

STEP 기반 CAD 데이터베이스의 액세스 성능 평가 실험

김준환⁺ · 한순홍

한국과학기술원 기계공학과

An Evaluation of Access Performance of STEP-based CAD Database

Junhwan Kim, Soonhung Han

Department of Mechanical Engineering, KAIST, Taejeon, 305-701

In shipbuilding area, data sharing is one of the crucial issues. Recently, for collaborative design, ship structural CAD systems adopt the database as its primary storage. Database is useful to deal with the large amount of design information among the heterogeneous design department and design stage. To make the database-based CAD system object-oriented database(OODB) and object-relational database(ORDB) can be used. It is important to select proper database because the CAD system performance mainly depends on access performance of database. In this research, using prototype CAD system from other research, access performance of OODB and ORDB form CAD system was evaluated. STEP application protocol was used as the database schema and experiment was made in query by property and query by region. The results give some idea of how to choose the database for CAD systems.

Keywords: CAD database, performance evaluation, STEP, large database (대용량 DB), ship structure

1. 서론

컴퓨터와 CAD/CAM 기술의 발달로, 생산성 향상을 위해 데이터베이스를 중심으로, 설계 초기 단계부터, 생산에 이르는 제품 전주기 정보를 생성, 관리 하는 시스템들이 각 산업 분야에 도입되고 있다. 조선 분야에서도 데이터베이스 기반의 차세대 CAD 시스템이 개발되고 있으며, 모델링 커널을 사용한 조선 CAD의 개발도 제안되고 있다(한순홍, 2001). 또한, 조선 분야에서는 설계 단계마다 서로 다른 CAD 시스템이 사용되어 왔는데, 점차적으로 통합되어 가는 모습을 보이고 있다. 선박과 같은 제품의 형상 모델과 관련 정보들을 단일 데이터베이스로 형성하고, 관련되는 제작업을 제한 없이 처리할 수 있는 컴퓨터 시스템의 구성은 현실적으로 어렵다(이종갑, 1991).

3차원 선체 설계 정보를 다루기 위해서는, 데이터 양이 매우 커지게 되고, 서로 다른 설계 단계 간, 또는 설계 부서 간의 액세스를 위해, 데이터베이스가 필요하다. 데이터베이스 기반의

CAD 시스템을 만들기 위해서, 객체-지향 데이터베이스(Object-Oriented Database; OODB), 객체-관계형 데이터베이스(Object-Relational Database; ORDB) 등의 상용 데이터베이스 시스템을 사용할 수 있는데, 각각의 경우 입·출력 또는 액세스 성능에 따라 CAD 시스템의 성능이 결정될 수 있으므로, 데이터베이스 선택과 적용의 문제는 중요하다.

본 연구에서는 상업용 데이터베이스를 사용한 CAD 시스템을 개발하여, 대용량의 데이터를 입·출력 하고, 설계에 활용하는 과정에서 액세스 성능 평가를 통해, 데이터베이스 기반 CAD 시스템에서 데이터베이스를 선택하고 적용하는 과정에 도움이 되고자 한다. STEP 응용 프로토콜을 스키마로 하여 데이터베이스를 구축한 후, 이것을 형상 모델링 커널과 연동하여 DB 기반 CAD를 구축하고, 속성 검색, 영역 검색 등, 설계자 작업 공간을 데이터베이스에서 가져 올 때의 성능 평가에 대해 연구하였다.

본 연구의 목적은 CAD 시스템 자체의 자료 구조에 기반한

⁺연락처 : 김준환 박사, 305-701 대전시 유성구 구성동 373-1 한국과학기술원 기계공학과 ME3080, Fax : 042-869-3210,

E-mail : everwind@icad.kaist.ac.kr

2003년 10월 8일 접수, 1회 수정 후 2004년 5월 15일 게재 확정.

(1) 자체 파일 포맷, (2) IGES나 STEP 같은 중립 포맷, (3) 형상 정보를 데이터베이스에 저장했을 때의 DB 저장 구조 등을 고려할 때, 액세스 성능은 어떻게 되는가를 조사하여, 좀더 개선된 저장 구조를 설계하고자 하는 것이다. 이 성능 평가의 결과로부터 상용 DB 기반 CAD 데이터베이스의 개선 방안을 찾아낼 수 있다.

