

STEP 데이터베이스를 Native Storage로 가지는 3차원 선체 CAD에서 형상 모델링 커널과 데이터베이스간의 인터페이스

김준환*, 한순홍**

Interface between Geometric Kernel and Database for a Ship CAD which has a STEP Database as the Native Storage

Kim, J. H.* and Han, S. H.**

ABSTRACT

It is difficult to support collaborative design with a conventional ship CAD system which manages design information using files. In this research, file storage has been replaced with a database. This paper describes the OpenDIS which is an interface between the geometric kernel and the database. Its main purpose is to implement the CAD system which has the STEP database as the native storage. A prototype CAD system has been implemented using that OpenDIS interface which is implemented by OpenCascade geometric kernel and the Objectstore object-oriented database. The STEP methodology is used as the database schema. This CAD system has been applied to the hull design of a ship in order to verify the usefulness of the interface.

Key words : CAD database, Data Interface, Geometric Kernel, STEP, AP218, CAD Data sharing

1. 서 론

제조업에서 형상 정보는 제품 개발 과정에서 생성되는 정보들 중 핵심적인 역할을 담당하지만, 제품 수명 전주기에 걸쳐서 통합 관리되지 못하고 있다. 이것은 각 설계 단계마다 다양하게 나타나는 모델링 요구 조건이 다르고 저장 구조가 다르기 때문이다. 현재의 CAD 시스템은 파일 저장 방식으로 형상 정보를 관리한다. 최근 들어 제품수명주기 단축을 위해 동시공학 개념이 도입되면서, 서로 다른 CAD 시스템을 사용하는 설계 부서간, 설계 단계간의 데이터의 공유가 필요하게 되었다. 현재는 IGES, STEP 등의 중립 포맷을 사용하여 데이터를 교환하거나, 직접 번역을 통해 교환을 하고 있다. 형상 정보를 공유하기 위하여 상용 CAD 시스템 벤더들은 PDM과 CAD를 연동하거나 새로운 추가 모듈을 개발하여 협력설계를 지원하려 한

다. 그러나, 운영체제의 파일 저장 방식에 기초하고 있기 때문에, 파일 간의 외부 참조 관계와 어셈블리와 파트 관계를 파일과 파일 간의 관계를 통해 다루어야 하고, 서로 다른 파일 안에 엔티티들이 분산되어 있을 때에는 원하는 작업 공간을 선택해서 작업하고 다시 저장하기가 어렵다는 단점이 있다. 동시 액세스, 논리적 수준에서의 데이터 관리, 외부 참조, 변경 관리(Change Propagation)을 위해 데이터베이스를 이용한 저장이 필요하다. 또한, 데이터베이스를 구성할 때는 STEP과 같은 표준의 적용이 필요하고, 메모리 구조와의 Impedance mismatch를 극복하기 위한 기술이 필요하다.

최근의 일반적인 CAD 시스템은, 알고리즘들의 집합이며 CAD 시스템의 엔진에 해당하는 형상 모델링 커널을 사용하여 개발된다. 본 연구에서는 형상 모델링 커널과 형상 모델을 저장하기 위한 데이터베이스를 기반으로 한 골격을 채택하여, 형상 모델링 커널과 데이터베이스의 인터페이스를 통해 새로운 CAD 시스템의 구조를 제안한다. 이를 위해서 형상 모델링 커널 데이터(Memory)와 데이터베이스(Persistent 데이터) 간에

*학생회원, 한국과학기술원 기계공학과
**중신회원, 한국과학기술원 기계공학과
- 논문투고일: 2002. 4. 17
- 심사완료일: 2002. 5. 31

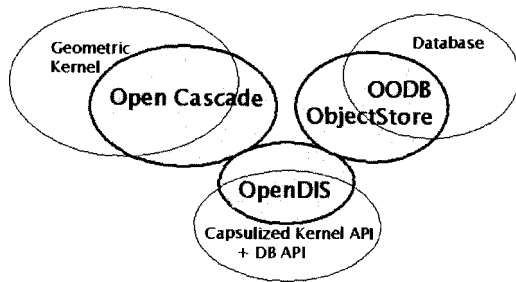


Fig. 1. Conceptual Structure of OpenDIS.

인터페이스가 필요하다. Fig. 1는 본 연구에서 개발된 OpenDIS(Opencascade Database Interface for STEP storage)의 개념적 구조를 보여준다. OpenDIS는 CAD 시스템의 API(Application Programming Interface) 역할과 더불어, 데이터베이스 데이터 구조와 형상 모델링 커널 데이터 구조를 실시간으로 연결하여, 데이터베이스기반의 CAD 시스템의 개발을 가능하게 한다.

