

## FlexDesigner: 계층적으로 모듈화된 구조의 객체 지향 방식 비다양체 모델링 커널

이강수\*, 이권우\*\*

### FlexDesigner: Object-Oriented Non-manifold Modeling Kernel with Hierarchically Modularized Structure

Kang-Soo Lee\* and Kunwoo Lee\*\*

#### ABSTRACT

Conventional solid or surface modeling systems cannot represent both the complete solid model and the abstract model in a unified framework. Recently, non-manifold modeling systems are proposed to solve this problem. This paper describes FlexDesigner, an open kernel system for modeling non-manifold models. It summarizes the data structure for non-manifold models, system design methodology, system modularization, and the typical characteristics of each module in the system. A data structure based on partial-topological elements is adopted to represent the relationship among topological elements. It is efficient in the usage of memory and has topological completeness compared with other published data structures. It can handle many non-manifold situations such as isolate vertices, dangling edges, dangling faces, a mixed dimensional model, and a cellular model. FlexDesigner is modularized hierarchically and designed by the object-oriented methodology for reusability. FlexDesigner is developed using the C++ and OpenGL on both SGI workstation and IBM PC.

**Key words** : CAD, non-manifold, modeling kernel, NURBS, modularization, object-oriented methodology

#### 1. 서 론

공학 분야에서 곡면 모델링 시스템 및 솔리드 모델링 시스템은 점차 그 사용 범위가 넓어지고 있다. 현재 산업계에서 많은 모델링 시스템이 제품의 형상 설계, 해석, 가공 등에 사용되고 있으나 기존의 솔리드 모델링 시스템은 개념 설계 단계와 공학 해석 시 사용되는 와이어프레임 모델이나 박판 모델 같은 축약 모델을 표현할 수 없으며, 다양한 형태의 모델을 지원해야 하는 통합 시스템에 사용될 수 없어 사용 범위가 제한된다. 즉, 기존의 솔리드 모델링 시스템은 개념 설계 단계에서부터 사용되지 못하고 형상이

완전히 결정된 최종 설계 단계에서만 적용될 수 있다. Fig. 1은 대표적인 기계 설계 문제로 축에 기어와 베어링이 결합된 동력 전달 시스템을 설계하는 과정을 보여 준다. (a)는 개념 설계 초기 단계로 단지 직선과 원을 이용하여 기어와 축, 베어링을 표현하고 있으며, (b)는 동력을 전달하기 위한 기어의 기본 설계가 되어, 모듈, 잇수, 폭 등 기어 관련 데이터가 정해진 상태를 나타낸다. 기어의 기본 설계가 끝난 후, 축에 대한 기본 설계를 하여 (c)와 같이 동력을 전달할 수 있는 축의 최소 지름을 결정하고 베어링 설계를 한다. 최종적으로 (d)와 같이 조립, 가공 등 제반 요소를 고려해 상세 설계가 끝나 축과 기어의 상세한 형상이 결정되며, 축과 기어를 결합하는 키와 로크 와셔, 로크 너트 등과 같은 관련 부품이 결정된다. Fig. 1에서 볼 수 있듯이 (a)와 같은 개념

\*학생회원, 서울대학교 기계설계학과

\*\*중신회원, 서울대학교 기계설계학과

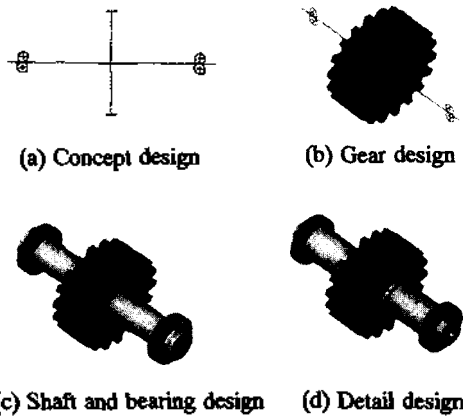


Fig. 1. Example of designing shaft, gear and bearing system.

설계 단계에서는 대부분 직선, 원, 곡선과 같은 선의 형태(wire)로 표현되며, (b)와 같이 설계가 진행되는 단계에서는 와이어와 솔리드가 혼합된 형태로 표현된다. 그리고 (c)와 같이 기본 설계가 성숙된 단계와 (d)와 같이 상세 설계가 끝난 상태에서는 완전히 솔리드로만 표현된다. 설계 작업을 곡면 모델링 시스템으로 할 경우에는 (a) 과정을 지원할 수 있으며, 솔리드 모델링 시스템으로 할 경우에는 (c), (d) 과정만을 나타낼 수 있으며, (b) 과정은 곡면 모델링 시스템이나 솔리드 모델링 시스템으로 표현할 수 없어 비다양체 모델링 시스템이 필요하게 된다. 빌딩 설계 시에도 유사한 예가 Sriram<sup>11)</sup>에 나와 있으며, 해석 시 솔리드 모델과 축약 모델이 함께 필요한 예가 이상현<sup>12)</sup>에 나와 있다.

그래서, 솔리드 모델과 축약 모델을 동시에 표현 하려는 공학적 필요성을 만족시키기 위한 연구가 행해졌으며<sup>37)</sup>, 그 결과 비다양체 모델링 시스템이란 개념이 생겼다. 유클리디언 포복체(Euclidean cell complex)를 0-차원 포복(0-cell)인 꼭지점, 1-차원 포복(1-cell)인 모서리, 2-차원 포복(2-cell)인 면, 3-차원 포복(3-cell)인 영역으로 구분하는데, 형상 모델링 용어에서 다양체 물체란 2-차원 포복인 곡면을 경계로 하는 부피를 가진 물체, 즉, 일반적으로 솔리드 모델을 지칭한다. 이에 반해 비다양체 물체란 다양체 물체가 아닌 것을 말하며, 균일하지 않은 차원의 형상도 포함되어 복합 다양체라고도 한다. 비다양체 표현에서는 서로 다른 차원의 모델이 섞인 상태를 표현할 수 있으므로 공학 및 비공학 분야에서 그 응용 범위가 넓다. 현재 제시되고 있는 여러 가지 비다양체 자료 구조에 대한 비교는 이상현<sup>17)</sup>에 나타나

있으며 본 연구에서는 이상현의 자료 구조를 채택하여 비다양체 모델링 커널인 FlexDesigner를 개발하였다<sup>18, 12)</sup>.

FlexDesigner는 SGI 워크스테이션에서 C++을 사용하여 개발되었으며, 그래픽 관련 부분은 OpenGL을 사용하였다. 또한 이 시스템을 Windows 95/NT 환경의 IBM PC 펜티엄에 이식하였다. 개발된 시스템에서는 135개의 API와 곡면 모듈, 위상 모듈을 비롯한 다양한 작업에서 180개의 클래스가 구축되어 있다<sup>18, 12)</sup>. 사용자는 커널 시스템에서 제공하는 API 또는 클래스(class)의 멤버 함수를 이용하여 응용 프로그램을 개발할 수 있다.

또한, FlexDesigner의 구조는 C++를 이용한 객체 지향 방법으로 설계되었으며, 각 구성 요소들이 객체로 표현되어 있어 객체의 상속성, 추상화, 캡슐화가 구현되었다. 그리고 시스템은 모듈화된 계층 구조로 설계되어 모듈간에 종속성 및 독립성을 명확히 하여 사용을 편하게 하고 확장이 쉽게 하였다. FlexDesigner의 구조 및 구성 모듈에 대해서는 4, 5, 6장에서 자세히 설명된다.

FlexDesigner는 공학 분야에서 다음과 같은 물체를 모델링할 수 있다.

- 일반 솔리드 모델링 시스템에서 모델링 가능한 물체
- 일반 곡면 모델링 시스템에서 모델링 가능한 물체
- 개념 설계 및 기본 설계 등 인간의 관념을 표현해야 할 때
- 공학 해석 분야에서 일부가 축약 모델로 표현될 때, 모델링 단계에서부터 해석을 고려해 모델링할 수 있다.
- 여러 차원이 혼합된 물체(mixed dimensional object)
- 모서리를 공유하는 물체(edge shared object)
- 꼭지점을 공유하는 물체(vertex shared object)
- 여러 가지 물질로 구성된 물체(partitioned object)

Fig. 2는 FlexDesigner를 이용하여 모델링한 예로 기존의 솔리드 모델링 시스템에서는 표현할 수 없는 비다양체 물체를 보여 주고 있다. (a), (b), (c)는 여러 차원의 요소로 구성된 물체를 보여준다. (d)는 모서리를 공유하고 있는 물체를 나타내고, (e)는 꼭지점을 공유하고 있는 물체를 나타낸다. (f)는 아래쪽의 육면체와 위 부분의 실린더 사이의 경계를 구분하는 면이 존재하는 것으로, 육면체와 원기둥이 서로 다른 물질로 되어 있을 때 또는 FEM에서 구분된 부분마다 서로 다른 요소를 사용할 때 모델링하는 방법이다.