2. 기존 연구

전산학 분야에서 여러 종류의 데이터베이스의 성능을 평가하는 방법, 기반 기술에 관한 연구는 현재까지 계속 이루어지고 있다. Fedorowicz은 인덱스 된 파일(indexed file) 환경에서, 성능 평가를 수행하였으며 (Fedorowicz,1987), Darmont 등은 클러스터링(clustering) 알고리즘의 평가를 위한 성능평가를(Darmont 1995), Schkolnick는 관계형 데이터베이스에서 업데이트(update) 작업의 비용을 측정하였다 (Schkolnick,1985). Graefe은 대용량 데이터베이스를 위한 Query의 평가 방안을 제시하였다 (Graefe,1993). 국내의 최신 연구로는, 김성진 등이 제안한 객체-관계형 데이터베이스 시스템을 위한 새로운 성능 평가 방법론에 관한 연구가 있다(김성진,2002).

CAD와 제품 모델 데이터베이스와 관련된 분야에서는, Express 스키마를 사용한 데이터베이스들의 성능을 평가한 연구가 있다 (Loffredo, 1998). Kempe 등은 CAD 데이터베이스로 적합한 데이터베이스를 결정하기 위해, 몇 개의 상업용 객체 지향 데이터베이스들의 성능을 벤치마킹(benchmarking)하였고 (Kempe,1995), 최근 연구로는 대용량 STEP 데이터베이스 구축에 관한 연구가 있다 (Sun *et al.*,1999).

김준환 등은 STEP 데이터베이스를 Native Storage로 가지는 3 차원 선체CAD에서 형상 모델링 커널과 데이터베이스 간의 인터페이스를 개발하였다(Kim, 2003). 한순홍은 모델링 커널을 이용한 조선 전용 CAD의 개발 방안을 제시하였다(한순홍, 2001). 본 연구는 STEP을 기본 구조로 한 조선 CAD 시스템 응용이라는 측면과, 데이터베이스와 커널을 연동한다는 점에서 기존의 연구들과 차별된다.

3. 시스템 구현

3.1 실험 시스템

본 연구에서 성능 평가를 위해 사용하는 저장 구조는 <Figure 1>에서 보듯이 크게, STEP 파일, Rose DB, ObjectStore, Oracle OODB의 네 가지이다. 선행 연구에서 개발된 데이터베이스 기반의 선체 CAD 시스템 프로토타입을 그대로 사용하고, (Kim, 2003), 저장구조를 위의 4가지 구조로 바꿔 가면서 액세스한다. 데이터베이스 스키마는 STEP AP218 엔티티와 일대일

대응되어 생성된다. <Figure 2>에서와 같이 형상 모델링 커널과 연동시키기 위해서, 커널의 함수를 캡슐화한 Driver 클래스를 구현하였는데, Driver 클래스의 속성(attribute)은 STEP AP218 (ISO, 1999) 엔티티의 attribute와 같다. 따라서, <Figure 2>와 같이 STEP, 데이터베이스와 형상 모델링 커널이 연동된다.

형상과 관련해서는 STEP의 Part 42의 일부를 지원하고 있으며, AP218도 Part 42를 지원하기 때문에, Plane, Bsplinesurface, Surface_of_linear_extrusion 등의 기본적인 형상을 지원한다.

테스트 시스템의 구현 환경은 <Table 1>과 같다.

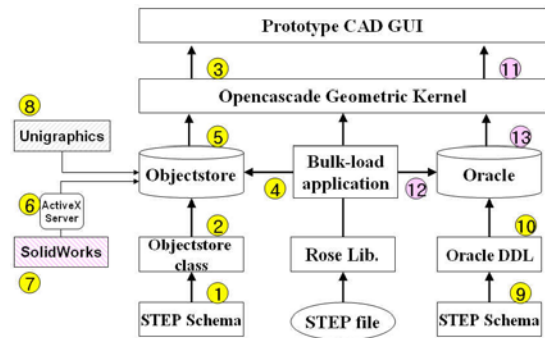


Figure 1. Procedure of system implementation.

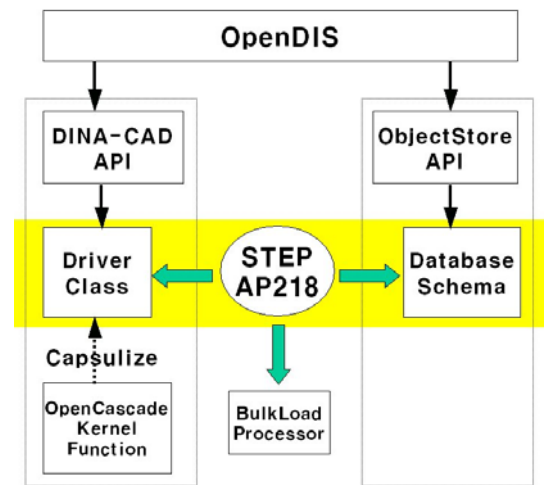


Figure 2. Conceptual structure of prototype CAD system(Kim, 2003).