본 논문의 구성은 다음과 같다. 2절에서는 데이터베이스를 기본 저장 구조로 갖는 CAD 시스템과 관련 연구들에 대해 설명한 후, 3절에서는 이를 구현하기 위한 요소 기술 중 형상 모델링 커널 데이터와 데이터베이스간의 연동과, 프로토타입 구현에 대해 설명한다.

2. DB를 Native Storage로 가지는 CAD

2.1 DB기반 CAD의 개요

일반적인 상용 CAD 시스템에서는 모델 파일을 여러 사람이 동시에 액세스 할 수 없으나, 데이터베이스기반 CAD는 기본 저장소를 데이터베이스로 하여 동시 액세스가 가능하다. 대형 선박을 건조하는 조선소 현장에서 실제 파트(부품)의 개수는 10만개가 넘는데, 그 파일 이름들이 각각 운영체제에 노출되어 있어 관리가 힘들지만, 데이터베이스기반 CAD는 데이터베이스 내부에서 부여하는 이름으로 관리하므로, 논리적 수준에서 이름 관리가 가능하다. 파트의 structuring 자체가 물리적 수준이 아닌 논리적 수준에서 이루어지므로, 다른 파트간의 참조가 논리적 수준에서 가능하다. 일반적인 상용 CAD에서는 파트 파일 중 원하는 부분을 가져 오기 위해서는, 모델 전체를 메모리로 가져와야 하지만, 데이터베이스를 사용하면 전체 데이터로부터 작업자가 원하는 모델 영역만을 가져 올 수 있다. 하나 이상의 파트에 영향을 주는 작업을 수행했을 때, 전체 데이터의 관리 기능의 도움으로 전체 모델을 열지 않고도, 데이터베이스 수준에서 Relation을 부여해서, 변

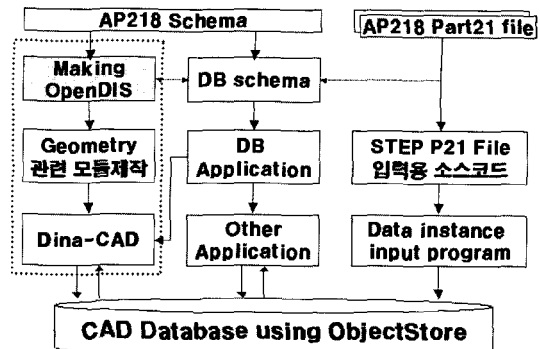


Fig. 2. Process of system implementation.

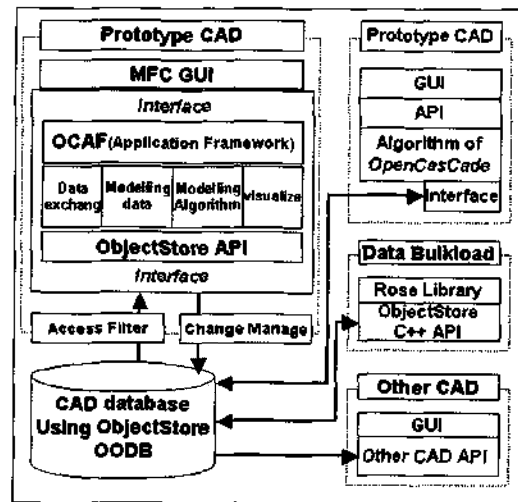


Fig. 3. System Structure of DINA-CAD.

경 관리를 할 수 있다.

시스템의 구축 과정은 Fig. 2와 같이, 선박 구조를 위한 응용 프로토콜인 AP218 스키마로부터 데이터베이스 스키마를 생성하고, 이를 참조하여 OpenDIS를 만들고, OpenDIS를 바탕으로 본 연구에서 개발하는 데이터베이스를 기본 저장 구조로 갖는 프로토타입 CAD 시스템인 DINA-CAD를 제작한다. DINA-CAD는 Fig. 3과 같이 데이터베이스 서버와 형상 모델링 커널 알고리즘을 둘러싼 인터페이스로 구성된다. 데이터를 Bulkload할 수 있는 모듈을 갖고 있으며, 상용 CAD 시스템에서도 액세스 할 수 있다.

DINA-CAD는 형상 모델의 생성, 수정, 삭제 등 기본 기능과 더불어 다음의 특징을 갖는다.

(1) CAD 시스템의 Native Storage를 상용 데이터베이스로 대체한다. 데이터베이스는 보조적인 수단

아니라 CAD 시스템에서 생성되는 모든 정보를 저장할 수 있는 기본적인 저장소이다. (2) 기하학적 형상 뿐만 아니라 속성 정보를 통한 검색이 가능하다. (3) 표준에 근거한 스키마를 바탕으로 데이터베이스가 구축되었다. (4) 공동 설계를 지원할 수 있다. (5) 다른 CAD 시스템뿐만 아니라 다른 응용 프로그램에서도 모델 데이터에 액세스 할 수 있다. (6) 3차원 모델을 사용하여, 전체 설계와 상세 설계를 동시에 지원할 수 있다. (7) 엔터티 간의 릴레이션과 변경 관리를 위해 데이터베이스를 사용한다.

