

유한요소 구조해석 프로그램의
전후처리 통합 운영 시스템을 위한 객체지향적 모델
Object-Oriented Models for Integrated Processing System
of Finite Element Structural Analysis Program

서 진국* 송 준엽** 신 영식*** 권 영봉****
Suh, Jin-Kook Song, Jun-Yeop Shin, Young-Shik Kwon, Young-Bong

ABSTRACT

The pre- and post-processor for finite element structural analysis considering the user-friendly device are developed by using GUI. These can be used on WINDOWS' environment which is realized the multi-tasking and the concurrency by object-oriented paradigm. They are designed to control integratedly the pre-processing, execution and the post-processing of the finite element structural analysis program on multiple windows. These object-oriented modeling approach can be used for complex integrated engineering systems.

1. 서 론

PC용 유한요소해석 프로그램들의 개발이 활발해짐에 따라 다양한 사용자를 위해, PC에서 구동할 수 있는 편리한 사용자 환경의 개발이 절실히 요구되고 있다. 이 중 사용자 대화 방식은 사용의 용이성, 정보의 표현력 등 여러가지 인간적 요소를 고려해서 구성되며, 그래픽 화면출력, 메뉴를 이용한 입력, 마우스를 통한 선택 등 여러가지 대화식 유형을 말한다. 유한요소해석을 위한 응용 프로그램은 이러한 사용자 환경을 고려한 전후처리기의 설계가 필수적이다. 이러한 대화식 소프트웨어는 사용자 접속장치를 응용 프로그램으로부터 독립적인 방법으로 사용해야 하는데, 이를 위한 설계방법론과 개발환경이 필요하게 되었다. 이에 대해, 여러가지 응용영역으로부터 프로그램들을 개발하고 통합하는 단일화된 개념으로서 최근에 개발되어 컴퓨터 공학의 많은 분야에서 대중성을 얻은 객체지향적 접근방법이 적용되는 것은 필연적이다. 이는 설계에서의 재사용성, 캡슐화의 원리와 신속한 원형제작, 반복적 개발과 같은 기법들을 사용함으로써 사용자 접속장치를 통합할 수 있는 요소들로 다시 세분할 수 있고, 사용자는 대화식 요구조건이 유사한 다른 응용 프로그램에서는 다시 사용될 수 없도록 신속하게 접속장치를 수정할 수 있으며, 응용 프로그램을 정의하는 코드에 대해 어떠한 영향도 끼치지 않고 코드를 변경하거나 재위치시킬 수 있는 등 여러가지 유리한 측면을 가지고 있다.

이에 본 연구는 효율적인 유한요소해석용 전후처리기를 설계하는데 있어서 유한요소를 위한 모델과 이것의 그래픽 표현을 위한 모델을 서로 통합하는 시스템을 윈도우즈 환경에서 객체지향적 설계방법론을 사용하여 해결하고자 하며, 이는 앞으로 공학의 여러 분야를 통합하는 설계 시스템 등에 기초가 될 것이다.

* 영남대학교 공업기술연구소 연구원
** 영남대학교 대학원 토목공학과 박사학위과정
*** 영남대학교 공과대학 토목공학과 교수
**** 영남대학교 공과대학 토목공학과 조교수

2. 개발환경

2.1 객체지향 설계방법론

객체지향 설계방법론은 응용영역의 모델을 구축하고 시스템을 설계하는 동안에 실행 세부사항들이 첨가되어 구성된다. 이러한 접근방식을 객체 모델링 기법(OMT)라고 하며, 방법론은 다음 단계를 가진다.

(1) 분석(analysis): 문제의 기술로부터 시작하여, 그것의 중요한 성질들을 보여주는 실세계 상태의 모델을 구축한다. 모델에서의 객체는 응용영역 개념이어야 하며, 자료구조와 같은 컴퓨터 실행 개념이 아니다.

(2) 시스템 설계(system design): 시스템 설계동안에 목표시스템은 분석구조와 제안된 구성 둘다에 근거한 하위시스템들로 조직된다.