Table 1. Implementation Components

구분	환경
Operating System	Windows XP
Geometric Kernel	Open Cascade 3.1
Hardware	CPU (Pentium4 2.4G) 1GB memory
Database	Object Store 6.0 Oracle 9i
Programming 환경	C++, Visual Basic, ActiveX,
STEP Library	ST-Developer 8.0 Win.

Loffredo는 오라클, ObjectStore, OpenODB 등의 STEP DB 액세스 성능을 평가하였는데, 비형상 정보 중심이었고, 저장한 데이터는 5M 이내이다. 조선 분야에서는 100메가 이상에서 수 GB 정도의 대용량 데이터의 실험이 필요하다.

성능(performance)은 기본적으로 커널에서 제공하는 저장(save) 기능을 사용하는 것보다는, STEP 데이터 타입 변환에 의해 성능이 느려진다. 하지만, “데이터베이스 기반 CAD”에서는 데이터 타입 컨버전 문제는, 어떤 스키마를 사용하건 발생하는 문제이므로 감수해야 한다. 속도를 빠르게 하기 위해, 가시화(view)만 한다면, 완벽한 데이터가 필요 없다. 이런 측면에서, 본 논문에서 제안된 Prototype CAD는, 가시화 모드, 출력(export) 모드, 작업 모드의 세 가지 모드를 제공한다(김준환, 2002). 본 연구에서는 Rose 라이브러리를 사용하여, 형상 엔티티, 엔티티 간 관계(relationship) 등을 파악하였고 데이터베이스에 저장하였다.

3.2 OODB 실험 환경

메모리가 400메가 바이트인 펜티엄3 개인용 컴퓨터에서, Actify Viewer를 사용하여 가시화하는데, 10MB 데이터가 40분 정도 시간이 걸리며, 비형상 관계(relation)까지 가시화하는 틀은 수시간이 걸린다. 이처럼, 대용량 데이터 처리 프로그램 구현에서는, 한번의 처리 시간이 매우 길기 때문에, STEP 파일을 분석한 후, STEP 파일에서 정보를 추출하여 저장할 수 있는 프로세서의 구현과 처리 범위를 결정하고, 처리할 엔티티를 결정 한 후, Rose 라이브러리를 사용하여 추출하고, ObjectStore 데이터베이스 API를 사용하여 저장하였다.

AP203은 형상 정보를 가질 뿐, 선체의 속성 정보를 가지고 있지는 않다. 속성 정보는 도면을 참조하여 설계자가 입력하여야 한다. AP218도 형상 정보 자체는 shell_based_surface_model 등 AP203에서도 사용하는 엔티티를 사용하므로, 형상 정보 자체의 변환에는 문제가 없지만, 속성과 관계(relation)들을 복원시켜야 하는 문제가 있다.

테스트 프로그램은 다음과 같은 순서로 진행된다.

- (1) OpenCascade 자료 선언 부분에는 Handle(Geom_BSplineCurve)와 같이 STEP 자료구조와 호환성이 있도록(compatible) 선언한다. ObjectStore 스키마는 직접 작성하였는데, db_surface_of_linear_extrusion,와 같이 db_ 접두사를 붙여서 사용하였다. 각각의 스키마는 헤더 파일에 정의되어 있다. 많은 양의 데이터들의 집합을 관리하기 위해서 Object-Store 컬렉션(collection) 객체로 선언하여야 하는데, <Figure 3>과 같이 여섯 개의 Collection을 사용하였다. 예를 들어, db_b_spline_curve가 가지는 control_point의 경우와 같이, 순서가 상관있는 경우에는 os_List를 사용하였고, 그렇지 않은 경우에는 os_Set을 사용하였다.
- (2) ObjectStore를 초기화한다.

```
(a) os_List<db_edge*>
(b) os_Set<db_plane*>
(c) os_List<os_int64*>
(d) os_List<my_double*>
(e) os_List<db_cartesian_point*>
(f) os_Set<db_surface_of_linear_extrusion*>
```

Figure 3. Collection object used for implementation.

