

CAD/CAM 응용 소프트웨어 개발을 위한 형상 커널 개발

정연찬*, 박준철**

Geometric Kernel for CAD/CAM Application Software Development

Chung, Y. C.* and Park, J. C.**

ABSTRACT

A geometric kernel is the library of core mathematical functions that defines and stores 3D shapes in response to users' commands. We developed a light geometric kernel suitable to develop CAD/CAM application systems. The kernel contains geometric objects, such as points, curves and surfaces and a minimal set of functions for each type but does not contain lots of modeling and handling functions that are useful to create and maintain complex shapes from an idea sketch. The kernel was developed on MS-Windows NT using C++ with STL(Standard Template Library) but it is compatible with UNIX environments. This paper describes the structure of the kernel including several components: base, math, point sequence curve, geometry, translators. The base kernel gives portability to applications and the math kernel contains basic arithmetic and their classes, such as vector and matrix. The geometry kernel contains points, parametric curves, and parametric surfaces. A neutral file format and programming and document styles are also presented in this paper.

Key words : Geometric kernel, Object class, Program styling, Documentation

1. 서 론

소프트웨어 시스템에서 커널(kernel)은 자동차의 엔진과 같이 시스템 내부의 가장 핵심적인 요소다. 기하형상(geometric shape)을 다루는 CAD/CAM 소프트웨어 시스템에서는 형상 모델링 커널(geometric modeling kernel)이 핵심 커널에 해당한다. 형상 모델링 커널은 기하형상을 정의하고 표현하거나 처리할 때 발생하는 수학적 연산을 담당한다. 널리 알려진 형상 모델링 커널로는 ACIS^[1], Parasolid^[2], Design Base^[3] 등과 같이 상용화된 것이 있으며, Open CASCADE^[4]와 같이 소스가 공개된 것도 있다.

이들 상용화된 형상 모델링 커널을 이용해서 응용 제품을 개발하는 데에는 몇 가지 어려움이 있다. 우선, 커널의 규모가 크고 복잡하다. 이러한 커널들은 특정한 응용 분야에 특화되어 개발된 것이 아니라 다양

한 요구가 하나의 일관된 틀 속에 구현되었기 때문이다. 따라서 사용법이 복잡하거나 용도에 적합하지 않은 경우가 많으며, 특정한 분야의 응용 시스템을 개발할 경우 모든 개발자들이 커널을 제대로 이해하는 것은 상당한 부담으로 작용한다. 그리고 프로그램의 규모가 크기 때문에 실행 파일의 크기가 큰 것이 일반적인데, 이는 응용제품의 크기를 키우므로 전체적인 수행속도를 저하시키는 요인이 된다.

또한, 커널에서 제공되지 않는 새로운 기능은 필요한 시점에 곧바로 개발되지 않는 것이 일반적이며, 어떤 경우에는 가능 구현 요청에도 불구하고 개발되지 않을 수도 있다. 커널에 버그가 있을 수도 있는데, 이에 대한 재빠른 대응을 기대하기 힘든 것 또한 사실이다.

상용화된 커널을 응용 소프트웨어 시스템 개발에 이용하는 경우에도 커널에서 제공되지 않는 형상 요소가 있을 수 있으며, 데이터의 변환을 위해 임시적으로 형상 정보를 저장할 그릇(데이터 구조)이 필요한 경우가 많다. 커널의 일반적인 기능을 활용하고, 커널에 없는 특화된 기능을 개발하는 경우에도 데이터를 담고 처리할 독자적인 데이터 구조를 별개로 유지할 필요성이

*중신회원, 큐빅테크 큐빅기술연구소
**정회원, 큐빅테크 큐빅기술연구소
- 논문투고일: 2001. 05. 14
- 심사완료일: 2001. 09. 03

있다. 특히, 일반적인 상용 커널은 가공 데이터 생성에 관한 특별한 기능을 갖고 있지 않다. 따라서 가공 기능을 개발하는 경우 별도의 데이터 저장 구조와 처리 기능을 개발해야 한다.