2.2 기존 연구

형상 데이터를 포함한 CAD 데이터의 관리를 위해 데이터베이스를 응용하는 연구는 1980년대부터 진행되었다. 1980년대 후반, Meier^[1]와 Harder^[2]는 솔리드 모델링에 관계형 데이터베이스 기술을 적용시키는 연구를 수행하였다. 관계형 데이터베이스에 솔리드 모델을 저장할 때의 한계점을 서술하고, 데이터를 효율적으로 검색하기 위한 방안을 다루었다. 김원^[3]은 객체지향 데이터베이스가 CAD환경에 적용되기에 적합함을 보였다.

객체 지향 데이터베이스와 형상 모델링 시스템을 연동시킨 GNOME^[4], OsconCAD^[6]와 같은 연구들이 있으며, 사용된 형상 모델링 커널과 데이터베이스 연동 방법은 Table 1에 정리 하였다. GNOME은 Geometric 엔진을 객체 지향 설계 방법론에 의해 구현하고, 객체 지향 데이터베이스와 연동시켰으며, 구현 시스템의 쿼

리, 버전관리 등 데이터베이스 기능에 관한 사항들을 다루었다. OsconCAD는 AutoCAD API를 사용하여 구축한 시스템으로, 파일 포맷을 사용하지 않고, 객체 지향 데이터베이스를 CAD 데이터베이스로 이용하였다. 현재 조선 CAD중 하나인 Autokon의 모체가 되는 Tornado^[7]라는 시스템은 자체 개발 network방식 데이터베이스이고, 물리적 레벨 데이터베이스까지 설계 하였으며, 제공되는 API는 매우 Low Level이고, Concurrent 기능은 없다. 현재 개발중인 조선 전용 GSCAD^[8]는 관계형 데이터베이스인 MSSQL을 저장소로 하고, ACTIS 커널과 연동하여 개발되고 있으며, Short Transaction 개념을 사용하고 있다.

STEP 기반으로 데이터베이스를 구축한 연구들과^[9-11,20], 조선 분야에 STEP을 활용하는 연구들^[12-14]이 국내외에서 이루어져왔는데, 본 연구에서는 STEP을 데이터베이스 스키마로 활용하되, 자세한 형상정보로부터 특징 형상, 도메인 관련 엔터티를 포함해서 속성 데이터까지 다룬다.

현재의 STEP 데이터베이스 구현 수준은 단순한 파일의 저장과 다를 것이 없으며 파일의 업로드-다운로드용 STEP 데이터베이스 스키마이고, 국부적인 수정도 어렵다^[9]. 이는 실제 STEP 데이터는 CAD 시스템 번역기에 의해 얻어지므로, 파일 정보가 중복 저장되는데 원인이 있다.

본 연구에서는 STEP 데이터베이스를 부분적으로 생성, 변경할 수 있도록 하는 인터페이스를 만들어서, CAD의 데이터베이스로 사용하였다. 즉, 상용 CAD 시

Table 1. Implementation method for combination among geometric kernel, database and STEP

1	형상 모델링 커널과 데이터베이스의 인터페이스 (본 연구에서는 OpenCascade와 ObjectStore)
1-1	형상 모델링 커널의 Class를 이용하여 (멤버로 가지거나, 상속 받거나 하여) 데이터베이스 Class를 선언하는 방법 (GNOME) [4]
1-2	응용 프로그램에 맞는 데이터베이스 스키마를 선언하고, 형상 모델링 커널의 연산 기능만을 이용 (OsconCAD) [6]
1-3	형상 모델링 커널의 Persistent Class를 사용하여 File저장 구현한 후 BLOB (Binary Large Object)를 사용하여 데이터베이스에 저장(GSCAD)[18]
1-4	형상 모델링 커널 API를 사용하여, 데이터베이스와 동기화
2	형상 모델링 커널과 STEP의 인터페이스 (본 연구에서는 OpenCascade와 STEP)
2-1	STEP을 저장구조로 하고, 형상 모델링 커널은 Viewer를 위해서만 사용하는 방법
2-2	형상 모델링 커널과 STEP간의 File Translator를 작성 (대부분의 상용 CAD 시스템)
2-3	STEP을 자료구조로 하는 자체 형상 모델링 커널을 작성
3	데이터베이스와 STEP의 인터페이스 (본 연구에서는 ObjectStore와 STEP)
3-1	STEP의 응용 프로토콜을 High Level Logical 설계 과정을 거쳐 데이터베이스 스키마 제작 (HLDAL) [5]
3-2	Geometry 부분을 STEP compatible 파일로 만들어 BLOB를 사용하여 저장 [20]
3-3	STEP의 Class를 이용하여 (멤버로 가지거나, 상속 받거나 하여), 데이터베이스의 클래스로 사용하는 방법[9]

시스템에서의 데이터베이스는 내부 자료구조를 위한 공유의 내부 데이터베이스이고, STEP과 같은 표준 스키마를 바탕으로 만들어진 데이터베이스는, 데이터 공유를 위한 외부 데이터베이스이다. 그런데, 본 연구의 데이터베이스는 Native Storage이면서도 CAD 데이터 공유를 할 수 있도록 구축하므로, 두 가지 측면을 모두 가진다.