- (3) 영속객체(persistent object)로 저장하고자 하는 객체는, 데이터베이스가 객체의 형식을 알 수 있게, os_typespec을 마련해야 한다. 모든 오퍼레이션에서 객체의 type을 알려주어야 한다.
- (4) 데이터베이스를 열고, 트랜잭션을 시작한다.
- (5) 데이터베이스의 커서(cursor)를 사용하여, 탐색(navigating)하거나, 쿼리 명령어로 원하는 객체를 가져온다.
- (6) 데이터베이스의 자료 구조를 OpenCascade 구조로 매핑한다. 예를 들어, STEP의 평면(plane)은 BRepBuilderAPI_MakeFace 클래스를 사용하여 구현할 수 있다. surface_of_linear_extrusion을 생성하기 위해, <Figure 4>와 같은 OpenCascade 클래스가 필요하다.

```
(a) Geom_BSplineCurve
(b) Geom_SurfaceOfLinearExtrusion
(c) Geom_RectangularTrimmedSurface
```

Figure 4. Example query statement of objectstore.

- (7) OpenCascade 자료 구조를 가시화한다.
- (8) 데이터베이스 트랜잭션을 마친다.
- (9) 데이터베이스를 닫는다.

```
os_Set<db_plane*> result =
    ptr->query("db_plane*",
    "this->position->x < 15000", db);
```

Figure 5. Example query Statement of Objectstore.

<Figure 5>는 ObjectStore가 지원하는 쿼리의 문법 형식이다. Db_plane 컬렉션에서, 위치의 x좌표가 15,000보다 작은 plane들을 모두 찾아내는, 쿼리의 예제를 보여준다.

3.3 ORDB 실험 환경

ObjectStore OODB는 사용자가 복잡한 LowLevel 프로그래밍까지 해 주어야 한다. 오라클은 고수준의 쿼리 언어(High Level Query Language)에 의해 쿼리할 수 있다. 현재의 ORDB로 사용되는 오라클에서는 <Figure 6>의 DDL과 같이 복합 객체를 선

언할 수 있다. “UNDER”를 사용하여 advanced_face에서 상속받은 클래스인 db_plane 클래스의 Inheritance를 나타낼 수 있고, db_direction 등의 사용자 정의 타입을 가질 수 있다.

관계형 데이터베이스에서 복합 객체를 사용하지 않고, CAD 데이터베이스를 구축하려면, 엔티티마다 테이블을 구축해야 하는데, AP203을 스키마로 쓴다면 수백 개의 테이블이 필요하고, 엔티티를 찾기 위해, 많은 조인 연산을 필요로 한다. 관계형 데이터베이스에서 지원하는 타입만으로 CAD 데이터베이스를 구성할 경우에는 가능하지만, 본 연구에서는 CAD 시스템 전반의 데이터베이스를 지원하기 위한 것이므로, 다양한 타입과 그들의 집합, 리스트 등 컬렉션 타입을 지원해야 하는데, 관계형 기능만으로는 가능하지 못하여, ORDB 옵션을 사용하여 구현하였다. 따라서, 본 연구에서는 오라클을 ORDB로 분류하였다.

```
CREATE TYPE db_plane UNDER advanced_face
(
  plane_number NUMBER,
  axis_dir db_direction,
  ref_dir db_direction,
  my_position db_cartesian_point,
  face_bound edge_list_typ
);
```

Figure 6. Data definition of complex object.

구현을 위한 과정을 살펴보면, 우선 데이터베이스 스키마를 DDL로 선언하고, DDL 문장을 실행하여 테이블을 생성한다. 그리고, STEP 파일의 정보를 오라클 데이터베이스의 콘텐츠로 변환해 주는 벌크로드(Bulkload) 프로그램을 사용하여, 대용량 데이터를 오라클에 입력한다. 물론 실시간으로 CAD GUI를 사용하여 입력할 수도 있다. 데이터베이스 액세스 과정은 다음과 같다.

- (1) OStartup()를 사용하여, OO4O를 초기화 시킨다.
- (2) ODatabase odb("", "ap203", "ap203"); 오라클 데이터베이스를 선언한다.
- (3) 검색된 결과들을 저장할 수 있는 동적인 컬럼 Odynaset를 선언한다.
- (4) 데이터베이스에서 검색된 후의 결과를 저장할 수 있는, 각각의 객체에 관한 Oobject 타입의 변수를 생성한다.
- (5) GetFieldValue 메서드를 사용하여, 결과를 얻는다.
- (6) 데이터베이스 접속을 마친다.