상용화된 커널을 사용하는 경우에 데이터 처리 및 저장 구조의 구현 방법은 몇 가지 방안이 있다. 첫 번째는 커널에서 객체 클래스(object class)를 상속 받는 방법이다. 그러나 이 방법은 클래스 구조의 중간에 위치한 추상화된 개념적 객체(abstract class)를 상속해서 확장할 수 없기 때문에 최하위 클래스를 상속 받게 된다. 그런데, 최하위 클래스는 대부분 종류가 많기 때문에 프로그램의 양이 많아지고 유지보수가 어려워지게 된다. 두 번째 방법은 커널의 객체를 구성 요소로 갖는 별도의 객체를 정의하는 방법인데, 이 방법은 새로 정의된 객체를 관리할 별도의 데이터 구조가 필요하다. 따라서 생성된 객체들을 커널과 동일하게 유지해야 하는 부담이 존재한다. 세 번째 방법은 커널의 객체 관리 기능을 사용하지 않고 커널의 연산 기능만 활용하는 방법이다. 별도의 객체 클래스와 데이터 관리 구조를 마련하고 커널의 연산이 필요한 경우에 커널에 데이터를 변환해서 전달하는 방법이다. 이 방법은 커널의 의존성이 가장 적은 방법이며, 커널이 담당하는 연산을 모두 대체할 수 있다면 상용 커널을 제거할 수도 있다. 또, 커널을 다른 커널로 쉽게 대체할 수도 있다. 그러나, 본 연구에서 개발한 형상 커널과 같이 형상 정보를 저장하고 처리할 수 있는 별도의 구조가 필요하다.

본 연구에서는 개발한 “형상 커널”은 복잡한 형상 모델링 기능을 지원하지 않는다. 형상을 저장할 그릇(객체 클래스)과 객체의 간단한 연산자만을 제공하는 가벼운 커널이 개발의 목표다. 몇몇 CAD/CAM 응용 소프트웨어의 경우에 이러한 가벼운 형상 커널만으로도 개발이 가능하였으며, 상용 커널을 써야 하는 경우에도 개발을 용이하게 하고 제품의 유지보수 능력을 향상할 수 있었으며, 상용 커널의 의존성을 줄일 수 있었다. 본 논문을 통해서 커널 개발의 경험을 공유하고자 한다.

2장에서는 개발한 형상 커널의 구조 및 내용을 기술하고, 3장에서는 커널 데이터를 저장할 중립파일^[5]의 형식을 제안한다. 4장에서는 커널 개발에 적용한 프로그래밍 표준 안에 대해 설명을 한다. 적용 사례 및 결론은 5장에서 기술한다.

2. 형상 커널의 구조

2.1 형상 커널의 전체적인 구조

형상 커널은 Fig. 1에서 보듯이 계층적으로 구성되

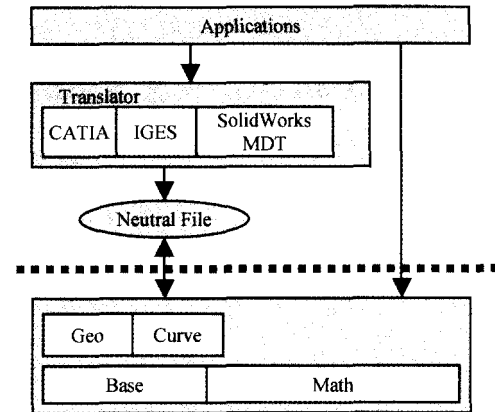


Fig. 1. Structure of geometric kernel.

어 있다. 가장 아래쪽이 기초적인 계층이고 응용 계층이 최상위에 존재한다. 상위 계층에서는 아래쪽 계층의 연산자를 사용할 수 있으며, 아래쪽 계층의 객체를 상속해서 정의할 수도 있다. 그러나 그 역은 불가능하다. 응용 프로그램(application program)은 모든 계층의 요소 커널(component kernel)을 호출할 수 있다. 데이터 변환기(translator)는 응용 프로그램의 한 종류로 볼 수 있다.

2.2 요소 커널의 내용

본 절에서는 형상 커널을 구성하는 요소 커널의 구현 내용과 특이 사항을 간단히 설명한다.

2.2.1 기초 커널

기초 커널은 형상 커널의 가장 기초적인 요소 커널이며, 모든 다른 요소 커널에서 사용된다. 기초 커널은 다음과 같은 내용을 포함하고 있다.

- 컴파일 환경에 따른 환경 설정문
- 상수(constant)
- 변수 타입(typedef)
- 기초 객체 클래스
- 컨테이너(container) 클래스
- 기초적인 인라인 함수(inline function)

커널은 윈도, 유닉스 등의 다양한 환경에서 컴파일되어야 하기 때문에 컴파일 환경에 따라 달라지는 부분을 고려하고 있다. 또, 상수 및 변수 타입도 컴파일 환경에 따라 상이할 수 있기 때문에 별도로 지정해서 사용하고 있다.