(3) 객체 설계(object design): 객체설계는 분석모델에 기초하여 설계모델을 구축한다. 시스템 설계동안 확립된 전략에 따라 설계모델에 세부사항들을 첨가한다. 객체설계의 초점은 각 클래스를 실행하는데 필요한 자료구조와 알고리즘이다. 응용영역 객체와 컴퓨터 영역 객체는 비록 다른 개념적 수준에 존재하지만, 둘다 동일한 객체지향적 개념과 기호를 사용하여 나타내진다.

(4) 실행(implementation): 객체설계동안에 개발된 객체의 클래스들과 관계들은 최종적으로 특정 프로그래밍 언어, 데이터베이스 또는 하드웨어 실행 등으로 전환된다. 모든 어려운 결정은 설계시 이루어질 것이기 때문에, 프로그래밍은 개발주기에서 비교적 중요하지 않고 기계적인 부분일 것이다. 목표 언어는 설계결정에 다소 영향을 미치나, 설계는 프로그래밍 언어의 세부적인 사항에 의존되지 않는다.

객체지향적 개념이 시스템 개발 생명주기, 즉, 분석으로부터 설계를 통하여 실행까지를 전 단계에서 적용될 수 있다. 즉, 동일성, 분류, 다형성, 상속성 등의 객체지향적 개념들이 전 개발주기를 통하여 적용된다. 이 중에서 시스템과 객체의 설계를 위한 객체지향적 분석은 프로그래밍의 객체지향적 활용에 대한 중요성을 획득할 수 있는 시스템 분석의 한 방법이며, 이는 객체와 속성, 분류와 상속, 구조의 조립이나 성분구성과 같은 조직의 세가지 기본적인 방법에 기초한다. 또한, 객체의 속성, 관계, 구조, 표현 등과 속성과 처리방식의 캡슐화, 객체의 분류구조, 상속에 의한 클래스들 사이의 공통성 표현 등과 같은 전형적인 실체의 인식을 포함한다. 이러한 분석은 다른 형태의 분석, 예를 들어 기능적 분해 등과 달리, 문제공간에서부터 모델까지의 직접적인 매핑을 허용한다. 이것은 처리방식의 적용이, 조직, 메시지와 통신, 객체에 의해 제공되는 처리방식에 대한 전반적인 구성 등에 근거함으로써 성립된다. 분석은 객체들의 인식, 클래스, 하위클래스와 이들의 상속, 속성의 정의, 처리방식과 메시지 연결의 정의와 같은 단계를 거쳐 수행된다.

다시 말하면, 일반적인 구조해석 문제에 유한요소법을 적용하는 것은 소프트웨어 설계에서 고도의 개념적 명확성을 요구한다. 이를 구현하는데 있어서의 확장성과 유연성을 달성하기 위해서는 프로그래머가 세부사항들까지 신경쓰지 않아도 되는 고도의 추상화를 사용하여 그 복잡성을 없앨 필요가 있다. 이러한 접근방법은 객체지향적 설계 및 프로그래밍에 의해 지원되며, 이는 코드의 재사용을 촉진하는 유한요소의 분류를 제공한다. 또한 캡슐화와 일반적인 동적 자료구조, 추상 자료형으로 유연성을 실현할 수 있다. 구체적으로는 클래스를 도입하여 추상화 개념을 구현한다. 하나의 클래스는 여러 개의 인스턴스를 가질 수 있으며, 모든 인스턴스는 클래스 추상화 개념을 공유할 수 있으나, 그것들의 동일성과 상태에 있어서는 일반적으로 서로 다르다. 또한, 클래스는 상속이나 계층구조의 블록을 구축하는데 사용될 수 있으며, 문제의 분류는 문제 영역에서의 추상화를 확인하고, 객체들의 명료한 상호작용에 대한 수단을 제공하는 추상화의 기능적 성질이라고 할 수 있는 메카니즘을 정의한다. 유한요소법의 분류를 통한 추상화에는 유한요소 소프트웨어의 정형적 표현의 필수적인 요소들로 구성된 1차적 추상화, 즉, 절절, 자유도, 요소 등 유한요소법이 공통적으로 가지는 것들이고, 하중, 구속조건, 형상함수, Gauss점, 재료적 성질, 응력 및 변형률 상태 등과 같이 선택적이고 특정 분야의 응용에 관한 내용들인 2차적 추상화로 나눌 수 있다.