<Figure 7>은 생성된 테이블에서 가능한 몇 가지 데이터 조각 언어 문장을 나타낸다. (a)는 cartesian_point를 저장하기 위한 객체(object)를 선언하는 것이다. 관계형 데이터베이스에서는 테이블에 기본 타입밖에 저장될 수 없는데, 관계 객체형 데이터베이스에서는 가능하다. (b)와 같이 사용자가 정의한 객체의 배열도 선언할 수 있고, 테이블에 저장도 가능하다. (c)는 다양

한 사용자 정의 타입을 사용해서 객체를 저장한 후 면(face)이 평면(plane)인 경우 평면의 번호를 구하는 쿼리이다. 면(face)이라는 객체가 들어갈 공간에는 평면(plane)은 물론, b_spline_surface나 cylindrical_surface 등 다양한 형태가 저장될 수 있다.

```
(a) CREATE TYPE db_cartesian_point AS OBJECT
(entity_id NUMBER, coord_1 VARCHAR2(20),
coord_2 VARCHAR2(20), coord_3 VARCHAR2(20));
(b) CREATE TYPE edge_list AS VARRAY(4) OF db_
edge;
(c) SELECT TREAT ( face AS db_plane ).plane_number
from ap203;
```

Figure 7. Data manipulation example.

4. 성능 평가 실험

4.1 실험 개요

본 연구의 실험 중 데이터베이스를 액세스하는 CAD 환경을 위해서 이미 개발된 프로토타입 CAD 시스템인 Dina-CAD를 사용하였다(Kim, 2003). <Figure 8-1>은 Dina-CAD 상에서 모델링한 선체의 중앙부이다. <Figure 8-2>는 Midship Section의 전체 모델 중에서 속성을 사용한 검색을 통하여, 대상 모델의 맨 윗 쪽 부분에 해당하는 Deck Plate를 가져왔다. <Figure 8-3>은 영역(zone) 검색의 예제를 보여주며, <Figure 8-4>는 검색 조건을 입력하기 위한 대화 상자이다. 본 연구에서 다루는 액세스 성능 평가 실험은 다음과 같다. 상용 기계-CAD에서의 STEP DB 액세스는 다루지 않는다.

- (1) 상용 CAD 시스템에서 STEP 파일의 액세스
- (2) 상용 CAD 시스템에서 자체 CAD 파일의 액세스
- (3) Dina-CAD 시스템에서 STEP 파일의 액세스
- (4) Dina-CAD 시스템에서 자체 저장소(STEP DB)의 액세스

Table 2. Experimental Cases

From	STEP 파일	CAD 파일	STEP DB
Dina-CAD	(2)	(2) (3) (4)	
상용MCAD	(1)	(1)	-

<Table 2>에 있는 번호에 따라, 4.3절에서 이에 따른 사항들을 토론했었다. 대상 상용 CAD 시스템으로는 Pro/E와 Solid-works를 사용하였다. 다른 CAD에서는 실험용 대용량 STEP 파일을 제대로 재생시키지 못하였다. DB 기반 CAD 시스템으로는 Dina-CAD를 사용하였고, DB 기반 CAD에서의 대용량 액세스와 전체 모델에서 필요한 부분만 가져오는 작업인 선택적 액세스(selective access)를 실험하였다.

예제로 사용된 STEP 데이터는 다음 <Table 3>과 같다. 이 파일들은 조선 CAD에서 STEP 파일로 export 받거나, IGES, SAT 등으로 export 받아 다시 STEP으로 변환시켰다. 이를 Bulkload 시스템을 사용하여, 데이터베이스에 저장한 후, 검색을 시도하였다. 다음의 대용량 테스트 모델들은 STEP AP203 데이터이다. 대용량 데이터 실험을 위해, 실제 데이터를 확보할 수 있는 AP203 데이터를 사용하였다. <Table 3>에서 Dbc-Solid 모델은 10.1메가의 용량을 갖고, 4132개의 surface를 가지며, 그 중에 plane은 3178개이고, 나머지는 extrusion_surface와 b_spline_surface 계열이다.

Table 3. Sample data

모델명	용량(Mb)	manifold	Surface	Plane
Dbc-Solid	10.1	828	4132	3178
Doc-Solid	14.9	890	6402	5352
Dbc-Surface	17.0	0	4132	3178
Doc-Surface	24.6	0	6415	5344

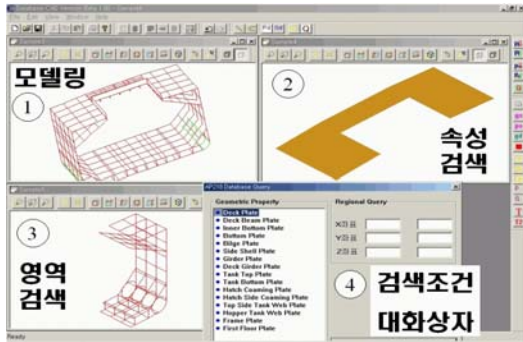


Figure 8. GUI of prototype CAD.

