발하려는 노력이 있었으나,

1) 1992년에 지정된 SQL-92를 "SQL2"라고 하고, 1999년에 지정된 SQL-99를 "SQL3"라고도 부름. 이후 XML 기능을 부가한 SQL2003, SQL2006 등이 있음.

2) BLIS-XML은 EXPRESS기반의 정보를 XML포맷으로 인코딩하기 위한 방법이며 BLIS-XML 스키마는 EXPRESS 스키마로부터 자동으로 생성되고 자유롭게 쓰고 읽을 수 있음.

3) SOAP은 XML 기반 메시지의 교환을 위한 프로토콜로서 보통 http/https에서 사용됨.

4) EXC (EXPRESS to XML schema Converter)는 EXPRESS 데

이터 스키마를 XML로 바꾸는 것임.

5) DALC는 EXC틀을 이용하여 만들어지고 IFC 모델 데이터를 import/export하기 위한 데이터베이스를 액세스함.

6) WSL (Web Server Layer)는 IFC모델 데이터의 업로드/다운로드 서비스를 제공하며 기존의 웹페이지 위에 사용자 인터페이스를 이용하거나 SOAP프로토콜을 이용함.

아직 개발기간이 짧아 질의나 시스템 안정성 면에서 문제를 보이고 있다.

3. Object-oriented Database의 특성

3.1 Relational Database(RDB)

관계형 데이터베이스는 상용화에 가장 성공한 데이터베이스 유형으로 모든 데이터들을 테이블과 같은 형태로 나타내고 저장한다. 테이블은 열과 행으로 이루어져 있으며 관계형 데이터베이스에는 저장된 데이터로부터 원하는 정보를 추출할 수 있는 방법이 정의되어 있다. 관계형 데이터베이스는 상속(inheritance)나 집합(aggregation) 등 개념 표현, 멀티미디어 등의 저장 등에 한계가 있다. 최근 CAD, CAM, CIM, GIS나 멀티미디어와 같이 더 복잡한 데이터베이스 애플리케이션이 디자인 되고 구현될 때 객체(object) 개념의 해석에 문제가 있는 등 번역과정이 복잡해지는 단점이 있다.

3.2 Object-oriented Database(OODB)와 Object-oriented Relational Database(OORDB)

객체지향형 데이터베이스(OODB)는 객체(object)와 객체의 속성(attributes 또는 properties)이라는 관점에서 데이터를 다룬다. 데이터 타입이나 질의 언어(query language)의 제한 없이 복잡한 데이터의 문제를 다룰 수 있는 유연성이 있으며 데이터베이스 설계자가 객체(object)의 구조(structure)와 오퍼레이션(operation)을 구체화(specify)할 수 있게 한다. 객체지향형에서의 클래스는 비슷한 특성을 가진 객체가 모여 있으며 이것은 관계형 데이터베이스의 테이블이나, EXPRESS의 엔티티(entity)에 대응하는 개념이다. OODB는 각 관계가 각각의 유일성을 식별하는 RDB의 기본 키 속성(primary key attribute)와 비교되는 모든 객체 식별자(object identifier(OID))를 제공한다. 기존 데이터베이스에서는 복잡한 객체(object)에 대한 정보는 많은 관계(relation)이나 레코드(record)로 나뉘어 있고 이것은 실제 객체(real world object)와 그 실제 데이터베이스 상의 직접적인 대응(correspondence)의 손실을 가져온다. 하지만 객체(object)기반의 데이터베이스에서는 그러한 손실이 없고 C++, SMALLTALK, JAVA와 같은 객체지향 프로그래밍 언어의 사용이 수월하다. 객체관계형 데이터베이스(ORDB)는 RDB의 개념에 OODB와 같은 객체(object)의 개념이 추가되었다. 이것은 RDB와 OODB의 장점을 모두 가지며 데이터를 매핑하고 구현(implementation) 할 때 두 타입의 방법이 모두 가능하다. 또한 질의어를 효율적으로 지원하며, 더 풍부한 형태의 데이터 타입이 처리 가능하다.