또한, 객체지향 프로그래밍 방식은 자료 추상화를 통한 오류수정의 용이성과 시스템 설계시 객체와 클래스별 분산으로 신속한 원형제작을 가능하게 하는 등 점진적 프로그래밍 방식을 가지며, 상속성을 이용한 반복적 과정의 특성을 가진다. 또한 사용자 접속장치 생성 시스템에서 흔히 사용되는 입출력 루틴을 상속성 및 캡슐화의 개념으로 효과적으로 정의 가능케 하며, 표준화

된 처리방법들을 라이브러리 형태로 미리 제공하여 이러한 상속성에 근거한 일련의 변형에 의해 프로그램 작성을 용이하게 하는 등 프로그래밍의 용이성과 향상된 기능의 접속장치를 지원한다.

2.2 윈도우즈 환경에서의 전후처리

사용자 접속장치 개발은 기본적인 그래픽 입출력 요소들을 조절하는 기능을 가지는 그래픽 패키지, 그래픽 패키지를 확장시켜 하나 이상의 응용 프로그램 대화를 만들어 내고 관리할 수도 있게 하는 윈도우잉 시스템, 프레임워크, 사용자 접속장치 구축 세트 등의 사용자 접속장치 개발환경(UIDE)을 이용하여 많은 대화식 소프트웨어의 개발을 자동화시키고, 또한 사용자가 시스템과 어떻게 대화할 것인지를 결정할 수 있게 하며, 사용자 접속장치의 신속한 원형제작과 개발에 필요한 기반을 제공한다. 사용자가 마우스와 그림을 이용하여 모든 명령을 입력가능케 하는 그래픽 사용자 접속장치(Graphic User Interface; 이하 GUI)는 화면안에서 아이콘이나 윈도우를 통해 프로그램을 수행하고, 화면 축소, 확대, 이동이 가능하며, 메뉴를 사용한 명령입력, 대화상자를 이용한 자료입력을 통하여 현재상태나 명령이 화면에 그림으로 표시되어 사용자가 이러한 내용을 기억할 필요없이 PC를 사용할 수 있게 한다. 그 중 윈도우즈 응용 프로그램은 모든 응용 프로그램이 동일한 윈도우즈 메카니즘을 이용하여 풀다운 메뉴나 대화상자, 아이콘 동작 등이 모두 같기 때문에 사용자 접근이 용이하다. 이와 같이 윈도우즈(Windows)는 강력한 그래픽을 이용한 GUI기능을 PC에 구현하며, 뛰어난 메모리관리와 진정한 다중처리기능을 제공한다.

본 프로그램은 다중 윈도우즈를 이용하여 입력자료, 구조물의 기하적 형태, 문자와 그래픽 정보 등 여러가지 정보를 입력과정중 동시에 확인가능하게 하여 입력중 오류를 방지하며, 이러한 입력자료와 여러가지 출력결과를 동시 비교가능케 함으로써 좀 더 편리한 유한요소 구조해석용 전후처리를 개발하여, 이를 윈도우상에서 실행 프로그램과 통합 운영할 수 있는 기반을 마련하고자 한다. 본 전후처리 프로그램은 OWL(ObjectWindows Library)을 기초로 하는 입출력 윈도우 클래스와 입출력 그래픽 클래스를 기반 클래스로 하여 다양한 클래스를 생성한다. 여기서, 입력을 받지 않고 주어진 프로그램을 수행하는 후면 처리는 입력 그래픽 윈도우가 되며, 입력이 이루어지는 전면 처리는 입력 텍스트 윈도우가 되는 협조 다중처리를 구현한다. 이 두 윈도우들은 객체 연결 및 결합기법(OLE)을 통하여 두 개의 독립적인 응용 프로그램이 메시지 전달에 의해 자료를 교환한다.