4.2 실험 결과

본 절에서는 구체적인 실험의 예제를 보여주고, 4.1절에서 언급하였던 실험들의 결과를 정리한다. 데이터베이스에는 선박 모델이 들어 있고, CAD에서 이 모델에 대해 설계 작업을 하기 위해 전체 모델의 일부를 Open한다. STEP 파일과 ObjectStore 두 가지 경우에 대한 모델을 Open하는 데 걸리는 시간을 측정 한 실험 결과를 <Table 4>에 나타내었다. <Figure 9-1>은 Doc_Solid 모델을, <Figure 9-2>는 Doc-Solid 모델의 단면 View, Figure 9-3은 Doc_Surface 모델, <Figure 9-4>는 Doc_surface 모델의 단면 View를 각각 보여준다.

Table 4. Opening time for query (Unit: sec)

모델명	STEP파일 (Plane)	ODB (Plane)	STEP파일 Ex_surface	ODB Ex_surface
Dbc-Solid	43	25	88	38
Doc-Solid	84	30	70	39
Dbc-Surf	70	30	75	40
Doc-Surf	135	87	94	43

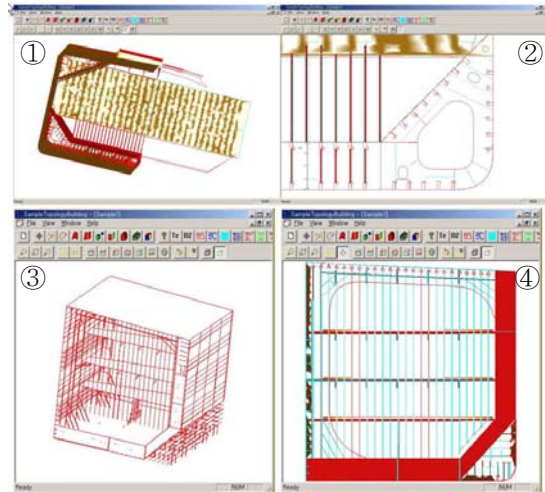


Figure 9. Large amount data access from database.

<Table 5>는 네 가지의 검색(query)을 실험한 것으로, (1) Part의 번호, (2) Part의 위치, (3) Plane의 방향, (4) Surface의 방향을 사용하여 모델의 일부를 가져오는 실험을 하였는데, CAD에서 Open된 결과는 <Figure 10>에 차례대로 나타나 있다.

Table 5. Response time for query by position (Unit: sec)

Geometric Property	전체 Part 개수	Query	결과 Part 개수	Time
(Query by part Number)	5344	P#<50000	700	8
(Query by position)	5344	X<25000	667	8
(Query by direction among plane)	5344	Dir_x==0	144	5
(Query by direction among surface)	880	Dir_y==0	528	12

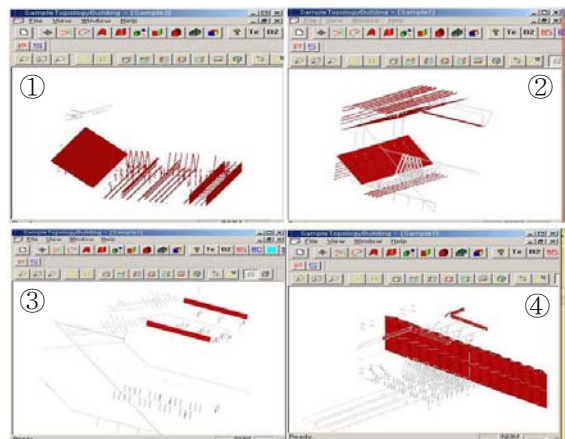


Figure 10. Query by geometric property.

실험 결과들을 정리하면, 다음과 같고, 항목을 나누어 아래에 개조식으로 설명하였다.

- CAD 시스템의 저장 구조로 파일 대신 OODB를 사용한 경우, 성능 저하가 없었다.
- 전체 데이터 중 일부를 가져오는 실험을 했을 때, OODB가 파일보다 좋은 성능을 보였다.
- ORDB를 사용한 경우, OODB보다 성능은 떨어지지만, 10~100메가 정도의 데이터의 입·출력 실험에서, 2~3배 차이밖에 보이지 않았다.

(1) <Figure 11>과 같이 자체 자료 구조를 읽을 때보다 변환해서 읽었을 때의 반응 시간을 실험으로 얻었는데, 약 10~15배 정도로 측정되었다. 중립 포맷을 변환한 후, CAD 시스템 자체 자료 구조로 저장하고 나면 빨라진다. STEP 파일의 정보가 일단 자체 자료 구조로 변환되고 나면, 로딩 시간이 약 10% 정도로 줄어든다.