4. EXPRESS로 정의된 프로젝트 모델의 구현(implementation)

ISO 10303 STEP이나 IFC와 같은 국제표준 프로젝트 모델은 국제규약(ISO 10303 Part 11)에 따라 모두

EXPRESS로 정의 되어 있다. EXPRESS 스키마를 실제 데이터베이스 시스템으로 구현하기 위한 방법은 프로그래밍 언어를 바인딩(binding)하는 방법과 직접 데이터베이스에 매핑을 하는 두 가지로 나눌 수 있다.

4.1 STEP 데이터와 언어의 바인딩(binding)

ISO 10303 STEP Part 21번에서 Part 28번까지는 스키마와 데이터베이스 시스템을 연결하는데 사용되는 프로그래밍 언어에 따라 바인딩(binding)하는 방식을 정의하고 있다. 그 내용을 간략히 정리하면 표 2와 같다.

표 2. ISO 10303 implementation

번호	내용
Part 21	문자기반 데이터 인스턴스 파일의 포맷 정의
Part 22	표준 데이터 접근 인터페이스 (SDAI) 구조 및 함수 정의
Part 23	C++언어와 SDAI의 바인딩 방법 정의
Part 24	C언어와 SDAI의 바인딩 방법 정의
Part 25	OMG (Object Management Group) XMI와 EXPRESS의 바인딩 방법 정의
Part 27	Java와 SDAI의 바인딩 방법 정의
Part 28	STEP데이터 파일을 XML형식으로 매핑 방법 정의

4.2 EXPRESS모델의 DB로의 매핑

EXPRESS를 데이터베이스에 매핑하는 방법은 데이터베이스가 관계형인지 객체기반인지에 따라 달라질 수 있다. EXPRESS는 객체기반의 언어이기 때문에 관계형 데이터베이스(RDB)의 경우 상속(inheritance)이나 특화(specialization), nested class와 같은 개념을 표현하려면 복잡한 매핑과정이 필요하다.

객체기반 데이터베이스(OODB)나 객체관계형 데이터베이스(ORDB)는 모두 객체(object) 개념을 포함하고 있어 EXPRESS의 매핑이 용이하다. 관계형 데이터베이스로 매핑 했을 때 발생하는 많은 테이블의 수와 복잡한 관계를 줄일 수 있고 객체형의 특성상 상속(inheritance), 특화(specialization)와 같은 개념의 실현이 쉬워진다. 또한 번역(translation)되는 과정이나 질의(query)의 처리 속도가 빠르고 데이터의 같이 객체데이터가 가지는 장점이 있다.

현재까지 상용RDB로의 매핑이나 데이터를 담는 그릇인 "리퍼지토리(Repository)"의 상용 개발 사례는 있지만 본 연구와 같이 OODB나 ORDB에 매핑한 사례는 없다.

5. EXPRESS에서 ORDB로의 매핑

여기서는 EXPRESS를 ORDBMS인 Cubrid의 SQL문을 사용하여 IFC2x3에서 핵심적인 ENTITY, SUPERTYPE/SUBTYPE, AGGREGATION 구문에 대해

객체지향의 관점에서 매핑하기 위한 방법을 살펴보고자 한다.

5.1 ENTITY의 구현(implementation)

다음과 같은 IFC2x3의 EXPRESS의 예가 있다.

```
TYPE IfcPositiveLengthMeasure = REAL;
END_TYPE;

ENTITY IfcDoor
SUBTYPE OF (IfcBuildingElement);
  OverallHeight : OPTIONAL
    IfcPositiveLengthMeasure;
  OverallWidth : OPTIONAL
    IfcPositiveLengthMeasure;
END_ENTITY;
```

ENTITY는 RDB의 TABLE에 해당되는 개념으로써 object관점에서 CLASS에 해당한다. IfcPositiveLengthMeasure는 객체지향 언어의 사용자 정의 데이터 유형(user-define type)이고, SUBTYPE의 절은 여기서는 생각하지 않도록 한다.