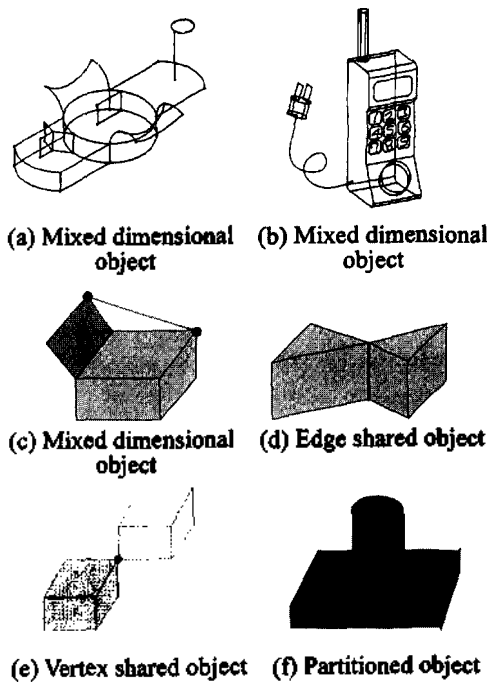


Fig. 2. Examples of non-manifold objects.

본 논문은 비다양체 모델링 커널 시스템인 Flex-Designer의 개발에 대한 것이다. 구체적으로는 비다양체 모델을 표현하기 위한 자료 구조와 커널 시스템을 개발할 때 계층적으로 모듈화 시키는 커널 시스템 구조 설계 방법론 그리고 커널 시스템을 구성하는 각 모듈을 객체 지향 방식으로 설계하는 모듈 설계 방법론을 제시한다. 이 외에도 FlexDesigner가 커널 시스템으로 가져야 할 다양한 모델링 기능과 API 함수에 대해서도 설명하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구, 3장에서는 본 논문에서 사용한 시스템 자료 구조, 4장에서는 간략화 된 FlexDesigner의 시스템 개요에 대해 설명하였다. 5장에서는 객체 지향 방법을 이용해 설계된 FlexDesigner의 클래스 구조, 6장에서는 FlexDesigner에서 사용할 수 있는 다양한 모델링 작업, 7장에서는 FlexDesigner를 사용해 응용 프로그램을 개발할 때 필요한 API에 대한 설명이 있고, 8장에서는 모델링한 예를 보였다.

## 2. 관련 연구

본 논문에서 새롭게 제시하는 부분은 비다양체 모델링 커널의 기본 구조, 모델링 커널 시스템을

개발할 때 계층적으로 모듈화 시키는 커널 시스템 구조 설계 방법론, 그리고 커널 시스템을 구성하는 각 모듈을 객체 지향 방식으로 설계하는 모듈 설계 방법론이다. 따라서 이 장에서는 비다양체를 표현하는 자료 구조에 대한 관련 연구와 모델링 커널 시스템을 개발하는 방법에 대한 관련 연구를 살펴 본다.

현재 모델링 커널로 사용되고 있는 상용 시스템으로는 ACIS, Parasolid, DESIGNBASE, SHAPES, CAS.CADE 등이 있다. 이 중 ACIS와 SHAPES, CAS.CADE는 비다양체 모델링 기능이 있으며 다른 시스템도 점점 비다양체 모델링 기능을 지원하는 추세이다. 비다양체 모델링 시스템 개발 시 가장 기본이 되는 자료 구조에 대한 연구는 Weiler<sup>[3]</sup>, Choi<sup>[4]</sup>, Yamaguchi<sup>[5]</sup>, Rossignac<sup>[6]</sup>, 이상현<sup>[7]</sup> 등에 의해 행해졌으며, 이에 대한 비교는 이상현<sup>[7]</sup>에 잘 나타나 있다. 이상현<sup>[7]</sup>의 비다양체 자료 구조는 솔리드 모델을 표현하는 half-winged edge 자료 구조를 비다양체 영역까지 확장 시켰으며 부분 위상 요소(partial topological element)를 사용한 점이 특색이다. 이 자료 구조는 효율적인 자료 저장 방식을 사용하여 다양한 형태의 비다양체 모델, 즉, 여러 차원이 혼합된 물체(mixed dimensional object), 모서리를 공유하는 물체(edge shared object), 꼭지점을 공유하는 물체(vertex shared object), 여러 가지 물질로 구성된 물체(partitioned object)를 표현할 수 있으므로, 본 연구에서는 이를 사용하여 비다양체 모델링 커널인 Flex-Designer를 개발하였다<sup>[8,12]</sup>.

비다양체 모델링 시스템을 개발한 사례도 발표되고 있다. Sriram<sup>[11]</sup>은 Rossignac<sup>[6]</sup>의 SGC(Selective Geometry Complex)를 사용하여 객체 지향 방식의 GNOMES란 비다양체 모델링 시스템을 개발하였으며, Cavalanti<sup>[13]</sup>는 역시 SGC를 사용해 공간 분할 방식으로 비다양체 모델링 시스템을 개발하였다. 본 연구는 Sriram과 객체 지향 방식을 적용한 점은 비슷하나, Sriram이 SGC를 사용한 반면 본 연구에서는 이상현의 부분 위상 요소를 사용한 자료 구조를 채택했으며, Sriram은 개발된 시스템의 구조 설계면에서의 고려가 없으나 본 논문에서는 모듈화된 계층 구조를 제시한 점이 다르다. 또한 Sriram이 제시한 GNOMES는 커널 목적으로 개발되지 않아 응용 프로그램에 제공해야 할 API 함수에 대한 고려가 없다는 점에서 본 연구와 다르다.

그 밖의 여러 가지 모델링 기능에 대한 관련 연구는 본문 중에서 언급하겠다.

### 3. 비다양체 모델을 지원하는 자료 구조의 개요

본 연구에서는 비다양체 모델을 표현하기 위해 이상현<sup>[7]</sup>이 제시한 자료 구조 방식을 사용하였다. 이 자료 구조는 Fig. 3과 같이 부분 위상을 이용한 간결한 형태의 계층 구조이며, 효율성을 유지하면서 중복되는 정보를 최대한 줄였다.

Fig. 3에서 사용된 위상 요소로는 꼭지점(VERTEX), 부분 꼭지점(PARTIAL VERTEX), 모서리(EDGE), 부분 모서리(PARTIAL EDGE), 루프(LOOP), 면(FACE), 부분 면(PARTIAL FACE), 셸(SHELL), 영역(REGION)이 있으며, 부분 위상 요소인 부분 꼭지점, 부분 모서리, 부분 면을 사용한 점이 특징이다. 부분 면, 부분 모서리 및 부분 꼭지점은 각각 면과 영역, 모서리와 면, 꼭지점과 모서리 간의 인접 관계를 나타내는 데 사용된다. 또한 솔리드 모델링 시스템과 유사하게 꼭지점, 모서리, 면은 각각 유클리드 공간 상의 위치를 나타내는 점(point), 곡선(curve), 곡면(surface)에 대한 정보를 갖고 있다. Fig. 3의 자료 구조는 외꼭지점 루프(single VERTEX LOOP), 와이어 모서리(wireframe EDGE), 외꼭지점 셸(single VERTEX SHELL)을 예외적인 경우로 취급하여 수용할 수 있게 하였다. 외꼭지점 셸이란 공간 상에 하나의 고립된 꼭지점만이 존재하

며 자체적으로 셸을 구성하는 것이고, 와이어 모서리는 어떤 면과도 인접하고 있지 않은 모서리이며, 외꼭지점 루프는 면 위에 하나의 고립된 꼭지점이 존재해 이 꼭지점만으로 내부 루프를 구성하는 경우이다. 본문 중 혼동을 막기 위해 위상 요소의 영어 표기는 모두 대문자로 하였으며, 위상 요소에 대한 클래스 정의는 첫 자만 대문자로 나머지는 소문자로 하였다.

### 4. FlexDesigner 시스템 계층 구조 개요

FlexDesigner는 C++을 사용하여 재사용과 모듈화, 보수 유지에 우수한 객체 지향 소프트웨어 개발 방법론(Object-oriented software development methodology)에 따라 설계되었다. 객체는 상속성, 추상화, 캡슐화의 특성을 지니고 있고, 이러한 특성을 이용하여 소프트웨어를 개발하는 방법은 다음과 같은 장점을 가지고 있다<sup>[14]</sup>.

- 객체 지향 설계는 객체 지향 언어(C++)의 장점을 살릴 수 있다.
- 객체 지향 설계는 프로그램이 작게 되며 재사용성이 뛰어난 코드를 생산한다.
- 객체 지향 설계는 변화에 유연하고 탄력적인 소프트웨어를 생산한다.
- 객체 지향 설계의 모듈화는 여러 개발 팀이 독립적으로 시스템을 개발할 수 있게 한다.
- 객체 지향 방법으로 설계된 시스템은 설계자, 시스템 분석자, 사용자가 이해하기가 쉽다.

객체 지향 설계 방법에서는 적절한 객체의 선정, 객체들간의 계층 구조 설계, 객체의 속성과 메소드 설계 등이 중요한 사항이며 FlexDesigner에서 적용된 구체적 방법은 5장에서 구체적으로 기술될 것이다.