기초 객체 클래스는 특정한 타입의 객체에 대한 공통 요소만 모아서 그 골격을 정의하는 추상(abstract)

클래스이다. 대표적인 예로는 화면에 그려지는 객체, 이름이 있는 객체, 파일 입출력이 가능한 객체 등등이다. 이러한 추상 객체들은 상속 객체들의 일관성 있는 구현을 강제하게 된다.

컨테이너 클래스는 리스트나 배열과 같이 데이터를 저장할 수 있는 그릇이다. 컨테이너 클래스는 STL (Standard Template Library)⁽⁶⁾을 상속하는 것을 기본 원칙으로 한다. STL은 비록 C++의 표준으로 채택되었지만 아직 컴파일러마다 상이한 점이 많으며, 실행 파일의 크기가 커진다는 단점이 있다. 특히, 대부분의 디버깅 도구에서 STL 반복자(iterator)의 내용을 볼 수 없다는 것은 치명적인 단점으로 지적된다. 그러나 STL에서 지원하는 다양한 클래스와 알고리즘을 모두 새롭게 구현하기 위해서는 많은 작업을 필요로 하며, 구현된 내용의 문서화와 유지 보수가 어렵다. 본 개발팀에서도 많은 이견이 있었지만 표준화 원칙에 가장 많은 비중을 두고 STL을 채택하게 되었다.

2.2.2 수학 커널

수학 커널은 기초적인 수학 연산을 담당하며 다음과 같은 내용으로 구성되어 있다.

- 벡터(vector/position)
- 행렬(matrix)
- 기초 도형(basic geometric entities)
- 인라인 함수(inline function)

벡터 클래스는 2, 3, 4차원 벡터를 다루고 있으며, 4차원 벡터는 NURBS 곡선 혹은 곡면 계산을 위한 연산자만 지원한다. 그런데 이들 2, 3, 4차원 벡터는 서로 상속(inheritance) 관계없이 구현되었다. 상속은 통상 가상(virtual) 함수를 사용하기 위해서인데, 가상 함수를 쓰는 경우 생성되는 모든 객체(instance)에 가상 함수 포인터를 저장하는 메모리가 추가로 할당되기 때문이다. 비록 4바이트 혹은 8바이트의 작은 양이지만, 이들 벡터는 아주 많은 개수의 객체가 생성될 것이기 때문에 메모리 부담이 커지게 된다. 또, 가상 함수를 쓰지 않으면서 상속을 받는 방안도 고려되었지만 디버깅(debugging)의 편의성을 위해서 배제되었다. 디버깅 작업에서 가장 빈번히 값들을 확인하는 것이 이들 벡터들인데, 대부분의 디버깅 도구에서 매번 상위 클래스를 열어 보아야 하는 불편함이 있기 때문이다.

행렬은 일반적인 행렬과 좌표 변환 행렬을 별도로 구현하였는데, 좌표 변환 행렬은 자주 쓰이고 계산 속도를 필요로 하는 경우가 많기 때문이다.

수학 커널에서 지원하는 기초 도형은 직선, 선분, 타원, 평면, 베지어 곡선 등이다. 이러한 기초 도형은 기

하 커널에 넣을 수도 있었지만, 기하 커널의 다양한 곡선, 곡면의 내용이 필요하지 않는 응용 프로그램이 많기 때문에 수학 커널에서 구현하였다. 자유 곡선을 표현하고자 하는 욕구를 충족시키기 위해 구현된 베지어 곡선은 매개변수 곡선 중에서 데이터 양이 적으면서도 엔지니어링 측면에서 직관적인 형상의 모델링이 가능하기 때문이다⁽⁷⁾.

2.2.3 점열 곡선 커널

점열 곡선은 기하 커널과 별도로 구현되었는데, 기하 커널의 곡선과는 쓰임새가 다르다고 판단하였다. 주요 기능으로는 2차원 영역을 정의하고 처리하는 부분인데, 2차원 영역의 내부와 외부를 판별하거나 폐곡선의 방향(시계 방향 혹은 반시계 방향)을 판별하는 기능이다.