이것을 CLASS로 정의하면 한 번 정의한 값을 재사용할 수 있다.

```
CREATE CLASS IfcPositiveLengthMeasure (
  PositiveLengthMeasure Double)

CREATE CLASS IfcDoor (
  OverallHeight IfcPositiveLengthMeasure,
  OverallWidth IfcPositiveLengthMeasure
)

[Instance 입력]
INSERT INTO IfcDoor (
  OverallHeight,
  OverallWidth)
VALUES (
  (INSERT INTO IfcPositiveLengthMeasure
    (PositiveLengthMeasure)
  VALUES (210),
  INSERT INTO IfcPositiveLengthMeasure
    (PositiveLengthMeasure)
  VALUES (90),
  .....);
```

이 경우 인스턴스값 입력시에 nested query를 써야 하는 번거로움이 있다. 하지만 ATTRIBUTE로 정의할 경우 아래와 같이 인스턴스 입력이 단순해질 수 있다.

```
CREATE CLASS IfcDoor (
  OverallHeight Double,
  OverallWidth Double
)
```

5.2 SUPERTYPE/SUBTYPE

같은 예제에서 SUPERTYPE/SUBTYPE를 사용하여 속성들을 합칠 필요 없이 자동적으로 상속 (INHERITANCE)이 가능하다.

```
ENTITY IfcBuildingElement
ABSTRACT SUPERTYPE OF (ONEOF
  (IfcBeam
  ,IfcBuildingElementComponent
  ,IfcBuildingElementProxy
  ,IfcColumn
  ,IfcCovering
  .....))
SUBTYPE OF (IfcElement);
END_ENTITY;
```

이것은 다음과 같이 OBDB에 SUBTYPE을 생성하거나 기존에 CLASS에 SUPERTYPE을 추가하는 방법으로 매핑할 수 있다.

```
[SUBTYPE 생성]
CREATE CLASS IfcElement
CREATE CLASS IfcBuildingElement UNDER IfcElement

[SUPERTYPE 추가]
ALTER CLASS IfcDoor ADD SUPERCLASS
IfcBuildingElement
```

5.3 SELECT 데이터 유형

SELECT 데이터 유형은 SELECT 리스트에 정의된 여러 개의 ENTITY 중 하나를 골라 사용할 수 있는 데이터 유형이다. SELECT 데이터 유형의 경우 대응하는 OODB 또는 ORDB의 개념이 존재하지 않기 때문에 별개의 속성으로 정의해야 한다.

5.4 AGGREGATION

AGGREGATION은 하나의 필드가 하나 이상의 값을 갖는 것을 의미한다. 이것은 RDB에서 정규형(normal form)에 위배되는 형태이지만, 구현이 가능하다면 실제 DB구현에서 매우 유용하게 사용할 수 있다. 예를 들어 건물 각 층마다 각 실의 개수가 2개에서 14개로 매우 유동적이고, 또 각 실마다 또 각기 다른 속성을 가지고 있다는 것을 표현할 수 있다.

OODB와 ORDB에서는 COLLECTION이 AGGREGATION에 해당하는데 EXPRESS에는 중복(duplication)과 순서(order)에 따라 SET, LIST, BAG, ARRAY의 네 가지 유형으로 나뉜다. SET은 순서(order)는 없지만 중복(duplication)이 허용되지 않는다. 반면에 LIST는 순서(order)가 정해져 있지만 중복(duplication)이 허용된다. BAG은 순서(order)가 없고 중복(duplication)이 가능하고 ARRAY는 최대 · 최소 인덱스 값으로 표현된 정해진

개수의 원소를 찾는다. EXPRESS의 SET, LIST, BAG, ARRAY 네 가지 개념은 다음 표 3과 같이 매핑 할 수 있다.