Fig. 4는 FlexDesigner 시스템의 간략화된 구조를 보여 준다. 커널에는 데이터베이스 관리(Database Manager), 기하 클래스(Geometry Classes), 위상 클래스(Topology Classes), 모델링 작업(Modeling Operations), API 함수(API Functions)가 있고, 보조적인 역할을 하는 인터페이스 모듈이 있다. 데이터베이스 관리는 모델링하는 물체의 데이터를 자료 구조에 맞게 관리하기 위해, 기하 클래스는 물체를 공간 상에서 표현하기 위한 곡선과 곡면을 정의하기 위해, 위상 클래스는 비다양체 물체를 표현하기 위해 구현된 것이다. 그리고 인터페이스 모듈은 개발된 커널 시스템을 사용하기 위한 코멘드 입력 방식의 사용자 인터페이스 부분과 그래픽 하드웨어 인터페이스 부

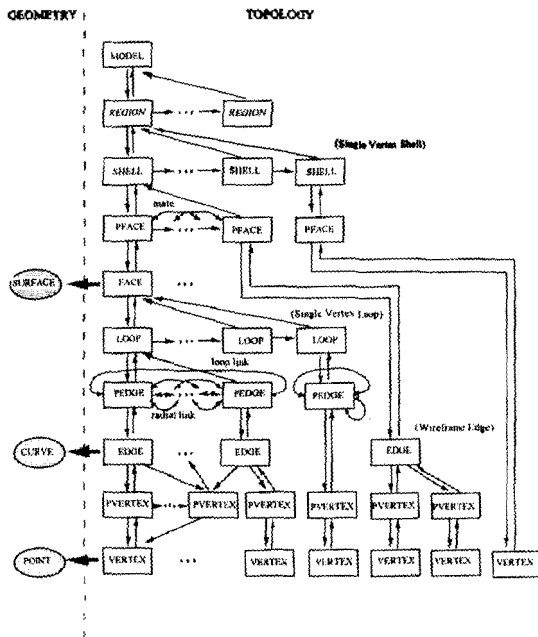


Fig. 3. Data structure for non-manifold models<sup>[7]</sup>.

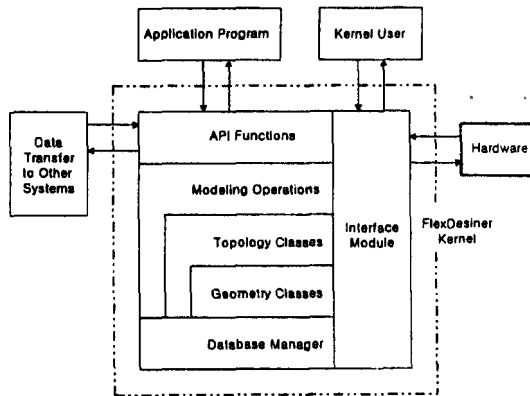


Fig. 4. Overview of FlexDesigner kernel system.

분이다. API 함수로 FlexDesigner 커널을 이용하여 응용 프로그램을 개발할 수 있으며, 또한 이 때 FlexDesigner가 객체 지향 방식의 클래스로 구현되어 있으므로 각 클래스를 직접적으로 이용할 수도 있다. API 함수 중 IGES 파일로 출력하는 함수를 이용해 FlexDesigner에서 모델링한 데이터를 타 시스템에 전달할 수가 있다.

5. 객체 지향 방식의 FlexDesigner 커널 설계

이 장에서는 FlexDesigner의 구조 및 설계 방법에 대해 자세히 설명한다. FlexDesigner는 Fig. 5와 같이 모듈화된 계층 구조로 개발되었으며 커널에 해당되는 부분은 중요한 구성 모듈로는 기본 모듈, 곡선 모듈, 곡면 모듈, 위상 모듈, 상위 모델링 모듈, API 모듈 등이 있다. 기본 모듈에는 벡터, 행렬 등에 대한 계산과 배열을 저장하는 방법이 구현되었고, 곡선 모듈에는 직선, 원호, 원, 자유 곡선 등을 표현하는 방법이 구현되었으며, 곡면 모듈에서는 평면, 자유 곡면 등을 표현하는 방법이 구현되었다. 기본 모듈, 곡선 모듈, 곡면 모듈을 합하여 기하 모듈을 구성한다. 위상 모듈에는 비다양체 자료 구조 방식이 구현된 부분이다. 모델링 모듈에는 하위 모델링 기능인 오일러 작업과 상위 모델링 기능인 기본 형상 생성 작업, 스윙핑 작업, 박판 모델 생성 작업, 와이어프레임 모델 생성 작업, 블리안 작업, 라운딩 작업, 부분 수정 작업, UNDO/REDO 작업 등이 있으며 또한 IGES 파일을 통해 생성된 모델을 타 시스템에 전달하는 기능도 있다. 그리고 API 모듈에는 커널을 이용하여 응용 프로그램을 개발할 수 있도록 다양한 함수가 구현되었다. 모델링 커널을 보조하는 기능을 하는 인터페이스 모

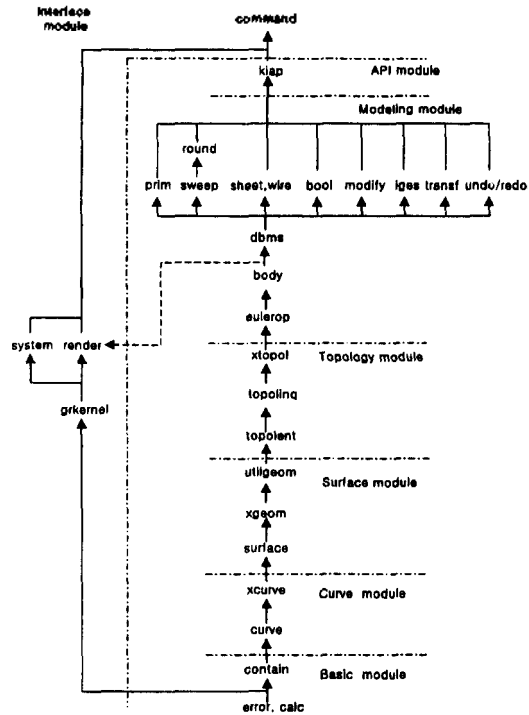


Fig. 5. Hierarchical structure of the system.

듈이 있으며, 인터페이스 모듈에는 사용자 인터페이스와 그래픽 하드웨어 인터페이스 기능을 한다. 코멘드 방식의 인터페이스를 구현하여 개발된 커널을 사용하고 그 기능을 검증할 수 있으며, 그래픽 처리는 OpenGL을 이용하였다. FlexDesigner는 SGI 워크스테이션과 IBM PC에서 같이 사용될 수 있으며, UNIX 환경의 SGI 워크스테이션에서는 X window 라이브러리를 사용하고, Windows 95/NT 환경 하의 IBM PC에서는 Win32 라이브러리를 사용하였다.

각 모듈이 의미하는 바는 Table 1과 같다. Fig. 5에서 화살표는 하위 모듈에서 상위 모듈로의 방향성을 의미하며 종속적인 계층 구조를 나타낸다. Fig. 5에서 상위 모듈은 하위 모듈을 포함하고 있다. 즉, 상위 모듈을 사용하려면, 하위 모듈까지 같이 포함해야 한다. 곡면 모듈을 사용하려면 곡선 모듈을 포함해야 하며, 위상 모듈을 사용하려면 곡면 모듈을 포함해야 한다. 그리고 모델링 작업 모듈에서 수평 방향으로 나열된 각 모델링 작업은 종속적인 관계가 아니라 서로 독립적으로 존재하는 것을 의미한다. 기본 형상 생성 작업, 스윙핑 작업, 박판 모델/와이어프레임 모델 생성 작업, 블리안 작업, 부분 수정 작업, IGES 출력 작업, 변환 작업, UNDO/REDO 작업

Table 1. Explanation of FlexDesigner modules

Module	Sub-module	기능
Basic	error	에러를 관리하는 기능
	calc	벡터, 배열 등 기본적인 계산 기능
	contain	배열 설정, 관리 기능
Curve	curve	NURBS 곡선을 표현하는 기능
	xcurve	곡선과 곡선의 인터섹션 기능
Surface	surface	NURBS 곡면을 표현하는 기능
	xgeom ufilegeom	곡면과 곡면(곡선)의 인터섹션 기능 곡선, 곡면의 유틸리티 클래스 정의
Topology	topolent	위상 요소 정의하는 기능
	topoling	위상 요소에 관한 정보 검색 기능
	xtopol	위상 요소 사이의 인터섹션 기능
Modeling	eulerop	오일러 작업 기능
	body	모델 조작 기능 (복사, 삭제 등)
	dbms	DBMS 기능 (저장, 복원 등)
	sweep	스위핑 작업 기능
	prim	Primitive 생성 기능
	round	라운드 작업 기능
	sheet	박판 모델 생성 기능
	wire	와이어 모델 생성 기능
	bool	불리언 작업 기능
	modify	국부 수정 작업 기능
	iges	인터페이스 용 IGES 파일 출력 기능
	transf	변환 관련 파일 (이동, 회전 등)
	undoredo	UNDO/REDO 기능
API	kiap	API 함수 지원 기능
	grkernel	그래픽 관련 기능
Inter-face	render	Rendering 기능
	system command	하드웨어와 인터페이스하는 기능 Test-bed 용 코멘드 입력 기능

은 서로 연관이 없이 독립적으로 이용될 수 있다. 그러나 라운딩 작업에서 라운딩 곡면을 생성할 때 스위핑 작업을 이용하므로 라운딩 작업과 스위핑 작업은 종속적인 계층 구조가 된다. Fig. 5에서 하위 모듈은 상위 모듈을 참조하지 않으므로, 커널을 이용하여 응용 프로그램을 개발하려면 하위 모듈부터 필요한 만큼만 사용할 수 있다. FlexDesigner을 이용해 와이어프레임 에디터를 구현하고자 하면 곡선 모듈까지 사용하여 개발할 수 있으며, 곡면 모델링 시스템을 개발할 때에는 곡면 모듈까지 이용해야 하며, 비다양체 모델링 시스템을 개발할 때에는 위상 모듈까지 또는 API 모듈까지 이용해야 한다. 이 시스템을 다른 기종에 사용할 때에는 인터페이스 모듈을 사용하고자 하는 기종에 맞게 수정하면 된다.

