유한요소 구조해석을 위한 전후처리 통합운영 시스템에 관한 연구

A Study on Integrated Processing System for Finite Element Structural Analysis

서 진 국*
Suh, Jin-Kook
송 준 엽**
Song, Jun-Yeup
신 영 식***
Shin, Young-Shik

요 약

본 연구에서는 유한요소 구조해석을 위한 전후처리 통합운영 시스템을 개발하였다. 이 시스템은 구조해석, 전처리 및 후처리가 윈도우즈 환경에서 통합운영될 수 있도록 설계되었는데, 다중처리, 객체연결 및 결합 기법 등을 사용한 다중 윈도우상에서 대화식 입출력 기능과 여러가지 입출력 결과의 동시표현 등으로 더 향상된 그래픽 사용자 접속장치 환경을 제공한다. 따라서 메뉴, 대화상자, 다중 윈도우상에서의 단계별 자동입력 등을 통하여 자료의 입력이 용이해졌고, 동일 화면상에서 입력자료와 출력결과를 동시에 나타낼 수 있게 되었다. 본 시스템의 타당성과 효율성을 검증하기 위하여 여러가지 구조물에 대한 정적 및 자유진동해석을 수행하였다.

Abstract

An Integrated processing system for finite element structural analysis has been studied. It is designed to control integratedly the preprocessing, the execution and the postprocessing of a finite element structural analysis program on Windows. It becomes a better graphic user interface(GUI) for the concurrent representation of various inputs and outputs through the dialog-type on multi-windows by the multi-tasking and the object linking and embedding(OLE). Data input can be done easily through menus, dialog boxes and automatic stepwise inputs on the multiple windows, and then output results can be seen with input data on the same screen. Efficiency and validity of the system were examined by solving several numerical examples.

* 경동전문대학 토목과 전임강사
** 영남대학교 대학원 토목공학과 박사과정
*** 영남대학교 공과대학 토목공학과 교수

이 논문에 대한 토론을 1995년 9월 30일까지 본 학회에 보내주시면 1996년 3월호에 그 결과를 게재하겠습니다.

1. 서 론

지난 10여년간 대부분의 소프트웨어는 비대화식에서 고도의 대화식(dialog-type)으로 빠르게 발전되어 왔다. 이러한 대화식 소프트웨어는 응용 프로그램으로부터 독립적으로 사용할 수 있는 대화식 그래픽 사용자 접속장치(graphic user interface : GUI)환경을 지원하기 위하여, 분석 및 설계방법론이며 개발환경인 객체지향적 소프트웨어 개발 방법론(object-oriented software development methodology)[1]에서의 재사용성(reusability), 캡슐화(encapsulation)의 원리와 신속한 원형제작(prototyping), 반복적 개발과 같은 기법들을 사용한다. 이러한 기법들은 사용자 접속장치를 함께 묶을 수 있는 요소들로 세분할 수 있고, 사용자는 대화식에 대한 요구조건이 유사한 다른 응용 프로그램의 개발을 위해 신속하게 접속장치를 수정할 수 있으며, 응용 프로그램을 정의하는 코드에 대해 어떠한 영향도 끼치지 않고 코드를 변경하거나 재위치시킬 수 있고, 반복적인 방법으로 접속장치를 개발할 수도 있는 등 몇가지 유리한 측면을 가지고 있다. 프로그래밍 과정에서도 이러한 객체지향적 방식은 자료추상화(data abstraction)를 통한 오류수정의 용이성과 시스템 설계시 객체(object)와 클래스(class)별 분산으로 신속한 원형제작을 가능하게 하는 등 점진적 프로그래밍 방식을 가지며, 상속성(inheritance)을 이용한 반복적 과정의 특성을 가진다. 또한 이 방식은 사용자 접속장치 생성 시스템에서 흔히 사용되는 입출력 루틴도 상속성 및 캡슐화의 개념으로 효과적으로 정의 가능하며, 표준화된 처리방법들을 라이브러리(library) 형태로 미리 제공하여 이러한 상속성에 근거한 일련의 변형에 의해 프로그램 작성을 용이하게 하는 등 프로그래밍의 용이성을 통하여 향상된 기능의 접속장치를 지원한다.