본 프로그램의 컴파일러(compiler)는 Borland C++ 3.1을 사용하였고, IBM호환기종인 2MB 이상의 메모리를 가진 386급 이상의 하드웨어와 MS-WINDOWS 3.1환경하에서 구동되도록 설계하였다.

2.3 C++언어

객체지향적 프로그래밍은 본질적으로 절차적 프로그래밍보다는 덜 효율적이다. 다시 말하면, 객체지향적 프로그래밍은 코드 조직이 가능한 인간의 사고방식과 밀접하게 되어 있는 고수준의 소프트웨어 설계를 조장하며, 반면에 절차적 프로그래밍은 기계가 연산과 자료를 조직하는 방법에 더 가까운, 훨씬 더 실행지향적이어서, 보통 계산적으로는 더 효율적이 된다. 그러나, 복합 언어인 C++은 C의 객체지향적 확장이고, 그래서 C의 모든 특징들을 포함하며, 그 효율성 때문에 수치적 문제를 해결하는데도 아주 적합하다. 또한, C++과 같은 전형적인 객체지향 언어는 일반적으로 OOP언어의 필수적인 특징들인 클래스, 상속성, 실행시 다형성은 물론이고, 가상함수, 연산자 및 함수 중첩, 다중 상속 등과 같은 몇가지 다른 특징들도 제공해 준다. C++은 각 클래스가 간단하고 잘 정의된 접속을 가지며, 그 접속에 영향을 주지 않고 수정 또는 조정될 수 있게 하는 클래스의 라이브러리들을 구축하게 해 준다.

C++이 프로그램 개발의 절차적, 객체지향적 방법 둘다를 지원하는 것은 공학용 소프트웨어에 대해 매우 바람직하다. 소프트웨어의 객체지향적 개발은 물리적 시스템에 대한 모델내 객체들의 모듈들에 근거한다. 소프트웨어의 절차적 개발은 실세계 시스템의 기능에 대응하는 모듈들에 대한 것이다. 이러한 방법은 둘다 공학용 소프트웨어 개발에 필수적이라고 할 수 있다. 그래서, 이러한 C++의 이점은 구체적으로 유한요소설계와 그래픽 프로그램의 설계에 전적으로 이용되며, 아주 적합하다고 할 수 있다.

3. 객체지향적 모델

구조해석과정은 두개의 분리된 단계로 모델화될 수 있다. 즉, 구조 시스템의 정의와 불연속 수치적 모델의 해석이며, 이 두 모델사이의 전달은 자동적으로 다뤄질 수 있다. 본 논문에서는 시스템의 정의, 요소망 생성, 수치해석, 결과의 시각화 등을 유한요소 모델과 그래픽스 모델사이의 상관관계를 통하여, 객체지향적 접근방법에 근거한 적절한 소프트웨어 구성을 제시한다.

3.1 유한요소 모델

요소의 성질은 그것의 기하형상, 운동학적 모델, 재료적 성질, 형상함수 등에 의해 정의될 수 있다. 이것들은 기본적으로 직교성이기 때문에 분리된 클래스들은 각각에 대해 정의된다. 유한요소 자체도 표현적 모델과 역학적 모델로 나누어 진다. 모든 종류의 유한요소해석을 위한 구성 요소에 공통되는 자료와 처리방식들을 포함하는 FEM클래스와 같은 표현적 모델은 요소와 구조적 수준사이의 유일한 접속역할을 한다. ElemLib클래스의 중심적인 내용은 역학적 모델들이 동일한 정식화 과정을 통하여 각각의 요소로 변형되어 정의된다는 것이다. 이는 다른 종류의 요소 매트릭스 계산에 대한 처리방식들을 하위클래스에 상속될 수 있다. 또한 그에 상응하는 형상함수 등을 제공하는 부수적인 클래스들이 요소형식에 따른 하위클래스에 부속된다. 그 밖에 Matrix, Vector 클래스들과 같은 utility클래스들은 모든 ElemLib클래스의 하위클래스에서 사용되는 부수적인 클래스들로 구성된다.