Fig. 5에서 기본 모듈과 곡선 모듈, 곡면 모듈, 위상 모듈에서 설계된 클래스 구조는 Fig. 6과 같다. 기본 모듈에는 Object에서 유도되는 클래스(Box, Vector, UVector, HVector, TMatrix)와 Object 클래스에서 유도

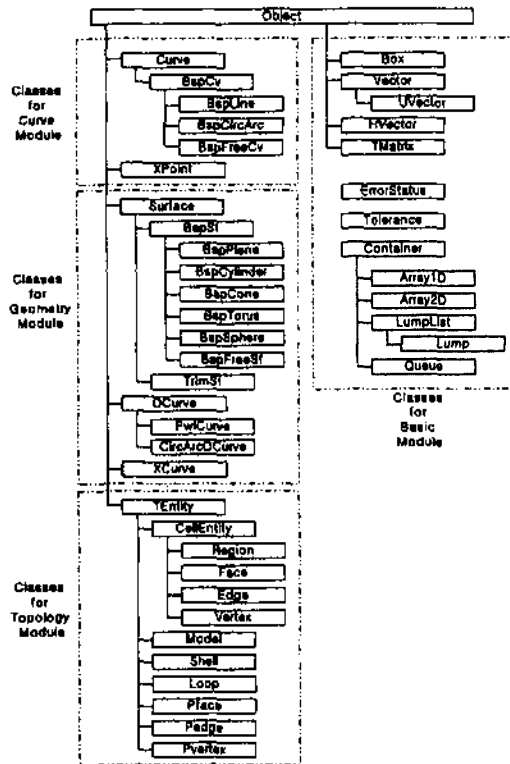


Fig. 6. Class structure of FlexDesigner.

되지 않은 클래스(ErrorStatus, Tolerance, Container와 하위 클래스)가 있다. 곡선 모듈과 곡면 모듈, 위상 모듈의 클래스는 모두 Object 클래스에서 유도된다. 곡선 모듈과 곡면 모듈에서는 곡선과 곡면을 표현하기 위해 NURBS(Non-Uniform Rational B-Splines)<sup>[15]</sup>를 사용하였다. 이는 현재 상용 모델링 시스템에서 사용하는 기하식 중 가장 유연성이 뛰어나고 2차 곡선식, 곡면식을 완벽하게 표현할 수 있기 때문이다.

### 5.1 기본 모듈

기본 모듈은 에러 처리, 공차 계산, 벡터와 행렬 계산과 같은 기본적인 계산을 위한 것으로 시스템의 가장 하위 부분을 구성한다. 기본 모듈에는 Fig. 6에서 나타난 바와 같이 Object에서 유도되는 Box, Vector, HVector, TMatrix 클래스와 Object 클래스에서 유도되지 않는 ErrorStatus, Tolerance, Container 클래스가 있다. 각 클래스에 대해서는 부록 A.1에 설명되어 있다.

### 5.2 곡선 모듈

곡선 모듈에서는 NURBS 곡선을 취급한다. 도면

작성과 같은 와이어프레임 모델링 시스템을 개발하려면, 곡선 모듈만을 이용하여 응용 프로그램의 개발이 가능하다.

곡선 모듈에는 Curve와 XPoint 클래스가 있으며, Fig. 6과 같은 계층 구조를 가진다. 현재는 NURBS로 표현되는 BspCv 만을 구현하였으나, 필요한 경우 다른 형태로 표현되는 Line(직선), CircArc(원과 원호), EllipArc(타원) 클래스 등을 확장할 수 있는 구조로 되어 있다. BspCv의 하위 클래스로 직선, 원과 원호, 자유 곡선을 표시할 수 있는 BspLine, BspCircArc, BspFreeCv가 있다. 각 클래스에 대해서는 부록 A.2에 설명되어 있다.

### 5.3 곡면 모듈

곡면 모델링 시스템의 경우는, 위상 정보는 필요 없으므로 곡면 모듈만을 이용하여 개발할 수 있다. 곡면 모듈의 구성의 곡선 모듈의 구성과 유사하다. Object 클래스를 상위 클래스로 하는 Surface 클래스가 최상위 추상 클래스이며 하위 클래스로 NURBS 곡면을 정의하는 BspSf 클래스가 있다. 현재는 NURBS로 표현되는 BspSf 만을 구현하였고, 필요한 경우 다른 형태로 표현되는 Plane(평면), Cylinder(원기둥), Cone(원뿔), Torus(토러스), Sphere(구) 클래스 등을 확장할 수 있다. 또한 BspSf의 하위 클래스로 특수한 곡면을 취급하는 BspPlane(평면), BspCylinder(원기둥), BspCone(원뿔), BspTorus(토러스), BspSphere(구), BspFreeSf(자유 곡면)가 있다. 각 클래스에 대해서는 부록 A.3에 설명되어 있다.

### 5.4 위상 모듈

위상 모듈에는 위상 요소 정의, 위상 요소 간의 관계 탐색에 대한 내용이 있다. 위상 요소에는 Fig. 3의 자료 구조에서와 같이 기본 위상 요소와 부분 위상 요소가 있다. 기본 위상 요소는 물체를 구성하는 물리적인 요소로 모델(MODEL), 영역(REGION), 셸(SHELL), 면(FACE), 루프(LOOP), 모서리(EDGE), 꼭지점(VERTEX)을 말하며, 부분 위상 요소는 기본 위상 요소 간의 관계를 나타내는 가상적인 요소로 부분면(PFACE, PARTIAL FACE), 부분 모서리(PEGE, PARTIAL EDGE), 부분 꼭지점(PVERTEX, PARTIAL VERTEX)이다. Fig. 3에 나타나지 않은 요소로 루프 순환(LOOP CYCLE), 디스크 순환(DISC CYCLE), 방사형 순환(RADIAL CYCLE)이 있다. 이는 물체를 구성하는 요소가 아니라 위상 사이의 관계를 탐색하거나 오일러 작업자를 적용할 때 사용된다. 위상

요소에 대한 자세한 설명은 이상현<sup>17)</sup>에 나타나 있다.

위상 요소 간의 관계 탐색은 자료 구조의 완전성을 나타내는 것으로, 개발 시스템에서는 이미 생성된 모델의 계층 구조적인 자료 구조로부터 주어진 위상 요소와 관련된 특정 위상 요소의 리스트를 구하기 위해 위상 정보 참조 함수를 구현하였다<sup>18)</sup>. 위상 정보 참조 함수는 다음과 같은 형식으로 구축되어 있다.

#### Inq\_“특정 위상요소”\_of\_“주어진 위상요소”(주어진 위상요소, 특정 위상 요소의 리스트)

위의 위상 정보 참조 함수는 주어진 위상요소와 관련된 혹은 이웃한 모든 특정 위상 요소를 참조하여 사용자가 참조된 결과를 리스트로 얻을 수 있게 해 주는 함수이다. 예를 들어 한 면(FACE)에 존재하는 모든 모서리(EDGE)를 찾고자 할 때 다음의 함수를 사용할 수 있다.

```
void Inq_edges_of_face (Face* given_face,
LumpList<Edge*> *inquired_edges)
```

클래스 Face와 Edge는 각각 위상 요소 FACE와 EDGE를 정의하는 클래스이며 이에 대해서는 부록 A.4에서 자세히 설명된다. 위의 식에서 주어진 면 given\_face에 존재하는 모든 모서리 inquired\_edges가 LumpList <Edge\*>로 출력된다. 검색된 모서리의 개수는 LumpList의 상위 클래스인 Container의 멤버 함수 count()로 알 수 있다.

위상 요소를 구성하는 클래스로는 Fig. 6에서와 같이 TEntity가 기초 클래스로 최상위 클래스가 되며, CellEntity, Model, Shell, Loop, Pface, Pedge, Pvertex가 TEntity의 하위 클래스가 된다. CellEntity의 하위 클래스로 Region, Face, Edge, Vertex가 있으며 이들은 불리안 작업 시 ownership과 status(DEAD/ALIVE)에 대한 정보가 필요해 이를 표현하는 CellEntity의 하위 클래스로 한다<sup>19)</sup>. 각 클래스에 대해서는 부록 A.4에 설명되어 있다.

## 6. 모델링 작업

모델링 모듈에는 오일러 작업자를 이용하는 하위 모델링 기능과 기본 형상, 스윙 작업, 불리안 작업 등을 이용하는 상위 모델링 기능이 있다. 오일러 작업은 위상 모듈에 포함시키기도 되나 모델링 가능한 작업이란 면에서 모델링 모듈에 포함시켰다.