사용자 접속장치 개발은 기본적인 그래픽 입출력 요소들의 조절기능을 가지는 그래픽 패키지(graphic package), 이 그래픽 패키지를 확장시켜서 하나 이상의 응용 프로그램 대화를 만들어 내고 관리할 수 있게 하는 윈도우잉 시스템(windowing system)과 그 외 프레임워크(framework), 사용자 접속장치 구축 세트 등의 사용자 접속장치 개발환경(UIDE)을 이용하여 많은 대화식 소프트웨어의 개발을 자동화시키며, 사용자가 시스템과 어떻게 대화할 것인지를 결정할 수 있게 하는 등 사용자 접속장치의 신속한 개발에 필요한 기반을 제공한다. 그 중 윈도우즈(Windows)[2] 응용 프로그램은 모든 응용 프로그램이 동일한 윈도우즈 메카니즘을 이용하여 풀다운(pull-down) 메뉴나 대화상자, 아이콘 동작 등이 모두 같기 때문에 사용자의 접근이 용이하다. 이와 같이 윈도우즈는 강력한 그래픽을 이용한 GUI기능을 PC에 구현하는 것 외에, 뛰어난 메모리 관리와 진정한 다중처리(multi-tasking) 기능도 제공한다.

현재 널리 사용되는 PC용 프로그램에는 SAP90[3], COSMOS/M[4], XETABS[5], MicroFEAP[6], MicroAIT[7] 등이 있는데, 이와 같은 기존의 PC용 구조해석 프로그램들은 주로 매크로 명령어(macro-command)를 통한 파일입력과 화면입력(text user interface ; TUI)이 대부분이며, 몇몇 대화식 유형을 취하는 것들도 각 부메뉴(sub-menu)를 독립적인 화면에 나타내는 단점이 있다. 이 중 SAP90은 매크로 명령어를 사용하여 구조물의 기하형태를 그래픽 화면을 통해 표현하는 등 GUI를 이용한 좀 더 진보적인 입력장치를 사용하고 있으나, 이것도 단일화면을 사용함으로써 문자자료와 그래픽자료를 동시에 비교하여 인식할 수 없으며, 전처리(preprocessor), 해석수행 및 후처리기(postprocessor)인 SAPLOT 등이 별도로 운영되고 있는 등 사용자 측면에서 개선해야 할 점이 많다고 할 수 있다.

따라서 본 연구에서는 다중 윈도우즈를 이용하여 입력자료, 구조물의 기하형태 등 여러가지 정보를 입력과정 중 동시에 확인가능하게 함으로써 입력중 오류를 방지하며, 이러한 전처리와 함께 구조해석 및 후처리를 윈도우상에서 통합 운영함으로써 입력자료와 해석을 통한 여러가지 출력결과를 동시에 비교가능케 하는 등 사용자의 편의를 고려하여 보다 편리한 유한요소 구조해석용 전후처리 통합 프로그램을 C++언어[8]를 사용하여 개발하였다.

2. 객체지향 유한요소 구조해석용 전후처리기의 구성

2.1 객체지향 유한요소 구조해석 프로그램과 입출력 윈도우즈와의 상관관계

Fig. 1은 뼈대 구조물의 정적 및 자유진동해석이 가능한 객체지향 유한요소 구조해석 프로그램(OOFESA)[9), 10)]과 전후처리기와의 상관관계에 대한 개념도이며, Fig.2는 윈도우즈환경에서 OOFESA의 클래스들과 전후처리기 클래스들 사이의 자료흐름을 통한 기능적 관계를 전반적으로 나타낸 것이다.