유한요소의 모델링은 요소 형식, 절점의 선택, 자유도, 요소 자동생성 방법, 해석절차, 자료의 저장 등을 포함하는 많은 요인들에 따른다. 기하형태와 구조형식의 표면특성에 근거한 요소의 특징, 입력하중, 요소형식, 다항 차수 등은 특정 소프트웨어에 중요한 요인이다. 각 요소에 대한 강성도 및 질량 매트릭스를 나타내는 방정식은 매트릭스 형태로 유도되고, 매트릭스 대수는 체계적이고, 개념적인 방법으로 모든 요소의 상호작용 효과를 조합하는데 사용된다.

유한요소해석의 3단계는 다음 Fig.1에서 보여 주듯이, 먼저, 요소 매트릭스를 생성하고, 매트릭스 방정식들의 조립과 경계조건의 적용, 방정식의 해법 등으로 구성된다.

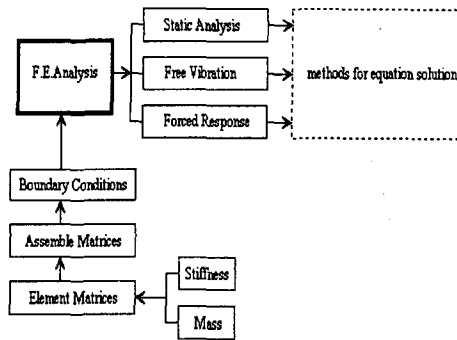


Fig.1 Overview of finite element analysis

재래식 언어에서는, 많은 수의 유한요소들로 구성된 유한요소 라이브러리를 개발하기 위해서는 요소의 성질들을 계산하는 루틴들이 각 새로운 요소에 대해 재작성되어야 하기 때문에, 많은 양의 코드가 중복되어야 했다. 그러나 객체지향 시스템에 의해 제공되는 메카니즘은 기존 요소들과의 차이점만 프로그래밍함으로써 쉽게 요소들을 첨가하게 해준다. 주어진 해석형식에 대해 형상함수가 한번 지정되면, 강성도 매트릭스와 하중벡터의 유도는 뚜렷한 개이를 따르기 때문에, 요소 개발시 그 처리과정을 컴퓨터에 위임할 수 있다. 다시 말하면, 매우 작은 양의 부가적인 코드로 요소들을 라이브러리에 빨리 첨가할 수 있다는 장점이 있다. Fig.2는 이러한 장점을 보여주는 상속에 의한 클래스 계층구조를 나타낸다.

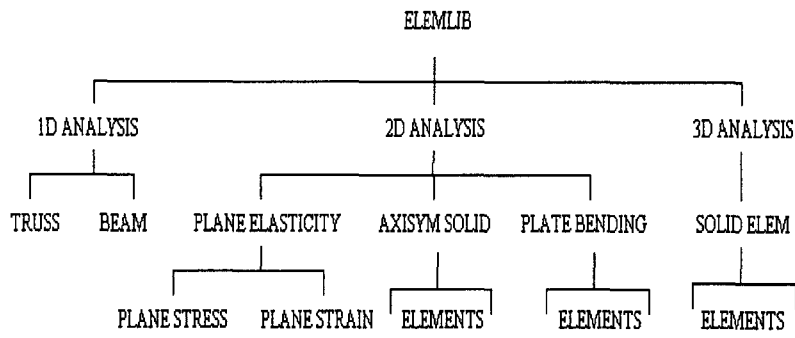


Fig.2 An illustrative finite elements hierarchy

ElemLib클래스는 모든 요소 클래스에 공통되는 속성들과 처리방식들을 통합한다. Load클래스와 그 하위클래스는 요소, 절점 또는 자유도에 작용하는 모든 하중들에 대해 수행된다. Member클래스는 요소의 재료적 성질을 관리하는 클래스이며, Node클래스는 절점의 위치와 절점하중을 조립하고, 그 자유도를 다루는 클래스이다.