표 3. ISO 10303의 구현(implementation)방법

EXPRESS	정의	OODB / ORDB
SET	순서 없음, 중복 불허용	Set
BAG	순서 없음, 중복허용	MultiSet
LIST	순서 있음. 중복 불허용	List or Sequence
ARRAY	최대, 최소 인덱스 값으로 표현된 정해진 원소를 찾음	-

6. 결론

통합정보교환기술을 실현하기 위해 많은 연구와 노력들이 이루어지고 있고, 객체지향의 EXPRESS로 표현된 IFC 스키마를 데이터베이스에 매핑하는 것은 그 중 핵심적인 사항이다. 복잡한 객체기반의 언어로 이루어진 데이터를 손실과 지연 없이 매핑하기 위해서는 상용화된 관계형의 데이터베이스 보다 객체기반의 데이터베이스가 효과적이다. 본 연구에서는 Cubrid의 SQL문을 사용하여 ENTITY 구현 (implemetation)과 상속(inheritance)관계, SELECT구문, SET구문 등의 AGGREGATION을 ORDBMS로 매핑 해보았다. 이것은 AEC(Architecture Engineering Construction) industry의 통합정보교환을 위한 IFC 관계형 데이터 매핑의 새로운 방법으로 제시 될 수 있으며, 건축물 수명주기의 관리를 위한 통합 건설정보관리시스템 (project lifecycle management system)의 첫 단계로서 의미를 가진다.

추후 연구에서는 SELECT 구문과 AGGREGATION관계의 예문을 보완하고, RULE과 WHERE절의 처리 등 여기서는 언급되지 않은 EXPRESS의 나머지 구문들과 예시를 추가 할 예정이다.

감사의 글

본 연구는 연세대학교 신진교원 연구과제 (과제 번호:

2006-1-0032)의 일부로 이루어진 것으로, 본 연구를 가능하게 한 연세대학교에 감사드립니다.

참고문헌

1. Christensen, D. S., "The Estimate at Completion Problem:A Review of Three Studies", Project Management Journal, PMI, Vol. XXIV, No. 1, 1993, pp. 37-42
2. Diekman, J. E., "Knowledge-based approach to construction project control", International Journal of Project Management, Vol. 10, No. 1, 1992, pp. 23-30
3. Kulkarni, D. V., "CM:CSCSC Integration - The Key to Success of Major Public Sector Programs", Project Management Institute Seminar/Symposium, PMI, 1991, pp. 280-286
4. You, S.-J., D. Yang, and C.M. Eastman. Relational DB Implementation of STEP based product model. in CIB World Building Congress 2004. 2004. Toronto, Ontario, Canada
5. Elmasri, Navathe, "Fundamentals of Database Systems 5th Edition", Addison Wesley, 2007
6. Stonebreaker, "Readings in Database Systems", Morgan Kaufmann, 1994
7. 김건식, "Earned Value Management System의 개요", 건설관리학회지, 한국건설관리학회, 제1권 제2호, 2000, pp. 3-7
8. 김선규, "공정관리 전문가시스템 활용전망", 전력기술지, 한국전력기술, 제2권 제3집4, 1991, pp. 53-59
9. <http://www.cubrid.com/>

Abstract

Mapping of EXPRESS, which is object-favored language to represent IFC model, to Relational Database is not straightforward. Model size can be much bigger and data can be missed through process. However mapping to the object concept added database, such as Object Oriented Database or Object Relational Database, may be simpler and have lots of advantages. This study investigates previous IFC mapping studies, concept of Relational Database and Object Oriented Database, and mapping methodology to Object Relational Database using object.

Keywords : IFC, EXPRESS, Mapping, Object, Implementation, Relarional Database(RDB), Object Oriented Database(OODB), Object Relational Database(ORDB)