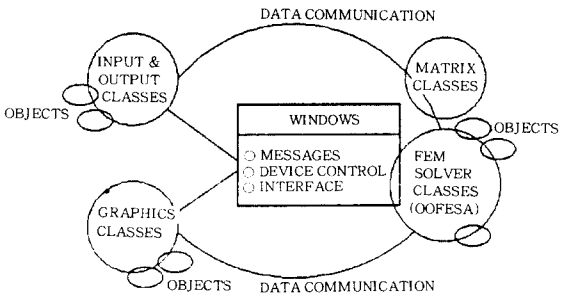


Fig.1 Conceptual diagram for interaction of system

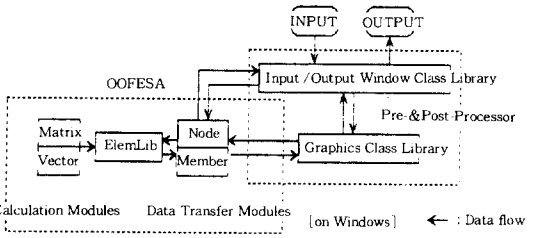


Fig.2 Overall functional structure of programs

2.2 입출력 윈도우 클래스 라이브러리

입출력 윈도우 클래스 라이브러리(input /output window class library)는 구조해석을 위한 여러가지 자료들을 윈도우를 통하여 사용자가 쉽게 입력할 수 있게 하는 클래스들을 포함한다. 이 클래스 라이브러리는 사용자와 컴퓨터간에 구조형태, 구속조건, 하중상태 등의 자료정보를 자유롭게 교환하도록 하기 위한 클래스들로서 윈도우즈의 특성인 협조형 다중처리(cooperative multi-tasking)와 객체연결 및 결합 기법(object linking and embedding ; OLE) 등을 이용하며, Borland C++[11]에서 제공되는 ObjectWindows Library(OWL)를 기반으로 하여 개발되었고, 이를

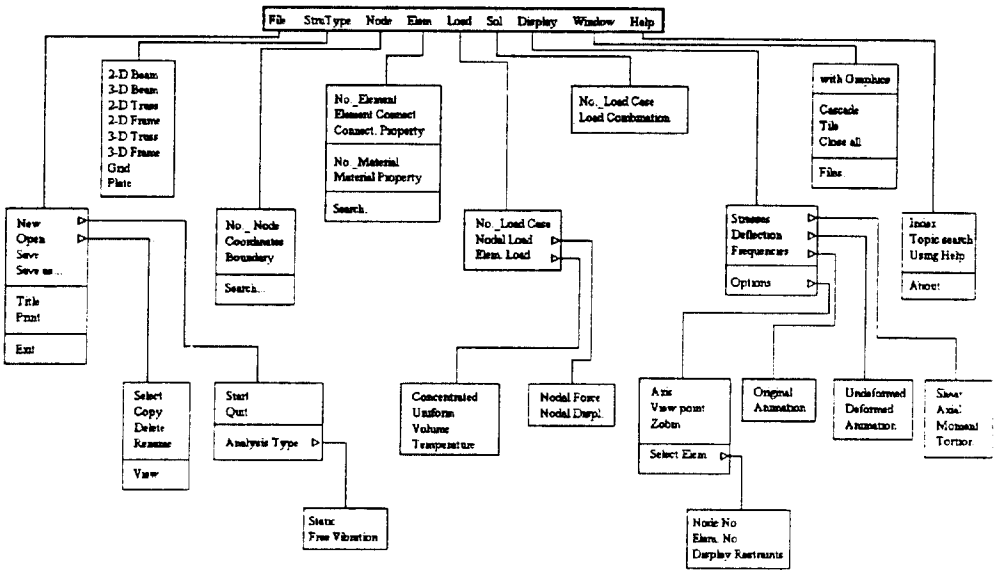


Fig.3 Architecture of Windows' menus for input and output

통하여 해석예비과정을 이해할 수 있도록 유도하는 대화식 입력형태와 이미 입력된 자료들을 참고할 수 있는 또 다른 윈도우들이 제공되며, 그래픽 클래스 라이브러리(graphics class library)와 연계되어 윈도우상의 여러가지 그래픽 입출력 결과를 보여주게 된다.

다음 Fig.3은 입출력 윈도우의 메뉴바(menu bar)에 나타나는 프로그램 메뉴와 각 메뉴별 부메뉴에 대한 계층구조를 나타낸 것이다.