2.2.4 기하 커널

점, 곡선, 곡면 등의 기하 형상을 처리하는 커널로 이들 기하 형상 데이터를 관리하는 구조인 파트(part)와 모델(model)을 포함하고 있다. 파트는 점, 곡선, 곡면의 집합이며, 이들 파트의 집합이 모델이다. 모델은 조립품에 대응하고 파트는 부품에 해당하는 개념으로 볼 수 있다. 모델과 파트 및 기하 요소의 저장 구조는 Fig. 2와 같다.

매개변수식 곡선 및 곡면은 대부분 NURBS^(6,9)로 변환되어 계산될 수 있음에도 불구하고 다양한 매개변수식 곡선 및 곡면이 구현되었다. 외부에서 입력되는 곡선, 곡면 식이 다양하기 때문에 데이터 변환을 위해 임시로 저장할 데이터 구조가 필요하기 때문이다. 따라서 모든 곡선, 곡면은 NURBS로 변환하는 연산자를 갖고 있다. 그런데, 회전 곡면의 경우 NURBS 곡면으로 변환 가능하지만 정의역(domain)을 완전히 일치시키는 어렵다. 따라서 회전 곡면의 정의역 값을 초기값으로 해서 자코비언 인버전(Jacobian inversion)⁽⁷⁾을 통해 정확한 NURBS의 정의역 값을 구하는 방식을 썼다. 기하 커널에서 지원하는 곡면과 곡선의 종류는 다

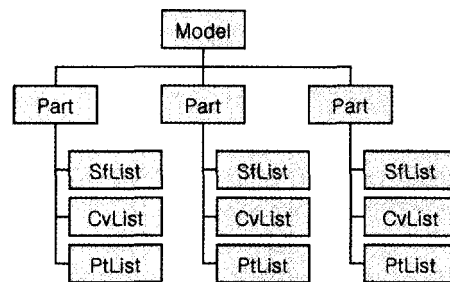


Fig. 2. Hierarchy of geometric objects.

음과 같다.

- 곡선: Line/Elliptic Arc/PtCurve/Bezier/Power-basis Polynomial/NURBS/Composite Curve
- 곡면: Plane/Bezier/Power-basis Polynomial/NURBS/Trimmed Surface

곡면들의 인접 관계를 저장하는 위상정보^[10,11]는 half-edge 데이터 구조^[12]로 구현되었으며, 재단곡면(trimmed surface)^[13] 구조에 포함되어 있다. 그러나 복잡한 위상 연산을 수행하는 것이 본 커널의 목적이 아니기 때문에 관련 연산자는 아주 기본적인 것들만 구현되었다. 그리고 데이터 변환기를 통해 외부에서 형상 데이터를 가져오는 경우 위상 정보를 갖고 있지 않은 경우가 많기 때문에 위상 정보를 복원할 수 있는 기능을 포함하고 있다. 그러나 본 연구에서 구현된 위상 정보 복원 기능은 초보적 수준이며, 완벽한 위상 정보의 복원을 위해서는 많은 추가 연구가 필요하다.

2.2.5 데이터 변환기

대표적 표준 중립 파일인 IGES^[14]를 읽고 쓰는 부분과 CATIA^[15] 등의 상용 CAD/CAM 시스템에서 데이터를 가져오는 부분이 있다. IGES(Initial Graphics Exchange Specification) 파일은 미국 상무부에서 제정한 파일 형식인데, 형상 데이터 교환의 실질적인 세계 표준으로, 거의 모든 CAD/CAM 시스템에서 지원하고 있다. 그러나 많은 상용 CAD/CAM 시스템의 IGES 변환기가 자체적인 결함이 있거나 필요한 정보를 가져오지 못하는 경우가 있다. 또, 사용의 불편함 때문에 직접적인 데이터 변환을 요구하는 경우가 많다. 상용 CAD/CAM 시스템에서 데이터를 변환하는 경우에는 해당 시스템의 API(Application Protocol Interface)를 이용해서 중립 파일을 쓰는 방식을 채택하고 있다.

3. 중립 파일(Neutral File)

본 절에서는 커널의 데이터를 파일로 저장하거나 응용 프로그램이 커널을 통해서 데이터를 파일로 저장할 때 사용되는 커널의 중립 파일 형식에 대해서 설명하고자 한다. IGES 혹은 STEP과 같은 표준 중립 파일의 형식을 차용 할 수도 있지만 확장성이 떨어지고 다양한 비형상 정보를 저장하기 곤란하기 때문에 별도의 중립 파일을 고안하게 되었다. 고안된 중립 파일은 확장성과 범용성을 가장 중요한 원칙으로 삼았다. 그 결과로 하위 버전과의 호환성을 보장하며, 다른 응용 프로그램에서 생성된 파일에서도 필요한 데이터를 얻어 낼 수 있다.