이러한 클래스들은 기본 클래스인 FEM클래스의 하위클래스들로서, 편리하게 재분류될 수 있다. 이 클래스는 문제의 주된 속성인 요소, 절점, 재료, 하중 등을 통하여 요소망을 구현한다. Fig. 3은 ElemLib클래스의 한 하위클래스(Element_1)를 예로 들어 각 구성 클래스들이 어떻게 작용하는지를 보여 준다.

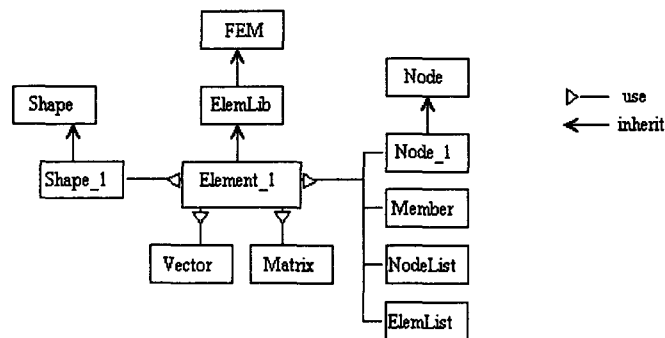


Fig.3 Class hierarchy of Element_1

3.2 그래픽스 모델

그래픽스 모델에서의 클래스들은 객체 인스턴스들의 생성을 위하여 형틀을 포함한다. 예를 들어, quadric클래스는 일반적인 2차 함수의 계수들 같은 자료영역들을 정의하고, spheres, cones 등과 같은 여러가지 2차 함수의 primitives의 인스턴스들을 허용할 수 있다. 이러한 자료들이 인스턴스 변수들로 취급되며, 이것들을 운용하는 여러가지 알고리즘들이 처리방식들이다. 여러가지 2차 함수의 primitives의 실제 인스턴스들은 이러한 형틀의 사용을 통한 응용 프로그램에 의해 생성될 것이다.

컴퓨터 그래픽스 프로그래밍에 대한 객체지향적 접근방법을 지원하는 여러가지 그래픽스 라이브러리들과 프레임워크가 개발되어 왔다. 이러한 것들은 하드웨어와 소프트웨어에 독립적인 객체지향적 그래픽스 시스템을 제공한다. 다음 Fig.4의 내용은 다중상속, 그래픽 클래스 계층, 다형성 등을 가지는 객체지향 언어를 통하여 그래픽 알고리즘 개발을 하게 한다. Fig.4는 2, 3차원 클래스 계층도를 보여준다. 이런 클래스의 통합체는 객체지향 설계의 고유의 이익, 즉 모듈

화와 자원공유에 의해 비교적 간단해야 한다.

기존의 능력들을 포함하고, 여러가지 응용들의 필요성을 도입하는 새로운 그래픽 응용 프로그램 개발은 객체지향적 개념 사용에 대한 장점들을 취하여 현재 연구중에 있다.