입출력 윈도우 클래스 라이브러리를 구성하는 중요한 하위 클래스들은, Object와 TStream클래스로 부터 파생된 추상적 클래스이며 OWL의 계층구조에서의 여러 접속장치 클래스들의 부모 클래스(parent class)가 되는 TWindowsObject클래스, 사용자 접속 기능과 TFile-Dialog의 기능과 같은 특수한 대화상자 기능을 제공하기도 하고, 정형(modal) 혹은 비정형(modeless) 대화상자를 생성하는 인스턴스를 지원하며, 그 인스턴스는 라디오 버튼(radio button), 체크 박스(check-box)와 같은 콘트롤 클래스의 인스턴스를 포함하는 TDialog클래스, 그리고 프레임 윈도우와 클라이언트 윈도우 생성기능을 포함하는 TMDIFrame 클래스 등이 있다.

TWindowsObject에서 제공하는 기능에는 여러 OWL 응용 프로그램의 접속장치를 통제하여 효율적인 메시지의 처리와 윈도우를 종료시키는 기능 등이 있다. TMDIFrame클래스는 클라이언트 윈도우를 통제하는 메시지에 응답하는 멤버함수들로 구성되어 있다. 이러한 메시지 기능에는 클라이언트 윈도우의 아이콘을 배열하고 윈도우를 닫으며, 새로운 윈도우를 생성하는 등의 기능이 있다.

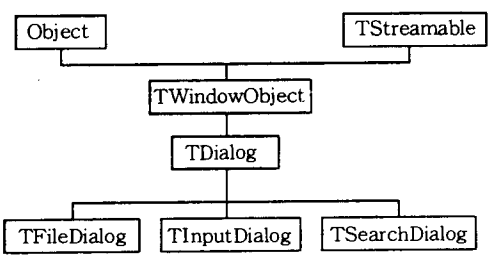


Fig.4 Hierarchy of classes for dialog-box

Fig.4는 OWL에서 제공하는 기본적인 대화상자의 계층구조를 나타낸 것이다. 본 전후처리기에서 사용하는 대화상자들은 Fig.5와 같이 TDialog클래스를 기반 클래스로 하여 생성되며, 여기서 TInfoDialog, TBaseDialog, TIndivDialog, TGenrDialog, TPropDialog 등은 선택된 구조형식에 대한 정보, 각 항목별 자료에 대한 기반 윈도우, 자료의 개별입력을 위한 대화상자, 자료의 자동생성을 위한 대화상자, 요소 성질의 입력을 위한 대화상자 등에 대한 기본 클래스들을 각각 나타내며, 예제해석에서 그 구현된 형태를 보여준다.

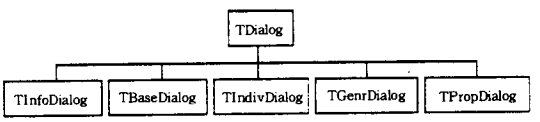


Fig.5 Classes for dialog-boxes of Input/ Output Window classes

본 전후처리기에서 사용하는 대화상자는 대부분 TInputDialog의 기본적인 형태에 몇가지 콘트롤이 추가된 형식을 가지는데, 일반적인 이러한 대화상자를 통한 정보의 교환은 먼저, 키보드를 통하여 편집 콘트롤(Edit Control)에 전달된 자료정보가 푸시버튼 콘트롤(Push-Button Control)로 인하여 윈도우즈 응용 프로그램에 전달되어 처리되는 유형을 취한다.

2.3 입출력 그래픽 클래스 라이브러리

계산결과를 한꺼번에 인식할 수 있도록 문자결과의 출력화면과 함께 응력도 및 변위도를 다중 윈도우를 통한 그래픽 화면으로 동시에 제공하며, 자유진동 모드도 사용자의 선택에 의해 화면출력할 수 있도록 하였다. 즉, 개별적 요소 및 절점에 대한 결과들인 절점변위, 요소응력, 단면력도 및 자유진동 애니메이션 등이 화상으로 표현된다. 또한, 이러한 결과출력을 통해 사용자가 입력오류를 자연스럽게 인지할 수 있도록 유도한다.