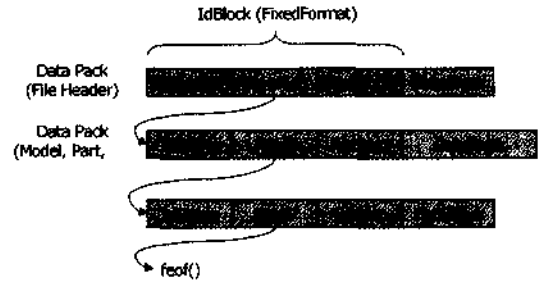


Fig. 3. Structure of neutral file.

본 연구에서 고안된 중립 파일은 Fig. 3과 같은 구조의 이진 파일(binary file)이다. 그림에서 보듯이 개념적으로는 '데이터 묶음(data pack)'의 단일 연결 리스트(single-linked list)로 구성되어 있다. 파일의 앞머리에는 항상 '파일 헤더 데이터 묶음'이 있어야 하고, 마지막 데이터 묶음은 파일의 끝(end of file)을 가리켜야 한다. 하나의 데이터 묶음은 '아이디 블록(ID block)'과 '데이터 블록(data block)'으로 구성된다. 아이디 블록은 형식을 바꿀 수 없으나 데이터 블록은 자유로운 형식을 가진다. 특정 응용 프로그램에서 커널에서 제공하지 않는 새로운 유형의 데이터를 저장할 경우에 새로운 아이디 문자열을 등록하고 데이터 블록의 형식을 나름대로 정하면 된다.

각 데이터 블록의 크기(size of data)를 데이터 묶음에 저장하면 굳이 리스트와 같은 구조를 갖지 않아도 다음 데이터를 찾아 갈 수 있다. 그러나 기하 요소들은 같은 종류의 객체 일지라도 실제 데이터의 크기는 다르므로 데이터 크기를 계산하는 연산자가 별도로 필요하다. 다양한 종류의 객체에서 데이터 크기 계산자를 항상 올바르게 유지하는 부담을 덜기 위해 본 연구에서는 리스트 형태의 파일 구조를 사용하였다.

참고로 '파일 헤더' 데이터 묶음의 아이디 문자열은 'CbNF'이며, 데이터 블록의 내용으로는 바이트 순서, 출력 날짜, 출력 시스템의 명칭 및 주석문이 있다. 바이트 순서는 이진 파일에서 1 바이트를 넘어가는 데이터의 바이트가 오른쪽에서 채워지는지 왼쪽에서 채워지는지를 저장한다.

4. 프로그래밍 표준

커널 개발을 진행함에 있어서 프로그래밍 스타일과 문서화의 표준안을 정립하였다. 이를 통하여 개발된 소스 코드의 가독성(readability) 향상하고, 개인적인 코딩 스타일 개발의 노력 절감 및 코드의 이식성(portability)을 향상하여 궁극적으로는 개발의 생산성을 도모하였다.

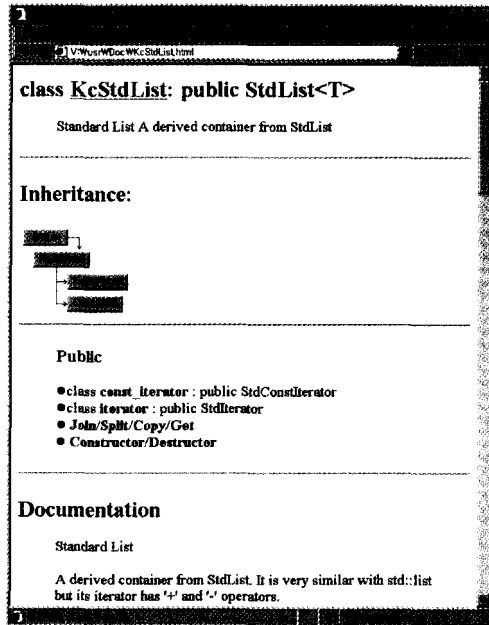


Fig. 4. Example of program documents.

4.1 프로그래밍 스타일

프로그래밍 스타일의 표준은 다음과 같은 원칙을 염두에 두었다.

- 표준을 따르는 것이 개발에 도움이 되게 한다.
- 가독성이 효율성에 우선 한다.
- 명백한 것을 더 명백하게 하려고 노력하지 않는다.