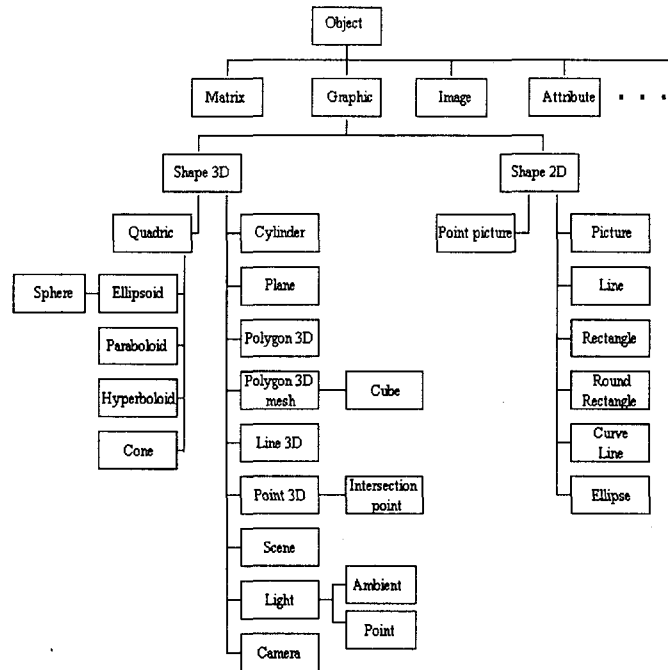


Fig. 4 Hierarchy of graphics models

3.3 각 모델들의 상관관계

유한요소의 초기 기하형상이나, 유한요소 모델에서 계산된 결과에 대한 변화된 형상들이 Node, Member클래스에 객체로 저장되고, NodeList클래스에 의해 연결되어, 그 정보가 Geometry클래스에 저장되며, 이는 그래픽스 모델로 전달된다. Fig. 5의 이런 유한요소 모델들과 2차원 그래픽스 모델의 클래스간의 연결구성을 보여준다. 여기서, Line클래스는 직선을 단순히 표현하는 것이며, Curve_line클래스는 곡선인 모든 객체들을 표현한다.

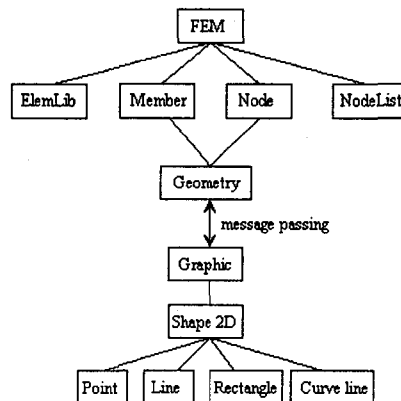


Fig. 5 Interrelationship between FE models and Graphics models

4. 실행예

다음의 윈도우들은 대화상자를 통해 입력된 자료들이 입력 윈도우상에 저장되어 축적된 후, 문자자료 파일 및 그래픽자료 파일이 생성된 결과를 보여준다. 다음 Fig.6,7과 같은 구조물을 본 전후처리기를 통하여 입출력한 결과, 메뉴를 통한 입출력이 편리하며, 다중 윈도우를 통하여 입력자료와 출력결과 및 여러가지 그래픽 정보 등을 한꺼번에 인식할 수 있어 여러 자료의 비교가 용이하다는 것을 알 수 있었다.