유한요소 구조해석을 위한 입력자료가 구조해석 클래스들과 입출력 그래픽 클래스로 부터 상속되어 도화처리되는 과정에서의 상관관계를 유한

요소 클래스의 한 객체인 평면 트러스(Truss2D)를 예로 들어 나타낸 것이 Fig.6이다. 또한 입출력 전반에 걸쳐 문자자료를 도화처리하는 과정을 나타낸 것이 Fig.7이다.

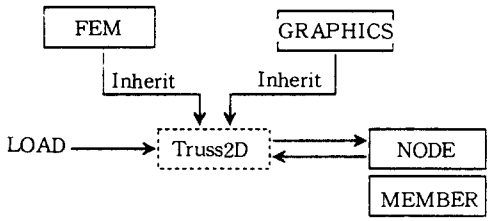


Fig.6 Inheritance of input data for structural shape

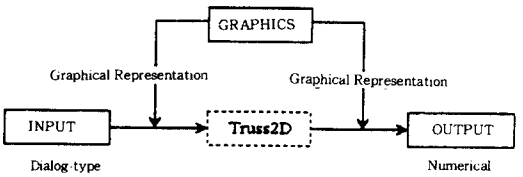


Fig.7 Graphic representation of input and output for each structural type

다음 Fig.8은 입출력 그래픽 클래스를 구성하는 2차원 그래픽 표현을 위한 기반클래스들의 계층도를 나타낸 것이다.

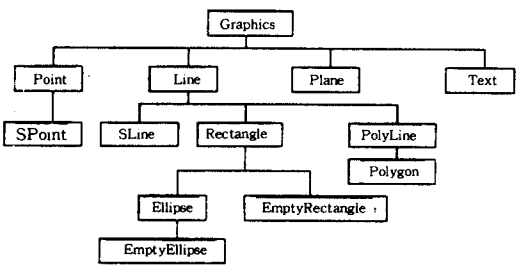


Fig.8 Hierarchy of 2D drawing tools

Fig.9는 이런 유한요소 모델들과 2차원 그래픽스 모델의 클래스들간의 연결구성을 보여주는데, 이러한 그래픽 클래스 모델들을 이용하며, 구조해석을 이한 유한요소의 초기 기하형상이나, 유한요소 모델에서 계산된 결과에 대한 변화된 형상들이 Node, Member클래스에 객체로 저장되고, NodeList클래스에 의해 연결되며, 그 정보가 접속역할을 하는 Geometry클래스를 통하여 그래픽 클래스 모델로 전달되어 화면상에 표현된다.

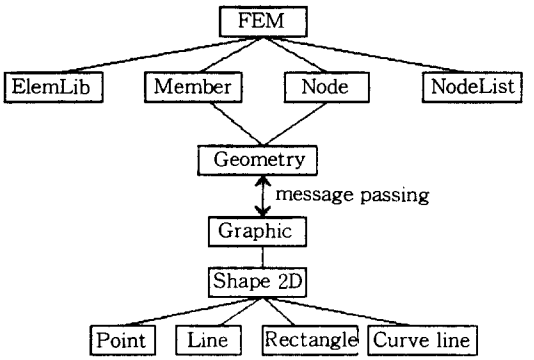


Fig.9 Interrelationship between FE models and Graphics models

2차원 그래픽 표현을 위한 클래스들은 Fig.8의 기반을 형성하는 Point클래스, Line클래스, Plane클래스 등으로 구성되는데, 그 하위클래스들인 Point2D클래스는 2차원 화면상의 점을 표현하는 기능을 가지며, Line2D클래스는 선분의 표현, Plane2D클래스는 2차원 평면을 각각 나타낸다. 이러한 클래스들은 3차원 표현에서도 기반이 되는데, 이 때는 또 다른 하위클래스로 Point3D, Line3D, Plane3D를 재정의함으로써 공간상에서의 점, 선, 면의 각 단위를 형성할 수 있다. 그 외에 화면좌표의 벡터 차원을 나타내는 Vector_g 클래스와 각 화상의 동작을 수행하기 위한 Matrix_g클래스가 수치적 문제를 다루는 Matrix, Vector 클래스들과 함께 그 하위클래스로 정의되며, 그래픽 모드에 관계없이 대상을 화면상에 표현하는 Screen클래스도 정의된다. 이와 같은 클래스들이 3차원 표현에도 기본이 되는 것을 고려하여 임의로 접근이 가능하도록 내부변수없이 모두 공용으로 선언하였다.