본 연구에서는 Jacobson^[15]과 같이 형상 모델링 커널이 할 수 있는 기능은 전적으로 커널을 사용하고, 커널 자체를 개발하는 연구^[14,16]는 아니다. 또한, CAD를 위한 자체 DBMS(Database Management System) 자체를 개발하는 연구^[17]도 아니다. 본 연구는 CAD 시스템 구축을 위해 상용 커널을 사용하고, 데이터베이스는 STEP을 스키마로 하여 상용 데이터베이스에 구축하였다. 즉, 기존 기술들을 사용하여 시스템을 구축하기 위한 인터페이스에 관해 다루었다.

HLDAI(High Level Data Access Interface)^[5]는 응용프로토콜의 AIM(Application Interpreted Model)을 직접 다루지 않고, application-oriented 모델인 APM(Application Program Model)이라는 데이터 모델을 작성하여, SDAI(Standard Data Access Interface)보다 높은 수준의 API를 작성하였다. 본 연구에서는 STEP 스키마를 직접 다루되, 응용프로그램에서 필요한 엔터티들을 선택하여 인터페이스를 작성하였다.

3. OpenDIS - 형상 모델링 커널과 데이터베이스간의 인터페이스

3.1 OpenDIS를 위한 모델링 커널, STEP, DB의 연동

본 절에서는 형상 모델링 커널과, STEP 스키마를 사용한 객체지향 데이터베이스의 기본 저장 구조를 사용하여, 프로토타입 CAD 시스템을 제작할 때, Memory와 Persistent data 간의 동기화를 위한 인터페이스에 대해 설명한다. 개발중인 인터페이스는 기본 저장 구조를 위한 데이터베이스 인터페이스를 제공한다는 의미에서 OpenDIS(OpenCascade Database Interface for STEP storage)라고 부른다.

본 연구에서 다루어야 하는 자료 구조는 크게 STEP 데이터, ObjectStore 데이터, 형상 모델링 커널 데이터 등 세가지가 있으며, OpenDIS는 이들을 연동하기 위해 만들어 졌다. OpenDIS는 데이터베이스와 형상 모델링 커널간의 자료구조 Mapping, 즉 데이터베이스 Schema와 형상 모델링 커널 자료구조를 실시간 매핑한다. OpenDIS는 "개별 엔터티를 매핑하고 번역 하는 캡슐화 된 형상 모델링 함수"라고 정의 할 수 있다.

Table 2. Rule for OpenDIS implementation

매핑대상	Rule의 내용
STEP과DB	STEP과 데이터베이스 스키마는 일대일 매핑된다.
형상 모델링 커널과 STEPDB	형상 모델링 커널의 형상 관련 명령어를 캡슐화 하여 만든 API의 Attribute들은 DB 스키마의 Attribute와 같다. 캡슐화 된 Command와 데이터베이스 인스턴스 생성함수가 함께 불림으로써 동기화 된다.

Table 2는 OpenDIS를 구현하기 위한 기본적인 규칙을 보여준다. 엔터티 연동은 크게 STEP과 데이터베이스 인터페이스, 형상 모델링 커널과 STEP 데이터베이스의 인터페이스로 나눌 수 있다.

데이터베이스와 STEP연동을 위해 STEP의 Class를 멤버로 가지는 ObjectStore클래스를 선언하였다. 기본적으로, 데이터베이스 엔터티는 STEP 엔터티와 일대일 대응된다. 매핑 할 때 네이밍(naming) 규칙은 다음과 같다. 형상 엔터티이면서 Objectstore 엔터티인 것은 STEP과 혼동을 피하기 위해서, 스키마 이름 앞에 접두사 DB_를 붙인다. 비형상 STEP엔터티 이름은 그대로 사용한다. 예를들어, DB_Line, DB_BsplineSurface과 같은 형상 엔터티와, Plate_design_definition과 같은 비형상 엔터티가 있다.

OpenDIS는 형상 모델링 커널과 데이터베이스를 연동하기 위해 만들어진 것으로, 형상 데이터의 생성, 수정 및 데이터베이스 저장과 검색 응용 프로그램을 작성하는데 사용된다.