세부적인 스타일의 표준으로는 객체 클래스, 변수, 함수 등의 이름을 짓는 방법에 관한 것과, 코딩 스타일에 관한 것이 있다.

4.2 문서화 표준

가능한 별도의 문서와 설명이 필요 없는 프로그래밍을 원칙으로 하며, 설명이 필요한 경우에는 소스 코드에 주석으로 문서화를 하도록 했다. 소스 코드의 주석문은 Doc++^[16]을 이용해서 HTML 형식의 구조화된 문서를 제공한다. 소스 코드 내부의 많은 주석은 소스 코드를 지저분하게 만드는 경향이 있지만, 소스와 문서의 일치성을 보장하고 유지보수의 장점이 있다. Fig. 4는 만들어진 문서를 웹 브라우저로 본 예제이다.

5. 결 론

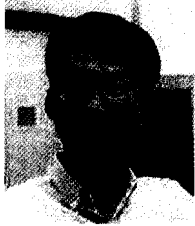
본 연구에서는 CAD/CAM 응용 시스템을 위해 개발

된 특화된 기능을 가진 가벼운 형상 커널 개발의 전반적인 내용을 소개하였다. 본 연구를 통해서 커널 개발이 소프트웨어 시스템 개발의 생산성을 높이는 좋은 대안임을 확인하였다. 특히 커널 개발을 통해 전체 시스템 개발 과정을 체계화 할 수 있었으며, 관련 경험 지식을 하나의 틀 속에 집적할 수 있었다.

본 연구에서 개발된 형상 커널은 회사 내외부의 여러 CAD/CAM 관련 상용 제품^[17,18]에 이식되어 성공적으로 사용되고 있으며, 학교에서 연구용으로 쓰고 있는 곳도 있다. 최근에는 개발된 형상 커널을 이용해서 가공 커널(machining kernel)을 개발하고 있다.

참고문헌

1. <http://www.spatial.com>.
2. <http://www.parasolid.com>.
3. <http://www.ricoh.co.jp/designbase>.
4. <http://www.opencascade.com>.
5. Goult, R. J. and Sherar, P. A.(Eds.), *Improving the Performance of Neutral File Data Transfers*, Springer-Verlag, 1987.
6. Musser, D. R. and Atul Saini, *STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library*, Addison-Wesley, Reading, MA, 1996.
7. Choi, B. K., *Surface Modeling for CAD/CAM*, Elsevier, 1991.
8. Les Piegl and Wayne Tiller, *The NURBS Book*, Springer, 1995.
9. 정형배, "NURBS Surface Global Interpolation에 대한 한 방법:II," 한국 CAD/CAM 학회논문집, 제3권, 제4호, pp. 243-250, 1998.
10. 이상현, 이진우, "비다양체 형상 모델링을 위한 간결한 경계 표현 및 확장된 오일러 작업자," 한국 CAD/CAM 학회논문집, 제1권, 제1호, pp. 1-19, 1996.
11. 김석환, 이진우, 김영진, "비다양체 모델을 수용하는 CAD시스템 커널을 위한 볼리안 조작의 개발," 한국 CAD/CAM 학회논문집, 제1권, 제1호, pp. 20-32, 1996.
12. Martti Mantyla, *An Introduction to Solid Modeling*, Computer Science Press, 1988.
13. 김형일, 채수원, "트립 곡면상에서 사각형 요소망의 자동 생성," 한국 CAD/CAM 학회논문집, 제4권, 제2호, pp. 153-161, 1999.
14. *The Initial Graphis Exchange Specification (IGES) Version 5.2*, IGES/PDES Organization, 1993.
15. <http://www.catia.com>.
16. <http://www.zib.de/Visual/software/doc++>.
17. <http://www.cubictek.com>.
18. <http://www.ets-soft.com>.



정 연 찬

1989년 한양대학교 산업공학과 학사
 1991년 KAIST 산업공학과 석사
 1996년 KAIST 산업공학과 박사
 1998년 DaimlerChrysler 연구원
 1991년~현재 큐빅테크 연구소 부소장
 관심분야: 가공경로생성, 형상모델링, 컴퓨터 그래픽스, 시스템 통합



박 준 철

1993년 서울대학교 산업공학과 학사
 1995년 KAIST 산업공학과 석사
 1995년~현재 큐빅테크 연구소 선임연구원
 관심분야: 형상모델링, 솔리드모델링, 데이터교환