4.1 평면 뼈대구조의 정적해석

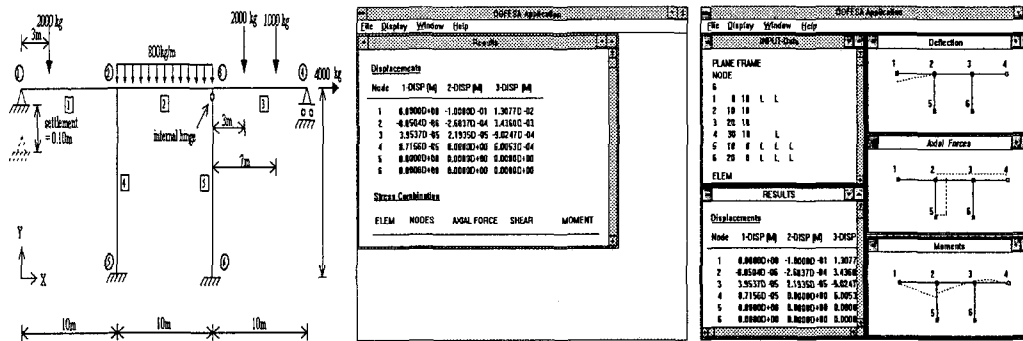


Fig.6 An Example of Text and Graphic Windows for a Plane Frame

4.2 캔틸레버 보의 자유진동해석

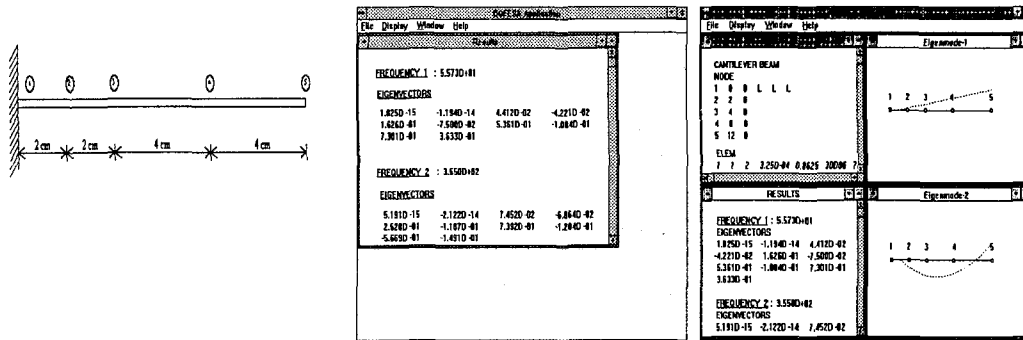


Fig.7 An Example of Text and Graphic Windows for a Cantilever Beam

5. 결론

본 연구에서는 C++언어를 사용한 객체지향 프로그래밍 기법으로 윈도우 환경에서 구동되는 유한요소 구조해석용 전후처리 프로그램을 개발하였다. 본 프로그램은 기존의 파일입력뿐만 아니라 사용자에게 친숙한 대화식 입출력 형태를 제공하며, 입출력 정보를 그래픽 처리하여 그 인식을 용이하게 하였고, 여러가지 문자정보와 그래픽정보를 다중 윈도우를 통해 연계, 운영할 수 있게 하였다. 또한 구조해석을 위한 입출력에 적합한 메뉴를 구성하여, 입출력 윈도우 클래스, 그래픽스 모델 등의 라이브러리와 매트릭스 클래스 및 유한요소 클래스들을 포함하는 유한요소 모델들을 상호 연관시켜 전후처리 및 실행을 윈도우상에서 통합운영할 수 있는 환경을 구현하였다.

이러한 객체지향 유한요소 모델링 접근방법이 복잡한 공학 시스템의 해결을 위한 다양한 통합 환경에 적용될 수 있을 것이며, 여기에서의 클래스들은 본 연구에서와 같이 공학문제들의 유한요소 해석을 위해 요구되는 기본적인 개념과 도구들을 형상화할 수 있게 한다. 본 연구는 고도의 모듈화, 재사용성, 확장성, 유지의 용이성이 있는 유한요소 소프트웨어 시스템의 새로운 개발을

위한 기초가 될 것이다. 이러한 객체지향적 개념의 특징들을 활용하여 앞으로의 효율적인 자료 관리를 위한 DBMS개발과 통합 설계시스템 개발에 확장되어야 할 것이다.

6. 참고문헌

1. 신영식, 서진국, 박영식, 최희옥, "PC용 객체지향 구조해석 프로그램의 개발", 한국전산 구조공학회지, 제5권 제4호, pp.125-132, 1992
2. 신영식, 서진국, "빠대구조물의 자유진동해석을 위한 객체지향 C++ 프로그램", 대한토목 학회 논문집, 제14권 제1호, pp.119-129, 1994
3. *Borland C++ 3.1*, Borland International Inc., CA., 1992
4. *Microsoft WINDOWS 3.1, Guide to Programming*, Microsoft Press, WA., 1992
5. *Object-Oriented Modeling and Design*, Rumbaugh. J. et al. Prentice Hall, NY., 1991
6. *Computer Graphics and Geometrical Modeling for Engineers*, Anand. V. B., John Wiley & Sons, NY., 1993