형상 관련 명령어는 STEP데이터베이스와 연동되기 위해 캡슐화된다. STEP이 데이터베이스 스키마로 사용되어 데이터베이스와 연동된 후, 형상 모델링 커널과 데이터베이스를 연동하는 캡슐화 API가 작성되고, 이 API와 데이터베이스 함수를 함께 부르면, STEP, 형상 모델링 커널, 데이터베이스 세가지가 연동될 수 있다.

OpenDIS는 내부적으로 OpenCascade API함수와 ObjectStore API함수를 갖고 있다. 즉, 한 개의 OpenDIS 함수는, OpenCascade API를 캡슐화한 형상 생성의 변경용 함수와, ObjectStore API를 캡슐화한 데이터베이스로 저장용 함수를 포함한다. 디버깅 용으로 파일 저장도 처리 하였는데, OpenDIS를 통해 작성된 Geometry는 물론 디버깅용 파일에도 저장될 수 있으나, OpenDIS를 통해 작성되지 않은 Geometry는 OpenCascade 함수를 사용하여 파일에만 저장 될 수 있다. Table 3에서 볼 수 있듯이, 캡슐화 된 API의 인수는 데이터베이스 스키마의 Attribute와 같다.

Table 3. Relationship among OpenDIS attribute, database attribute, and STEP attribute

OpenDIS Function	DB Related entity	STEP Related entity
<pre>void ODIS_Create_Line(int ID, DB_cartesian_point pnt, DB_direction vector) { TS_Commands::Create_Line (ID, pnt, vector); new DB_Line(ID, pnt, vector); } </pre>	<pre>Class DB_Line { int ID; DB_cartesian_point pnt; DB_direction vector; } </pre>	<pre>ENTITY line SUBTYPE OF (curve); pnt : cartesian_point; dir : vector; END_ENTITY; </pre>

TS_Commands::Create_Line와 DB_Line 클래스의 컨스트럭터에 들어가는 인수가 같다. 두개의 함수, 즉, 형상 모델링 커널의 메모리 자료구조를 생성하는 함수와 데이터베이스 인스턴스를 생성하는 함수는 함께 불림으로써 동기화가 된다.

상업용 CAD 시스템들은 각각 고유의 API를 가지고 있으며, Unigraphics의 OpenAPI, ProEngineer의 Pro/Toolkit등이 그 예이다. 본 연구에서 개발된 DINA-CAD 시스템의 API는 OpenDIS이다. 기존 MCAD (Mechanical CAD) API들은 데이터 파일을 액세스하고, 데이터를 생성하는 모듈을 제공하는데, OpenDIS는 형상 모델링 커널의 함수와 데이터베이스 API가 조합되어 있다.

3.2 OpenDIS의 구축 순서

- (1) Instance가 가능한 엔터티 선택
- (2) Driver 클래스를 작성
- (3) Driver들을 Command로 등록
- (4) Command와 DB 함수를 묶어 OpenDIS를 작성
- (5) User Level Programming을 통해 DINA-CAD를 완성

첫번째 단계는 Instance가 가능한 엔터티를 결정하는 단계이다. 각각의 Domain과 응용프로그램 특성에 맞도록 Instance가 가능한 엔터티들을 STEP 엔터티 중에서 결정한다. 본 시스템에서는 선체 설계 분야에서 구현하였는데, Midship 섹션 모델을 대상으로 하여, 모델을 구현 할 수 있는 최소한의 엔터티만을 대상으로 하여, Table 4와 같은 엔터티들로 결정하였다.

OpenDIS로 개발 되지 않은 엔터티에 대해서, 데이터베이스에 저장하는 작업을 하기 위해서는 Translator를 따로 개발해야 한다.

Table 5는 persistent 데이터와 Transient 데이터의 예제를 보여준다. OpenCascade의 클래스 형태로 된 엔터티TopoDS_Line이나, gp_line 등은 형상 모델링 커널에서는 사용하지만, Persistent 데이터로 관리 되지는 않는다. 즉, Topo_DSLine이나 gp_line 엔터티는,

Table 4. Entity which has an OpenDIS function

구분	엔터티 이름
Curve	PolyLine, Circle, Line
Surface	Bspline_Surface, Bezier_Surface
Solid	Box, Cylinder, Cone
STEP	Plate_design_definition non_manifold_surface_shape_representation

Table 5. Examples of persistent vs. transient

Persistent	Transient
Plate_design_definition DB_Line	Plate_design_definition TopoDS_Line, gp_line, Line

필요시 DB_Line 엔터티의 OpenDIS를 이용하여 저장되거나, 아니면 메모리에서 활용된 후 소멸된다. Plate_design_definition은 Transient이면서 Persistent이다.

두 번째 단계로, 각각의 Geometry 엔터티의 생성, 수정을 할 수 있도록, OpenCascade 함수를 캡슐화한 Driver 클래스를 작성하였는데, OpenCascade의 OCAP (OpenCascade Application Framework) 모듈을 사용하였다.