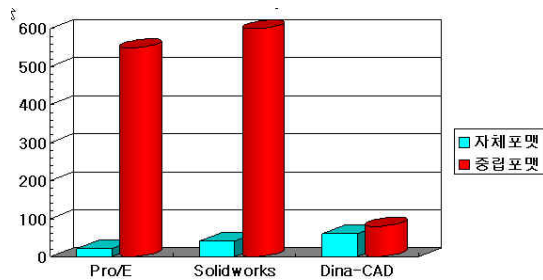


Figure 11. Comparison of CAD file access times.

(2) 파일 대신에 데이터베이스를 사용해서, CAD 시스템을 제작할 때, STEP과 같은 자료 구조에서는 데이터베이스이기 때문에 느려지는 것은 거의 없다(성능 약 90% 정도 발휘한다). 이는 Parametric 관계(relation)가 포함되어 있지 않기 때문일 수 있다. <Figure 12>에서 보듯이(STEP 스키마에 따라 저장된 정보를 OODB로 저장했을 때(파일로 저장했을 때), ORDB에 저장했을 때의 순서로 반응 시간이 빨랐다. 파일보다 OODB가 빠른 이유는 STEP 파일에는 비형상 정보도 담겨 있는데, 읽는 과정에서 이 정보들의 파싱까지 필요했기 때문으로, 구현에 따라 변할 수 있는 요소이다. ObjectStore OODB에서는 형상 정보를 다루는 자세한 수준의(Fine Granularity Level, 형상 모델링 커널 레벨) 프로그래밍을 한 것을 감안하고, ORDB의 성능이 점점 향상되고 있는 것을 고려할 때, 상황에 따라 앞으로 ORDB를 선택할 수도 있겠다.

(3) 전체 데이터 중 선택적으로 일부를 가져오는 실험을 했을 때, OODB가 파일보다 좋은 성능을 보였다. 선택적 액세스(selective access)를 하는 데는 데이터베이스에서는 별도의 프로그래밍이 필요하지 않다. 물론, 파일 기반에서도 똑같은 기능을 구현할 수는 있으나, 추가적인 작업이 필요하다. 전체 데이터에서 일부의 데이터를 가져 올 필요가 빈번한 경우, 데이터

베이스를 사용하는 것이 효율적이다. 일반적으로, 파트 1개를 1개의 파일로 관리하는 경우 선택적 액세스가 필요 없다. 하지만, 하나의 섹터를 한 개의 정보로 관리하고자 하는 경우는 각각의 설계자 관점에서 선택적 액세스가 필요하다.

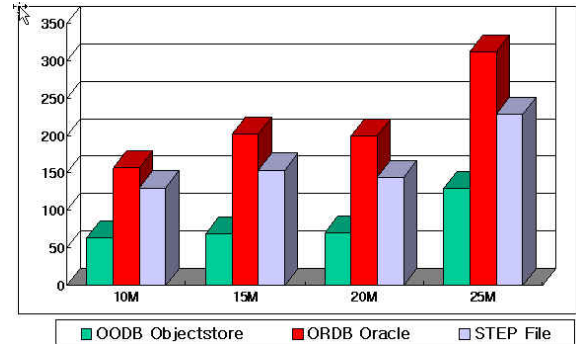


Figure 12. Comparison among database.

(4) 프로토타입 시스템인 Dina-CAD에서 STEP 정보를 빠른 시간에 읽을 수 있었던 이유는 OpenCascade와 STEP의 자료 구조가 유사하여, 내부적인 데이터 변환에 시간이 걸리지 않았기 때문이다. 다른 CAD 시스템 데이터를 액세스하는 경우에는 액세스 자체보다, 변환에 많은 노력이 들 수 있고, 데이터 변환 성능에 관해서는 좀더 연구되어야 한다.

향후 연구에서는 속도를 빠르게 하기 위한, 다음과 같은 요소들에 의해, 얼마나 영향을 받는지 측정해야 한다.

- 액세스 모드를 구분한다.
- 인덱스를 만든다.
- 캐쉬 파일(cash file)을 사용한다.
- 자주 검색되는 부분을 클러스터링(clustering)시킨다.

또한, 향후에는 ObjectStore가 지원하는 다양한 종류의 검색을 실험할 계획이다. 그 예로는, 존재 질의(Existential Queries), 중첩 질의(Nested Queries) 같은 것들이 있다. 또한, 현재는 데이터베이스 검색(Retrieve Operation)만 고려하였는데, 향후에는 수정(Update Operation)의 성능도 고려해야 한다.