Table 2. Euler Operators for Non-manifold Modeler<sup>[7]</sup>

	Name	Description	VEFL SCR
Basic Euler Operations	MEV (KEV)	make (kill) EDGE, VERTEX	1100000
	MEC (KEC)	make (kill) EDGE, CYCLE	0100010
	MFKC (KFMC)	make (kill) FACE, kill CYCLE	00100-10
	MFR (KFR)	make (kill) FACE, REGION	0010001
	MVS (MVS)	make (kill) VERTEX, SHELL	1000100
	MVL (KVL)	make (kill) VERTEX, LOOP	1001000
Additional Euler Operators	SEM (JEKV)	split (join) EDGE, make (kill) VERTEX	1100000
	MEF (KEF)	make (kill) EDGE, FACE	0100100
	KEML (MEKL)	kill (make) EDGE, make (kill) LOOP	0-101000
	KEMS (MEKS)	kill (make) EDGE, make (kill) SHELL	0-100100
Additional Topological Operators	MMR (KMR)	make (kill) MODEL, REGION	

### 6.1 오일러 작업

비다양체 모델링 시스템의 오일러 작업자에 대해서도 연구도 많이 수행되었다<sup>[17-19]</sup>. 오일러 작업자를 구현하기 위해 다음과 같은 오일러-포앙카레(Euler-Poincare) 식을 사용하였으며, Table 2와 같은 오일러 작업자가 시스템에서 구현되었다<sup>[9]</sup>.

$$V - E + F - L = S \cdot C + R$$

오일러 작업에서는 관례에 따라 다음과 같은 약어를 사용하였다.

M - make K - kill S - split J - join

V - vertex E - edge F - face L - hole loop (ring)

S - void shells C - cycle R - region

### 6.2 상위 모델링 작업

상위 모델링 모듈은 4장에서 설명된 기본 모듈, 곡선 모듈, 곡면 모듈, 위상 모듈까지를 사용하여 실제로 모델링 작업에 편리한 작업자가 구현되었다. 이에는 기본 형상 생성 작업, 스위핑 작업, 와이어프레임 모델 생성 작업, 박판 모델 생성 작업, 불리안 작업, 라운딩 작업, 부분 수정 작업, 모델 이동, 타 시스템과의 인터페이스 작업 등이 있다. 각각의 작업은 전술한 바와 같이 Fig. 5에서 횡 방향으로 표현되고 서로 독립적으로 구현되어 있어 필요한 경우 원하는 작업만 뽑아 낼 수 있다. 그러나 라운딩 작업은 블렌딩 곡면 생성 과정에서 스위핑 작업을 사용하므로, 스위핑 작업에 종속되어 있다.

상위 모델링 모듈은 기존 솔리드 모델링 시스템에 비해 비다양체 물체를 취급할 수 있다는 점이 다르다. 특히 스위핑 작업, 불리안 작업, 라운딩 작업, 부분 수정 작업에서는 비다양체 모델링 시스템의 특성

이 부각된다. 와이어프레임 모델, 박판 모델, 솔리드 모델이 혼합된 비다양체에서 위의 작업자가 적용될 수 있으며, 작업 후 생성된 모델도 비다양체가 된다.

FlexDesigner에서는 기존의 솔리드 모델링 시스템에서 제공하는 기본 형상인 육면체, 원기둥, 각기둥, 각뿔대, 각뿔, 구, 원뿔, 원형 튜브, 각형 튜브, 토러스 등을 기본 형상으로 제공한다<sup>[8, 12]</sup>. 비다양체에 대한 불리안 작업에 대하여 많은 연구가 있었으며<sup>[20-22]</sup> 기존의 연구들이 채택하고 있는 공통점은 병합과 선택(merge & selection)을 사용하는 것이다. FlexDesigner에서도 불리안 작업에 병합과 선택을 사용하며, 자세한 설명은 김성환<sup>[10]</sup>에 발표되어 있다. FlexDesigner의 라운딩 작업과 부분 수정 작업에 대해서는 이강수<sup>[11]</sup>에 발표되어 있다.

FlexDesigner에서는 스위핑 작업으로 솔리드 모델 뿐 아니라 비다양체 모델을 직접 생성시킬 수 있다. 점, 곡선, 곡면 또는 꼭지점, 모서리, 면을 스위핑 시킬 수 있고, 스위핑 방법으로 병진 스위핑, 경로 스위핑, 회전 스위핑, 나선 스위핑 등을 사용할 수 있다. 이를 정리하면 Table 3과 같이 된다.

특히 비다양체 모델링 시스템에서의 스위핑 작업에서는 일반 솔리드 모델링 시스템에서 부분 수정 작

Table 3. Sweeping operation

Object Entities	Sweeping method	Generated Entities
point, VERTEX	transitional sweeping	wireframe MODEL or
curve, EDGE	path sweeping	sheet MODEL or FACE
	rotational sweeping	
surface, FACE	spiral sweeping	solid MODEL



**Table 4.** Comparison of API function with ACIS v.1.6

Group	Acis	FlexDesigner
Body & DBMS	17	12
Boolean	4	8
Topology	21	39
Intersection	8	4
Primitive	7	8
Sheet & Wire	16	14
Sweep	15	12
Rounding	14	22
Modify	0	6
Interface	0	1
Undo/Redo	4	4
<b>Total</b>	<b>107</b>	<b>135</b>

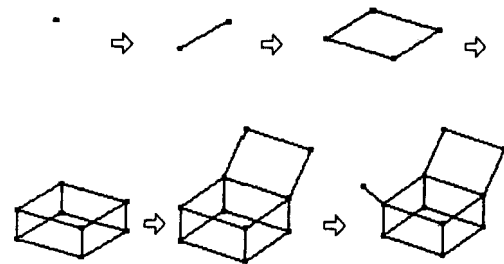
업으로 따로 제공되는 리프팅 작업을 포함한다. 즉, 모델에서 꼭지점을 스위핑 시키면 와이어 모서리가 생성되고, 모서리를 스위핑 시키면 라미나(laminar) 면, 면을 스위핑 시키면 돌출된 특징 형상이 생성된다.

**7. API 모듈**

개발 시스템과 응용 프로그램과의 인터페이스를 위해 API를 제공한다. API의 주 목적은 응용 프로그램을 개발하기 위해 안정된 인터페이스를 제공하는 것이며, FlexDesigner에서는 각 작업과 모듈에서 다양한 API 함수를 제공하고 있다. Table 4는 시스템 개발 시 목표로 하였던 ACIS<sup>TM</sup>의 API 기능과 비교한 것이다. API 함수는 상위 모델링 작업에 대한 것이 많으며, 기본 모듈과 곡선 모듈, 곡면 모듈, 위상 모듈을 사용하고자 할 때에는 클래스의 멤버 함수를 이용할 수 있다.

**8. 적용 예**

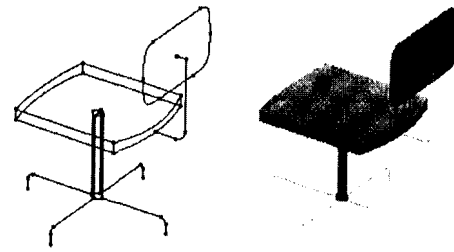
다음은 개발된 시스템으로 모델링한 몇 가지 예를 보여 준다. Fig. 7은 스위핑 작업으로만 물체를 생성하고 리프팅 작업을 적용한 예이다. 공간 상에서 점을 생성시켜 이를 위상 요소인 꼭지점으로 변환한다. 이 상태는 외꼭지점 셀로 본 연구의 자료 구조에서는 이 자체로 완벽한 비다양체 모델이 된다. 꼭지점에 스위핑을 적용시켜 와이어 모서리를 생성시킨 후, 스위핑 작업을 적용시키면 박판 모델이 되며, 이는 다시 솔리드 모델이 된다. 다섯번째 그림은 솔리드로 생성된 모델의 한 모서리에 스위핑 작업을 적용하여 dangling 면이 있는 비다양체를 생성시킨 것이고, 마지막으로 꼭지점에 스위핑 작업을 적용시키면 dangling 모서리가 있는 물체가 생성된다.



**Fig. 7.** Modeling by sweeping operation.



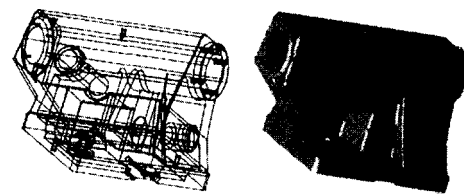
**Fig. 8.** Flange modeling.



**(a) Wireframe view (b) Shading view**

**Fig. 9.** Chair modeling.

Fig. 8은 스위핑과 불리안 작업을 이용하여 플랜지 형상을 모델링한 예이며, Fig. 9는 개념 설계 시 사용하는 축약 모델로 의자를 모델링한 것으로 앉는 부분과 주 지지대만 솔리드로 되어 있으며 나머지 연결부는 와이어, 등 받침대는 박판 모델로 되어 있어 개념 설계와 해석에 편리한 모델이다. Fig. 10은 심압대를 모델링한 것으로 기본 형상 생성 기능, 스위핑 작업, 불리안 작업을 사용해 모델링한 것이다.