3. 입출력 단계 및 기능

입력형식은 연속입력모드(sequential mode)와 편집모드(editing mode)로 나뉘는데, 연속입력모드는 윈도우의 메뉴바 중 'File'메뉴의 'New'항목을 선택함으로써 도입되며, 이는 각 입력단계에서 대화상자를 통하여 자료를 입력하면 자동으로 다음 단계로 전개되어 새로운 프로젝트를 생성하는 경우를 말한다. 이 때 기본 윈도우에서는 텍스트 파일을 생성하며, 'Window'메뉴의 'with Graphics'를 선택하거나, 단축키에 의해 필요시 문맥 전환(context switching)할 수 있는 팝업(pop-up) 윈도우 형태로 나타나는 그래픽 화면에 해당 좌표값이 동시에 그림으로 표시된다. 반면 편집모드란 기존의 프로젝트 파일을 열어 문서 편집기 상에서 편집, 수정하거나 입력 윈도우의 해당 메뉴를 풀다운한 후 대화상자를 통해 수정할 수 있는 형식을 말한다. Fig.10은 연속입력모드에서의 평면 뼈대구조에 대한 입력단계를 예시한 것이다.

STEP 1	Select Starting Menu ; New
↓	
STEP 2	Select Analysis Type ; Static
↓	
STEP 3	Select a Structure Type ; 2-D Frame
↓	
STEP 4	Input Node Data - Coordinates of Nodes - Boundary Conditions
by Individual Input or Linear Generation	
↓	
STEP 5	Input Element Data - Element Connectivity - Element Properties
by Individual Input or Linear Generation	
↓	
STEP 6	Input Load Data
↓	
STEP 7	Solving & Results Output

Fig.10 An example of input stream in sequential mode

입력과정을 화면을 통해 단계별로 설명하면 다음과 같다.

다음 Fig.11은 주메뉴 'StruType'의 부메뉴들인 여러가지 구조형식 중 '2-D Frame'을 선택하는 것을 보여준다. 이로써 입력 윈도우에서 유한요소해석 클래스에 요소형식에 관한 메시지가 전달된다. 이 메시지에 의해 ElemLib클래스의 하위클래스인 Frame2D클래스의 객체가 생성된다. 이는 또한 다음 단계들에 자동적으로 나타나는 화면들의 조합을 지정해 주며, 입력자료들을 받아들일 준비를 한다. 다음 Fig.12의 화면은 선택된 구조형식에 대응하는 요소형식의 전반적인 정보를 보여준다. 이는 사용자가 미리 입력할 자료들의 전체적인 윤곽을 파악하게 해주며, 해당 요소형식에 대한 이해를 돕는다. 이러한 화면에는 유한요소해석 과정을 이해하는데 도움을 주는 여러가지 메시지나 프로그램 사용에 대한 지시어 등을 간단하게 포함할 수도 있다. 내용을 확인한 후 계속 자료입력할 의사가 있다면, 'OK'항목을 선택하면 다음 단계로 진행되며, 'Cancel'을 선택하면 초기화면 상태로 돌아간다. Fig.13에서 Fig.16은 요소의 기하학적, 재료적 성질들에 관한 정보를 입력하기 위해 절점자료, 요소자료, 하중자료, 재료적 성질 등을 입력하는 단계를 나타낸 화면이다. 여기서 개별입력이나 자동생성 기능을 이용한 입력 등을 각 정보마다의 대화상자와의 문답을 통해 자료를 축적하여 저장할 수 있다. Fig.13은 개별입력과 자동생성을 선택하는 버튼과 축적된 절점 정보들을 검색하는 스크롤바 등을 갖추고 있다. Fig.14는 대화상자를 통하여 각 절점의 좌표를 입력하는 형식을 보여준다. Fig.15는 선형 자동생성 기능을 통하여 절점좌표를 자동으로 발생시키는 모습을 보여준다. 이렇게 각각의 대화상자를 통하여 입력된 자료들은 Fig.13과 같은 자손 윈도우(child window)에 축적된다. 위와 같은 단계들을 통해 입력된 자료들은 기반 윈도우에 그 자료들이 축적되고, 이는 여러가지 자료를 통합하는 자료파일을 생성시킨다. 이것은 자료파일의 형식만 정해지고 이의 접속장치만 설계된다면 기존의 범용 프로그램들과의 자료교환이 가능하다는 것을 보여준다. Fig.16은 요소의 재료적 성질을 입력하는 대화상자이다. 입력의 마지막 단계로 기반 윈도우상에 파일이 생성되고, 이와 함께 입력 자료들을 검토