Driver를 작성할 때 고려할 점 두 가지 중 첫째는, 외부로 공개되는 Attribute들은 데이터베이스를 인스턴스 할 때의 Attribute와 같게 맞춘다. Driver를 사용해서 응용 프로그램을 작성하는 사람들은, 내부 구조는 고려하지 않고, 같은 Attribute를 사용해서, 형상 모델링 커널과 데이터베이스 인스턴스를 만들 수 있다. Driver를 작성할 때 고려할 다른 한가지는, Geometry 관련 함수를 결정하는 것이다. Line Driver에서 핵심 Geometry 함수는 BrepBuilderAPI_Make_Line을 사용하였다.

세 번째 단계는, 작성된 Driver들을 Command로 등록한다. Driver는 CAD 시스템의 API 수준 함수라고 볼 수 있고, Command는 사용자 수준의 함수라고 볼 수 있다. Table 3에서 사용된 Create_Line 함수가, TS_Command 클래스의 멤버 함수 중의 하나이다.

Table 6. Process of geometric modeling transaction

	작업	ObjectStore 함수
초기화	DB 초기화	objectstore::initialize();
	Collection 초기화 TypeSpec 마련 DB 열기	os_collection::initialize(); new os_typespec("db_line"); os_database::open("DB");
모델링	트랜잭션 시작 Root 찾기	OS_START_TXN(tx1) db1→find_root
	OpenDIS 실행	ODIS_Create_Line 또는 ODIS_Modify_Line 또는 ...
	생성한 형상 모델링 커널 자료구조를 바탕으로 속성 세팅 및 가시화용 형상 모델링 커널 함수의 실행	
	트랜잭션 끝	OS_START_TXN(tx1)
끝	DB 닫기	os_database::close ("DB");

네번째 단계로, 작성된 Command와 데이터베이스관련 함수를 묶어, Table 4의 ODIS_Create_Line 함수와 같은 OpenDIS함수를 만들었다.

데이터베이스 관점에서 형상 모델링 작업은 Table 6과 같이 몇 개의 단계로 나누어 볼 수 있다. 모델링을 하기 위해 파일을 불러오는 대신, 데이터베이스를 초기화 하고, 생성할 타입에 대한 TypeSpec을 마련한 후 작업 데이터베이스를 열고, 작업 공간을 가져온다. 모델링 작업을 시작하기 위해, 데이터베이스 트랜잭션을 시작한다. 데이터베이스의 루트를 기준으로 해서, 인스턴스들을 list, set 등의 collection으로 저장한다. OpenDIS는 명령어 레벨에서도 Undo, Redo가 가능하다. 모델링 작업은 필요한 OpenDIS함수를 실행하는 작업으로 이루어진다. 작업중에 설계자가 save 명령어를 주면, 하나의 데이터베이스 Transaction이 끝나게 된다. 모델링 작업을 완전히 마쳤으면 데이터베이스를 닫는다.

마지막 단계로, GUI를 구축하여 DINA-CAD 시스템을 완성한다. Fig. 4은 DINA-CAD Client 응용 프로그램의 클래스 구조를 나타낸다. 이 구조는 캡슐화된 드라이버 클래스, OpenCascade 커널 형상관련 클래스, GUI 관련 클래스 세 부분으로 되어 있다. Plane과 BSpline두개의 드라이버를 예제로 들어서, 전체적인 클래스들 간의 관계를 보여준다.

Driver 클래스는 OCAF 모듈 중의 Tfunction_Driver로부터 상속받아서 명령어 모듈들을 작성한다. AP218 스키마에 따른 인스턴스들을 생성, 수정, 관리, Create, Modify, Undo, Redo 저장하기 위한 인터페이스이다.

Driver 클래스는 OpenCascade의 모델링 함수를 사용하여 멤버함수를 작성한다. 즉, TFunction_Plane_PlaneDriver의 Geometry 관련 함수는 Opencascade

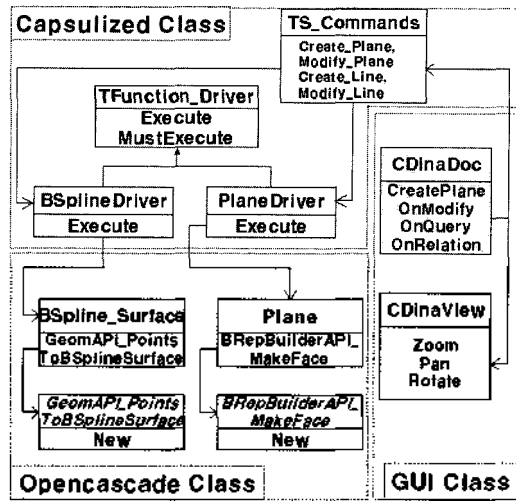


Fig. 4. Relationship of OpenDIS classes.