**(a) Wireframe view (b) Shading view**

**Fig. 10.** Tailstock modeling.

## 9. 결 론

본 연구의 결과로 비다양체 형상의 물체를 모델링하는 CAD 시스템 커널인 FlexDesigner를 개발하였다. 이 시스템은 계층 구조의 모듈화된 구조를 하고 있으며, 모듈의 하위 구조인 기본 모듈, 곡선 모듈, 곡면 모듈, 위상 모듈은 객체 지향 설계 방식을 사용하여 개발되었으며, 상위 부분인 모델링 모듈, API 모듈은 응용 프로그램에서 편하게 사용될 수 있는 개방형 구조로 하였다. 위상 모듈에서는 새로운 비다양체 자료 구조인 이상현<sup>17)</sup>의 자료 구조를 사용하였다. 모델링 모듈에는 기존의 모델링 시스템이 제공하는 다양한 모델링 작업을 구현하였으며 이를 API 함수 형태로 제공한다. 위의 기능을 가진 비다양체 모델을 지원하는 개방형 커널 시스템을 개발함으로써 이를 이용한 응용 프로그램 개발의 활성화를 꾀할 수 있으리라 생각되며 FlexDesigner에 대해 다음과 같은 향후 연구가 필요하다.

- 데이터베이스 관리 기능 강화: 현재 사용하고 있는 데이터베이스 시스템은 현재의 자료 구조에 맞게 개발되었다. 이를 향후 데이터 교환이나 시스템의 기능 강화에 적합하도록 분산 시스템을 지원할 수 있는 객체 지향 데이터베이스 시스템(distributed object-oriented database system)으로 확장 시킨다.

- GUI, 구속 조건 관리, 특징 형상 지원, 조립체 모델링 기능: 현재 FlexDesigner는 커널로서 개발되었기 때문에 사용자 인터페이스가 불편한 코멘드 입력 방식으로 개발되어 있어, 이를 사용하기에 편한 GUI 방식으로 대체할 필요가 있다. 그리고 순수한 커널 부분은 완성되었으나, 모델링 시 구속 조건을 부여하는 방법, 특징 형상을 지원하는 기능, 조립체 모델링을 할 수 있는 기능 등이 추가되면 그 유용성이 더욱 크리라 생각된다.

- 비다양체 모델링 시스템 구현: FlexDesigner는 비다양체 모델링 커널이므로 이를 이용하여 비다양체 모델링 시스템을 만드는 연구를 진행 중이다. FlexDesigner는 SGI 워크스테이션과 IBM PC에 사용이 가능하지만, PC가 범용성과 이식성, 경제성이 뛰어나므로 현재 PC 환경하에서 개발 중이다. 또한 PC에서는 MFC가 훌륭한 GUI를 제공해 줄 수 있다. 여기에는 GUI, 구속 조건 관리 기능(constraints management), 특징 형상 관련 기능 등에 대한 연구가 진행되어야 한다.

## 감사의 글

이 연구는 통상산업부 G7 프로젝트인 "설계 자동화 기술: CAD 시스템 Kernel 개발"로 (주)큐빅테크의 주관 아래 진행되었다.

## 참고문헌

1. Sriram, R.D., Wong, A. and He, L.X., "GNOMES: an object-oriented nonmanifold geometric engine", *Computer-Aided Design*, Vol. 27, No. 11, pp. 853-868, Nov. 1995.
2. 이상현, 이진우, "비다양체 모델링 시스템을 이용한 사출 성형용 CAD 및 CAE 시스템의 통합", '96 한국 CAD/CAM학회 학술 발표회 논문집, pp. 125-130, March 1996.
3. Weiler, K., "Topological Structures for Geometric Modeling", *PhD. Thesis*, Rensselaer Polytechnic Institute, Aug. 1986.
4. Choi, Y., "Vertex-based Boundary Representation of Non-manifold Geometric Models", *PhD. Thesis*, Carnegie Mellon Univ., Aug. 1989.
5. Yamaguchi, Y., Kobayashi, K. and Kimura, F., "Geometric Modeling with Generalized Topology and Geometry for Product Engineering", in J. Turner, J. Pegna and M. Wozny (Eds.) *Product Modeling for Computer-Aided Design and Manufacturing*, North-Holland, pp. 97-115, 1991.
6. Rossignac, J. and O'Connor, M., "Selective geometry complex: a dimension independent model for pointsets with internal structure and incomplete boundaries", in Wozny, M.J., Turner, J.U. and Preiss, K (Eds.) *Geometric Modeling for Product Engineering*, North-Holland, pp. 145-180, 1990.
7. 이상현, 이진우, "비다양체 위상 표현을 바탕으로 한 박판 형상 모델링 및 솔리드로의 변환", 한국정밀공학회지, 제 13권, 제 7호, pp. 100-114, June 1996.
8. 이진우 외, "설계 자동화 기술: CAD 시스템 Kernel 개발", 서울대학교 정밀기계설계 공동연구소, Nov. 1996.
9. 한영현, 이진우, 이상현, 김성환, 김영진, 배석훈, 안재홍, 이경진, "비다양체를 지원하는 CAD 시스템 커널 개발을 위한 기반 구축", 산업 공학, 제 8권, 제 3호, pp. 141-153, 1995.
10. Han, Y., Lee, K., Kim, Y., Lee, K.S., Hong, J., Ahn, J., Hur, J., Lee, K., Lee, S. and Kim, S., "A Framework in Developing Kernel Modeler Based on Nonmanifold Boundary Representation", *Pacific Conference on Manufacturing '96*, Vol. 11, pp. 195-200, Oct. 1996.
11. 이강수, 이진우, 허정훈, 박성준, 김성환, "비다양체 모델링 시스템의 국부 모델링 작업 개발", *Proceedings of the 4th Advanced Manufacturing System Workshop*, pp. 32-45, Sep. 1996.

12. 이강수, 이진우, 한영현, 홍진웅, 박상근, 허정훈, 안재홍, 이경진, 박성준, 김성환, 이상현, 김영진, 정진평, "비다양체 모델을 지원하는 개방형 커널 시스템의 개발", '97 한국 CAD/CAM학회 학술 발표회 논문집, pp. 26-33, Feb. 1997.
13. Paulo, R.C., Paulo, C.P.C. and Luiz, F.M., "Non-manifold modelling: an approach based on spatial subdivision". *Computer-Aided Design*, Vol. 29, No. 3, pp. 209-220, 1997.
14. 김형주, "알기 쉬운 객체 지향 시스템", 동아출판사, 1993.
15. 이상찬, "자동차 외형 설계를 위한 CAD 시스템의 개발", 박사학위논문, 서울대학교, Aug. 1992.
16. 김성환, 이진우, 김영진, "비다양체 모델을 수용하는 CAD 시스템 커널을 위한 볼리안 조작의 개발", 한국 CAD/CAM학회 논문집, 제 1권, 제 1호, pp. 20-32, March 1996.
17. Yamaguchi, Y. and Kimura, F., "Nonmanifold Topology Based on Coupling Entities", *IEEE Computer Graphics and Applications*, Vol. 15, No. 1, pp. 42-50, Jan. 1995.
18. Luo, Y. and Gabor, L., "Generalized Euler Operators for Non-manifold Boundary Solid Modeling", *Geometric Modelling Studies 1990/3*, MTA SZTAKI, Hungary, pp. 19-34, 1993.
19. 이상현, 이진우, "비다양체 형상 모델링을 위한 간결한 경계 표현 및 확장된 오일러 작업자", 한국 CAD/CAM학회 논문집, 제 1권, 제 1호, pp. 1-19, March 1996.
20. Gursoz, E.L., Choi, Y. and Prinz, F.B., "Boolean set operations on non-manifold boundary representation objects", *Computer-Aided Design*, Vol. 23, No. 1, 1991.
21. Crocker, G.A. and Reinke, W.F., "An Editable Non-manifold Boundary Representation", *IEEE Computer Graphics and Applications*, pp. 39-51, March 1991.
22. Masuda, H., "Topological Operators and Boolean Operations for Complex-based Non-manifold Geometric Models", *Computer-Aided Design*, Vol. 25, No. 2, pp. 119-129, Feb. 1993.
23. Spatial Technology Inc., "ASIC Geometric Modeler, API Reference", Spatial Technology Inc., USA, 1994.



### 이 강 수

1987년 서울대학교 기계공학과 학사  
 1989년 서울대학교 기계공학과 석사  
 1989년 ~ 현재 대우중공업(주) 과장  
 1995년 ~ 현재 서울대학교 기계설계학과 박사과정  
 관심분야: Solid modeling, Power train design, Intelligent CAD, Design methodology



### 이 진 우

1978년 서울대학교 기계공학과 학사  
 1981년 미국 MIT 공학석사  
 1984년 미국 MIT 공학박사  
 1984년 ~ 1986년 Univ. of Illinois at Urbana-Champaign 조교수  
 1986년 ~ 현재 서울대학교 기계설계학과 교수  
 관심분야: Computer aided geometric design, NC tool path generation and verification, Solid modeling, Rapid prototyping

## 부록: 객체 지향 방식으로 설계된 클래스

### A.1 기본 모듈

기본 모듈에는 Object에서 유도되는 Box, Vector, HVector, TMatrix 클래스와 Object 클래스에서 유도되지 않는 ErrorStatus, Tolerance, Container 클래스가 있다.