5. 결론

본 논문에서는 섹터의 협력 설계 환경을 위한, 데이터베이스 기반 CAD 시스템의 구조를 위한 요소 기술 중, 데이터베이스와 형상 모델링 커널 간의 인터페이스를 채택한 CAD를 중심으로, 대용량 데이터의 액세스 성능에 관해 다루었다. 대용량 데이터 처리를 위해 STEP OODB 기반 CAD 시스템에서, 대용량 섹터 설계 데이터를 입·출력하고 관리할 때의 성능 평가에 관한 연구를 수행하였다. 성능 측면에서는 OODB가 우수하고, 이미 여러 응용 프로그램과 연동되어 있는 등 편리성 측면에서는 ORDB가 우수하므로, 이를 고려하여 데이터베이스를 선

택할 수 있다. 앞으로, 조선 설계 프로세스와 STEP AP218을 좀 더 밀접하게 통합시켜야 하며, 본 논문에서는 데이터베이스 스키마를 STEP Part 42의 일부로 작성하여 조선 분야 선체 설계에 적용하였으나, STEP의 다른 응용 프로토콜을 사용하는 분야에도 같은 방법이 이용될 수 있다.

참고문헌

- 한순홍 (2001), 모델링 커널을 이용한 조선 전용CAD의 개발방안, *조선학회지*, 38(1), 94~99.
- 이종갑, 김준호 (1991), 범용 CAD시스템을 기초로 한 조선 의장설계 일관 시스템, *대한조선학회지*, 29(3), 28-35.
- Fedorowicz J. (1987), Database performance evaluation in an indexed file environment, *ACM transaction on Database*, 12,(1), 85-110.
- Darmont J., Ammar A., Gourgand M. (1995), Performance evaluation for clustering Algorithms in Object-Oriented Database Systems, 6th International Conference, DEXA 95, London.
- Schkolnick M., Tiberio P. (1985), Estimating the cost of updates in a relational database, *ACM transaction on Database*, 10(2), 163-179.
- Graefe G., (1993), Query evaluation techniques for large databases, *ACM Computing Surveys*, 25(2).
- 김성진, 이상호 (2002), 객체 - 관계형 데이터베이스 시스템을 위한 새로운 성능 평가 방법론, *한국 정보처리학회 논문지*, 7(7).
- Kemp J., Kowarschick W., Kiebling W, Hitzelberger R. (1995)

- Benchmarking OODB System for CAD, 6th International Conference, DEXA 95, London September 4-8, 167-176.
- Kim J.H., Han S. (2003), Encapsulation of geometric functions for a ship structural CAD which has a STEP database as the native storage, *Computer-Aided Design*, 35, Issue 13, 1161-1170.
- 김준환, 한순홍 (2002), STEP DB를 Native Storage로 가지는 3차원 선체CAD에서 형상모델링커널과 데이터베이스간의 인터페이스, *한국CAD/CAM 학회 논문집*, 7(4).
- 김준환, 한순홍 (2002), 상업용 CAD에서 STEP 기반 객체지향 데이터베이스 내부의 형상 인스턴스 검색 및 수정, *대한산업공학학회지*, 28(4).
- ISO TC184/SC4/WG3 N799, ISO/CD 10303-218 Product data representation and exchange - Application protocol - Part 218: Ship structures, 1999.
- 손정모 외 (1999), 분산환경에서 표준모델(STEP)을 이용한 내용검색, *한국CAD/CAM 학회 논문집*, 4(6), 285-294.
- Krebs T., Luehrsen H. (1995), STEP databases as integration platform for concurrent engineering, *Proc. 2nd International Conf. on Concurrent Engineering* (McLean, Virginia, August 23-25), Johnstown, PA: Concurrent Technologies Co., 131-142.
- Loffredo D., Efficient Database Implementation of EXPRESS Information Models, RPI, Ph.D. Thesis 1998.
- Sun J., Hardwick M., Building an integrated large scale STEP database for virtual enterprises, *Proceedings of ASME DETC99/DFM*, Sept. 1999, Las Vegas, USA
- Jacobsen K. (1997), Development of a ship modeling application using CAS. CADE, an object oriented software development environment, ICCAS. GRAD Inc., <http://www.gradinc.com/>.



김준환

KAIST 정밀공학과 학사
KAIST 기계공학과 석사
KAIST 기계공학과 박사
현재: 미국 NIST 방문연구원
관심분야: CAD database, STEP, Internet CAD



한순홍

서울대학교 조선공학과 학사
서울대학교 조선공학과 석사
영국 뉴캐슬대 석사
미국 미시간대 박사
현재: KAIST 기계공학과 교수
관심분야: STEP, VR, Knowledge-based design