API함수이다. Plane클래스와 관련된 형상 모델링 함수가 BRepBuilderAPI_MakeFace라는 것을 나타낸다. MFC로 작성된 GUI 관련 View 클래스에서는 Zoom, Pan, Rotate 등의 기본 기능을 수행한다. 속성 검색을 지원하기 위한 대화상자 등의 클래스도 여기에 해당한다.

클래스 동작 메커니즘은 다음과 같다. 사용자 GUI를 통해 입력하면, CDinaOcfaView 클래스에서 GUI 관련 작업을 처리하고, CDinaOcfaDoc 클래스에서 User 수준의 함수를 실행한다. 이때, 모든 형상 생성과 변경명령어를 관장하는 클래스인 TS_Command 클래스가 인스턴스 되면서, TS_Command중 Geometry관련 멤버함수 실행이 실행된다. 이것은 Tfunction_driver로부터 상속받아서 작성한, 각 Geometry당 한 개씩 있는 드라이버를 인스턴스 하고, "Create_" 멤버함수 실행한다. 저장 시에는 실행이 끝난 명령어가 생성한 모델들을 모두 저장하고, 모델 변경 시에는 해당 모델의 드라이버를 다시 호출하여 "Modify_" 멤버 함수를 실행한다.

Create와 관련된 인터페이스에서는 형상 모델링 커널 함수로 엔터티를 생성한 후 데이터베이스에 저장할 수 있는 함수를 제공하고, Get과 관련된 인터페이스는 데이터베이스에서 가져온 후 알맞은 엔터티로 GUI 상에 가시화 한다.

3.3 OpenDIS를 활용한 프로토타입 CAD 구현 예

Fig. 5는 DINA-CAD의 Client GUI를 보여준다. 기본적인 형상 Primitive의 모델링 기능들을 구현하였고,

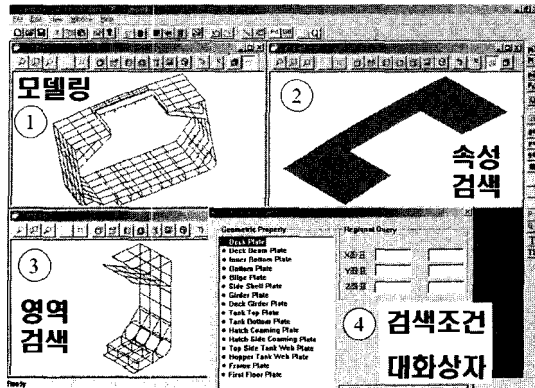


Fig. 5. Graphic user Interface of DINA-CAD Client.

Undo, Redo 등의 기능을 갖고 있다. 실험을 위한 모델로 25,000개의 엔터티, 400여개의 파트로 이루어져 있는, 선박의 Midship Section을 사용하였으며, Fig. 5-1은 DINA-CAD 상에서 모델링 한 그림이다.

기본적으로 데이터베이스 서버에 저장된 데이터는 한 개의 매우 큰 파일과 개념적으로 같다. 데이터베이스 서버는 파일의 모든 부분을 메모리로 읽어들이지 않고 필요한 부분만을 가져 올 수 있게 하며, 다른 클라이언트들이 동시에 액세스 하는 것을 제어 한다.

Fig. 5-2는 Midship Section 전체 모델중에서 속성을 사용한 검색을 통하여, 대상 모델의 맨 윗쪽 부분에 해당하는 Deck Plate를 가져왔다. Fig. 5-3은 영역검색의 예제를 보여주는 것으로, Fig. 5-1에 보인 전체 모델 중에서 원하는 영역을 XYZ 좌표 값으로 주었을 때, 만들 수 있는 가상의 블록 내에 있는 부분을 가져 온 예제이다. Fig. 5-4는 검색 조건을 입력하기 위한 대화상자이다.

4. 결 론

본 논문에서는 선박의 협력 설계 환경을 위한 데이터베이스 기반 CAD 시스템의 구조를 위한 요소 기술 중, 데이터베이스와 형상 모델링 커널간의 인터페이스를 다루었다. 개발된 CAD 시스템의 저장 구조는 Native Storage이면서도 외부 시스템과 CAD 데이터 공유를 할 수 있도록 활용될 수 있다. CAD 시스템 구축을 위해 상용 커널을 사용하고, STEP을 스키마로 한 상용 객체 지향 데이터베이스를 저장 구조로 사용하였다. 이때, 형상 모델링 커널 함수를 캡슐화하고 이를 데이터베이스 API와 연동하여, 설계 작업 시 형상 모델링 커널의 데이터와 데이터베이스의 데이터 간에 동

기화가 유지되도록 하였다.