**Object:** 이 클래스는 최상위 추상 클래스로서 실제 응용 프로그램에서는 사용되지 않지만 다른 객체가 생성될 때 함께 생성되는 가장 기본이 되는 클래스이다. 하위 클래스로서 Box, Vector, HVector, TMatrix, Curve, XPoint, Surface, DCurve, XCurve, TrimSf, TEntity 등의 클래스가 있다.

**메소드:** 출력 기능이 있어 클래스의 내용을 출력하여 디버깅하는 데 유용하다.

**Box:** 점 또는 벡터 집합을 입력 받아 주어진 집합을 포함하는 최소 크기의 직육면체(box)를 만들기 위한 클래스이다. 인자로 직육면체(box)를 이루는 대각선 방향의 두 점(최소점, 최대점)을 저장한다.

**메소드:** 주어진 점을 수용하는 최소 크기의 Box를 만들어, 점 집합 사이의 빠른 교차 검사, 임의의 한 점이 점 집합(Box)에 속해 있는 지에 대한 빠른 검사 등이 가능하다.

**Vector:** 일반적인 3차원 벡터를 생성하여 처리하는 클래스이다. Point도 벡터로 정의 되어 있기 때문에 3차원 상의 임의의 점의 표현도 이 Vector 클래스를 통하여 가능하다. 하위 클래스로 UVector 클래스가 있으며 벡터를 구성하는 x, y, z 값을 인자로 가진다.

**메소드:** 일반적인 벡터 연산(+, -, \*, /, =)에 대한 함수가 구현되어 있다.

**UVector:** Vector 클래스의 하위 클래스로, 인자를 정규화 시켜 단위 벡터가 되도록 한다.

**HVector:** Homogeneous Vector를 표현하는 클래스이다. Homogeneous Vector란 벡터 변환을 연속적인 행렬의 곱의 형태로 표현하기 위하여 x, y, z 좌표값 이외에 가중치인 w값을 첨가하여 좌표계에서의 특정 위치를 표시하는 것으로, non-rational B-Splines를 취급할 때 유용하게 이용된다. 일반적인 벡터를 이루는 원소인 x좌표, y좌표, z좌표 이외에 또 다른 요소인 w 값이 존재한다는 것이 벡터 클래스와 다르다. 상위 클래스로 Object 클래스를 가지고 있다.

**ErrorStatus:** ErrorStatus는 발생한 에러의 정도, 에러가 발생한 파일 이름과 함수 이름, 라인 번호 등을 기억하는 인자가 있다. 에러의 심각한 정도를 나타내는 값으로는 SYS\_WARNING, SYS\_ERROR, SYS\_PANIC 이 있다. SYS\_WARNING은 가장 정도가 가볍고 단순히 경고의 차원일 때, SYS\_PANIC은 시스템이 하드웨어 혹은 소프트웨어적으로 치명적인 피해를 입어 더 이상 작업 수행이 곤란할 때, SYS\_ERROR는 그 중간 정도로 보통의 경우 에러는 여기에 해당한다고 볼 수 있다. 나머지 변수들은 에러의 종류, 발생 위치를 저장한다.

**메소드:** 에러 처리 기능이란 예상하지 못한 내부 에러 상황으로부터 커널 시스템을 보호하는 기능, 혹은 예상 가능한 사용자의 실수로 커널 시스템이 정지하지 않도록 하는 기능을 말한다. 에러 처리 모듈은 이러한 기능을 구현할 수 있도록, 에러가 발생했을 때 그 에러의 내용을 저장할 수 있는 방법과 프로그래머가 커널 시스템의 에러 상태를 원할 때마다 검사할 수 있는 방법, 그리고 최종적으로 한 명령어를 수행한 결과 발생하는 에러들을 프린트할 수 있는 방법 등을 제공한다. 에러 처리 기능은 커널 시스템의 모든 모듈에 적용할 수 있는 가장 기초적인 기능이며 프로그래머는 이 기능을 잘 이용함으로써 프로그램의 작동상태를 디버깅할 수 있는 좋은 장치로 활용할 수도 있다.

**Tolerance:** 커널 시스템에서 발생하는 공차 문제는 부동 소수점 연산 시 발생하는 오차(round-off error)와 연산 오차(operational error)가 있다. 이 중 시스템의 안정성과 정확한 작동을 위해서는 연산 오차를 처리하는 기능이 좋아야 한다. 사실상 컴퓨터의 연산 정밀도는 우리가 관심을 갖는 기하학적 연산 문제에 대해 생각할 때 충분히 정밀하다고 할 수 있다. 따라서 이로 인한 오차는 연산 허용 오차에 비하여 훨씬 작기 때문에, 실제 오차 문제는 연산 허용 오차의 일관성 여부에 좌우된다고 할 수 있다. 특히 연산 허용 오차는 기하 공간(Euclidean space)과 위상 공간(topology space) 간의 사상 연산(mapping operation)에 중요한 척도로 사용된다. 여기서 사상 연산이란, 수치 데이터(numerical data)로 표현되는 기하 상태를 논리적으로 확정적인 위상 상태로 변환하는 연산을 의미한다. 즉, 점의 좌표값, 매개 변수값 등과 같은 기하 공간 상의 부동 소수 데이터를 SAME, IN, OUT, ON, LEFT, RIGHT, UPPER, LOWER 등과 같은 논리 데이터로 변환할 때의 변환 기준이 바로 '연산 허용 오차'라고 할 수 있다.

개발된 시스템에서는 연산 오차를 처리하는데 주력하였으며 일관된 허용 오차의 적용, 동적 허용 오차 적용, 기하 공간과 매개 변수 공간의 분리라는 원칙을 적용하였다<sup>[8]</sup>. 동적 허용 오차란 허용 오차 기준을 일정한 값으로 설정하는 것이 아니라 모델링 공간의 크기에 따라 변화 시키는 방법이다. 인자로 주어진 숫자를 "0"으로 취급하는 기준값, 두 점을 갈게 취급하는 기준값, 파라미터 도메인에서 주어진 두 파라미터를 갈게 인식하는 기준값 등이 정의되어 있다.

**메소드:** 주어진 수가 "0"인지 결정하고, 두 숫자 사이의 논리적인 판단(==, >, <, <=, >=)을 한다. 파라미터 도메인에서도 같은 기능을 한다.

**Container:** 객체를 배열 데이터로 효율적으로 저장하고 관리하기 위한 모듈 중 가장 상위 클래스이며, 하위 클래스로서 Array1D, Array2D, LumpList, Queue 클래스가 있다. 이 자료 저장 객체로 다른 여러 종류의 객체들을 하나의 집합체로 효율적으로 묶을 수 있다. 이 부분은 대형 프로그램에서 가장 기본이 되면서도 가장 중요한 부분이 된다.

**메소드:** 저장된 배열의 개수를 알아 내는 기능과 저장된 데이터를 출력하는 기능이 있다.

**Array1D:** 특정 객체를 1차원 배열로서 저장하고자 할 때 사용하는 클래스이다. 상위 클래스로 Container 클래스를 가지고 있으며, 모든 배열 객체는 template로 구현되어 있어 어떠한 형태의 객체도 하나의 집합으로 구성할 수 있다. Template는 클래스를 int, double과 같이 내장된 자료형(built-in data type)과 같이 취급할 수 있게 해준다.

**메소드:** 내장된 자료형을 1차원 배열로 사용할 때와 사용법이 유사하며, 객체를 배열로 저장하기 위해 기억 용량을 할당하고 해제하는 기능이 있다. 또한 저장된 데이터를 검색하는 기능을 한다.

**Array2D:** 특정 객체를 2차원 배열로서 저장하고자 할 때 사용하는 클래스이며 Array1D와 유사한 객체 지원 특성이 있다. 사용 방법은 내장된 자료형을 2 차원 배열로 사용할 때와 유사하다.

**LumpList:** LumpList 클래스는 자료 리스트의 특수한 형태로, 방대한 양의 데이터를 저장할 때에 빠른 참조를 가능하게 하기 위하여 일정한 양의 데이터들의 묶음이 Lump 단위로 여러 개 존재하며, 각 Lump는 포인터로 링크되어 있다. 역시 객체는 template로 구현되어 있어 어떠한 형태의 객체(또는 데이터)도 하나의 리스트로 구성할 수 있다. 이 관계가 Fig. A1에 나타나 있다.

**메소드:** 데이터를 첨가, 삭제, 검색하는 기능이 있으며, 두 개의 LumpList에 있는 데이터를 비교하여 합하거나 빼며, 공통된 데이터를 추출하는 기능 등이 있다.

**Lump:** LumpList를 구성하는 기본 단위인 Lump를 구현하는 클래스다. 이 곳에서 Lump의 크기가 일정한 값으로 정의 될 수 있으며 이러한, 특정한 크기를 가진 Lump들이 모여서 하나의 리스트를 이루게 되면 그것이 LumpList이다. 각 Lump에는 임의의 객체들이 저장될 수 있으며, 뒤에 연결될 또 다른 Lump를 포인터로 가리키게 된다. 이 관계가 Fig. A1에 나타나 있다.

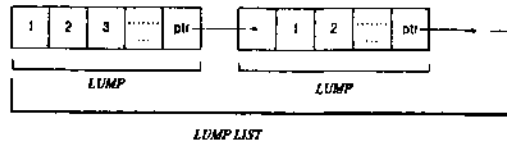


Fig. A1. Structure of Lump and LumpList.

**Queue:** Queue 클래스는 기본적으로는 배열 구조와 흡사한 데이터 저장 양식이지만, FIFO (First-In First-Out) 법칙을 따른다는 점에서 다르다. Queue에 데이터를 저장하고 검색, 인출하는 기능이 있다.

## A.2 곡선 모듈

곡선 모듈에는 Curve와 XPoint 클래스가 있으며, Curve의 하위 클래스로 BspCv, BspCv의 하위 클래스로 BspLine, BspCircArc, BspFreeCv가 있다.

**Curve:** 곡선 모듈의 기초 클래스로 NURBS 곡선을 정의하는 BspCv 클래스를 하위 클래스로 두고 있다.

**메소드:** 추상 클래스로 다음과 같은 곡선에 대한 기본 작업들이 정의되어 있다.

- 곡선의 복사
- 곡선의 정보 질의 (곡선 형태, 차수, 파라미터 관련 정보 등)
- 곡선 변환 (이동, 축소/확대 등 변환 행렬 이용)

**BspCv:** NURBS 곡선에 대하여 정의되어 있으며 Curve 클래스에서 유도되고 하위 클래스로는 BspLine(직선), BspCircArc(원과 원호), BspFreeCv(자유 곡선)가 있다. 인자로 NURBS 곡선의 차수, 조정점 개수, 절점 벡터, 조정점이 있다.

**메소드:** NURBS에 대한 기본 작업들이 구현되어 있다.

- 데이터 설정/정보 검색 기능: 인자의 값(차수, 조정점 개수, 절점 벡터, 조정점)을 설정하고 또 저장된 값을 알아낸다.
- 곡선 변형 기능: 차수 증가, 절점 추가, 회전/이동 등 기능
- 계산 기능: 주어진 파라미터에 대한 위치, 미분값, 곡률 등을 계산하는 기능

**XPoint:** 두 곡선의 교차점을 취급하기 위한 것으로 교차점과 교차점에서의 매개 변수값을 저장한다.

**메소드:** 교차점과 교차점에서의 매개 변수를 설정하고 검색하는 기능을 한다.

## A.3 곡면 모듈

곡면 모듈에는 Surface와 DCurve, XCurve 클래스가 있으며, Surface의 하위 클래스로 BspSf와 TrimSf가 있다.

또한 BspSf의 하위 클래스로 BspPlane(평면), BspCylinder(원기둥), BspCone(원뿔), BspTorus(토러스), BspSphere(구), BspFreeSf(자유 곡면)가 있다.

**Surface:** 곡면 모듈의 기초 클래스로 NURBS 곡면을 정의하는 BspSf와 트림 곡면을 정의하는 TrimSf를 하위 클래스로 두고 있다.

**메소드:** 추상 클래스로 다음과 같은 곡면에 대한 기본 작업들이 정의되어 있다.

- 곡면의 복사
- 곡면의 정보 질의 (곡선 형태, 차수, 파라미터 관련 정보 등)
- 곡면 변환 (이동, 축소/확대 등 변환 행렬 이용)

**BspSf:** NURBS 곡면을 정의하며, 상위 클래스로는 Surface가 있고 하위 클래스로는 BspSf의 하위 클래스로 BspPlane(평면), BspCylinder(원기둥), BspCone(원뿔), BspTorus(토러스), BspSphere(구), BspFreeSf(자유 곡면)가 있다. 인자로 NURBS 곡면의 u, v 방향의 차수, 조정점 개수, 절점 벡터, 조정점을 저장하고 있다.

**메소드:** NURBS에 대한 기본 작업들이 구현되어 있다.

- 데이터 설정/정보 검색 기능: 인자의 값(차수, 조정점 개수, 절점 벡터, 조정점)을 설정하고 또 저장된 값을 알아낸다.
- 곡면 변형 기능: 차수 증가, 절점 추가, 회전/이동 등 기능
- 계산 기능: 주어진 파라미터에 대한 위치, 미분값, 곡률 등을 계산하는 기능

**TrimSf:** 트림 곡면에 대해 정의하는 클래스로 상위 클래스는 Surface이다. 여기에는 베이스 곡면과 경계 곡선에 대한 정보를 저장하고 있다.

**DCurve:** 매개 변수 공간 상에서 존재하는 곡선에 대해 정의하는 클래스로 상위 클래스는 Object이다. NURBS 곡면과 같이 매개 변수 (u, v)로 정의되는 곡면 상에 존재하는 3차원 곡선을 2차원인 매개 변수 공간으로 매핑한 것이 매개 변수 곡선이다. 이 클래스는 특히 불리안 작업에서 두 곡면의 교차 곡선을 구할 때 사용된다.

**XCurve:** 두 곡면 간의 교차 곡선에 대해 정의하는 클래스로 상위 클래스는 Object이다. 교차점을 먼저 계산하여 교차점으로부터 교차 곡선을 생성한다. 이 때 생성된 교차 곡선을 두 곡면에 매핑시킨 매개 변수 곡선에 대한 정보도 저장하고 있다.

#### A.4 위상 모듈

위상 요소를 구성하는 클래스로는 TEntity가 기초 클래스로 최상위 클래스가 되며, CellEntity, Model, Shell, Loop, Pface, Pedge, Pvertex가 TEntity의 하위 클래스가 된다. CellEntity의 하위 클래스로 Region, Face, Edge, Vertex가 있다.

**TEntity:** 위상 요소를 정의하기 위한 기초 클래스이다. 위상 요소의 id에 해당하는 인자가 있다. 이 인자는 모든 위상 요소의 공통 사항이므로 위상 요소를 정의하는 클래스 중 가장 상위 요소인 TEntity에 정의한다.

**메소드:** 위상 요소의 id를 설정하는 기능과 추상 함수로 위상 요소의 이름, 타입, 위상 요소가 속한 MODEL을 검색할 수 있다.

**CellEntity:** 불리안 작업의 병합 집합(merged set)에서 ownership과 status(DEAD/ALIVE)를 갖는 위상 요소 Region, Face, Edge, Vertex의 상위 클래스이다. 불리안 작업에 대해서는 김성환<sup>116)</sup>에 자세하게 나와 있다. 불리안 작업의 병합 집합에서 필요한 특징인 ownership과 status를 인자로 갖는다.

**메소드:** ownership과 status의 값을 설정하고 검색하는 기능을 한다.

Fig. 6에서와 같이 TEntity와 CellEntity에서 파생되는 클래스로 Region, Face, Edge, Vertex, Model, Shell, Loop, Pface, Pedge, Pvertex가 있다. 각 클래스는 Fig. 3의 구성도에 맞게 멤버 인자와 멤버 함수가 구축되어 있다. 임의로 클래스를 예로 들면 다음과 같다.

**Face:** Fig. 3의 자료 구조를 만족시키기 위해 상위 PFACE를 가리키는 인자(\_parent)와 다음 면을 가리키는 인자(\_next), 그리고 하위 LOOP를 가리키는 인자(\_loop)가 정의되어 있다. 그리고 면의 곡면식을 가리키는 인자(\_geometry)가 있다.

**메소드:** 위상 요소인 면의 위상 데이터와 곡면 데이터를 설정하고 검색하는 기능이 구현되어 있다. 즉,

**Table A1.** Typical member variables and functions of Class "Face"

	Member variables/functions	Note
Member variables	Face* _next	Pointer to the next FACE
	TEntity* _parent	Pointer to the parent entity
	Loop* _loop	Pointer to the LOOP of this FACE
	Surface* _geometry	Pointer to surface of this FACE
Member functions	void set_next(Face* face)	Set the next FACE
	Face* next()	Get the next FACE
	void set_parent(TEntity* parent)	Set the parent entity
	TEntity* parent()	Get the parent entity
	void set_loop(Loop* loop)	Set the first loop of this FACE
	Loop* loop()	Get the first loop of this FACE
	void set_geometry(Surface & sf)	Set the surface of this FACE
	Surface* geometry()	Get the surface of this FACE
	PFace* pface()	Get the parent PFACE if the parent entity is PFACE
	PFace* front_pface()	Get the front PFACE having the same normal direction as this FACE
	PFace* back_pface()	Get the front PFACE having the opposite normal direction against this FACE
	int num_loops()	Get the number of loops
	int num_hole_loops()	Get the number of hole loops
	int classify(const Point & point)	Classify the position of the given point(on/in/out)
	Box bound_box()	Get the Box bounding this FACE
	Point point_in_face()	Get a point on this FACE
	Point center()	Get the center point of this FACE
UVector normal()	Get the normal at the center point	
Bool is_laminar_face()	Inquire whether this FACE is laminar	

Note:

1 "Point" is defined as the same as "Vector", or, "Point" is the alias of "Vector"

2 "Bool" is the logical value which represents TRUE or FALSE.

\_parent, \_next, \_loop, \_geometry를 설정하고 관련된 데이터를 검색하는 기본적인 기능과, 관련 위상 요소에 대한 검색을 위한 함수가 구현되어 있다. 구체적인 함수 구현은 Table. A1에 나타나 있다.

위상 요소를 정의하는 다른 클래스도 Face와 같이 자료 구조를 만족시키며 관련된 데이터를 설정하고 검색할 수 있도록 구현되어 있다.