본 논문에서는 데이터베이스 스키마를 AP218을 사용해서 작성하여, 조선분야 선체 설계에 적용하였으나, STEP의 다른 응용 프로토콜을 활용함으로써 기계분야 등 다른 분야에도 확장 응용 될 수 있다. 본 논문에서 제시한 방법은, 데이터베이스 기반 CAD 시스템을 구현하기 위해 형상 모델링 커널과 데이터베이스의 인터페이스를 제시한 프로토타입 시스템이며, Change Propagation, 복잡한 Relation의 관리 문제, 상용 CAD 시스템들에서의 데이터베이스 공유 등은 향후 더 구체적으로 연구되어야 한다.

참고문헌

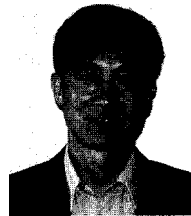
1. Andreas Meier, "Applying relational database techniques to solid modeling," *Computer-Aided Design*, Vol. 18 No. 6, July 1986.
2. Hader, Th. Hubel, Ch. and Mitschang, B., "Information Structures and Database Support for Solid Modelling," *Proc. of Int. Workshop on Theory and Practice of Geometric Modelling*, Blaubeuren, Germany, 1988.
3. Kim, W. "Object-Oriented database support for CAD," *CAD*, Vol. 22, No. 8, October 1990.
4. Sriram, R. D., Albert Wong and Li-Xing He, "GNOMES: an Object-oriented nonmanifold geometric engine," *CAD*, Vol. 27 No. 11, pp. 853-868, 1995.
5. Farhi Marir, Ghassan Aouad, Grahame Cooper, "OsconCAD: A model-based CAD system integrated with computer applications," *Icon*, Vol. 3, 1998.
6. Ozdogan Karacali, "An introduction to object design technology with computer graphics to support concurrent engineering," *Modelling and graphics in science and technology*, Springer, 1995.
7. Stig Ulfby, Steinar Meen, Jorn Oian, "Tornado: A Data-base management system for graphics applications," *IEEE Computer Graphics and Application*, pp. 71-79, 1982.
8. 손경모 외, "분산환경에서 표준모델(STEP)을 이용한 내용검색," 한국 CAD/CAM학회 논문집, 제4권, 제4호, pp. 285-294, 1999.
9. Krebs, T. and Luehrsen, H., "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., pp. 131-142, 1995.
10. Loffredo, "Efficient Database Implementation of EXPRESS Information Models," *RPI*, Ph.D. Thesis, 1998.
11. Jianhong Sun, Martin Hardwick, "Building an integrated large scale STEP database for virtual enterprises," *Proceedings of ASME DETC99/DFM*, Sept.

- 1999, Las Vegas, USA.
12. Shin, Y. J. and Han, S. H., "Data Enhancement for Sharing of Ship Design Models," *Computer Aided Design*, Vol. 30, No. 12, pp. 931-941, 1998.
 13. 황호진, 한순홍, "선급 검토를 위한 중앙단면 STEP AP218 구조 모델의 생성," 대한조선학회 '2000 춘계 학술발표회 논문집, pp. 77-82, 2000년 4월.
 14. 박광필 외, "STEP을 근거로 한 선체화물창부 구조 데이터 모델에 관한 연구," 한국CAD/CAM학회 논문집, 제4권, 제4호, pp. 381-390, 1999.
 15. Ketill Jacobsen, "Development of a ship modeling application using CAS.CADE, an object oriented software development environment," ICCAS, 1997.
 16. 정연찬, 박준철, "CAD/CAM응용 소프트웨어 개발을 위한 형상 커널 개발," 한국 CAD/CAM학회 논문집, 제6권, 제4호, pp. 271-276, 2001.
 17. ISO TC184/SC4/WG3 N799, ISO/CD 10303-218 Product data representation and exchange - Application protocol - Part 218: Ship structures, September 1999.
 18. GRAD Inc., <http://www.gradinc.com/>.
 19. Jianhong Sun, Martin Hardwick, "Building an Integrated Large Scale STEP Database For Virtual Enterprises," ASME DETC99/DFM-8959, 1999. 9.
 20. Susan D Urban 외, Integrated product data environment: data sharing across diverse engineering applications, Int. J. CIM, Vol. 12, No. 6, pp. 525-540, 1999.



김 준 환

1995년 한국과학기술원 정밀공학과 학사
 1998년 한국과학기술원 기계공학과 석사
 1998년-현재 한국과학기술원 기계공학과 박사과정
 관심분야: CAD 모델 표준(STEP), CAD 데이터베이스, 네트워크 CAD



한 순 홍

1977년 서울대 조선공학과 학사
 1979년 서울대 조선공학과 석사
 1985년 영국 Newcastle대 석사
 1990년 미국 Michigan대 박사
 1979년-1992년 해사기술연구소(현재 기계연구원)
 1993년-1995년 한국과학기술원 자동화 및 설계공학과
 1996년-현재 한국과학기술원 기계공학과 교수
 관심분야: CAD 모델 표준(STEP), Intelligent CAD, 설계전문가시스템, 형상모델링 커널