하기 위한 또 다른 그래픽 화면이 제공된다. 이 팝업 그래픽 화면은 주메뉴 'Window'의 'with Graphics'항목을 선택함으로써 나타나는데, 이를 통하여 입력오류를 확인할 수 있게 된다.

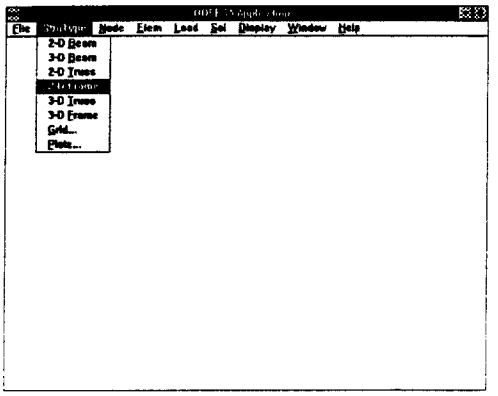


Fig.11 Screen for selecting a structural type

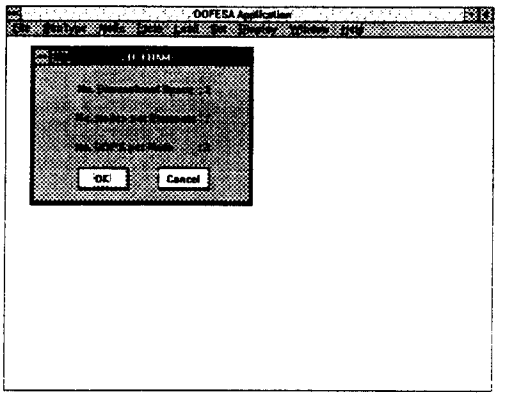


Fig.12 Screen for starting data input

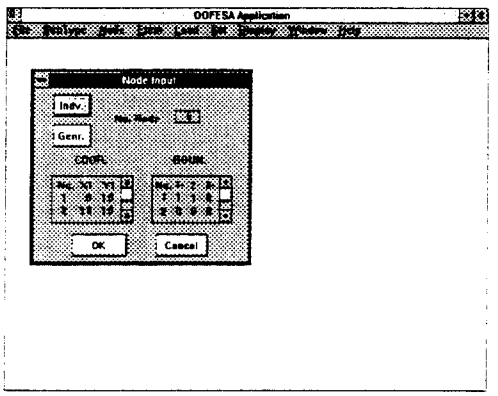


Fig.13 Screen for node input child window

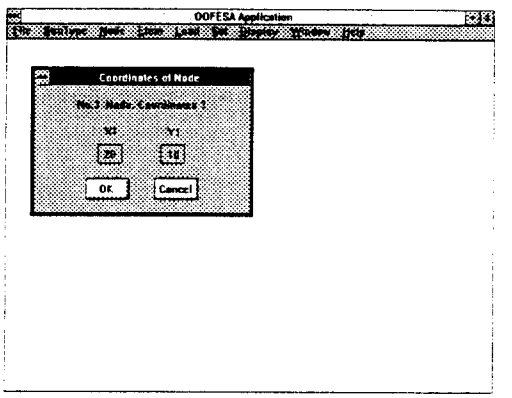


Fig.14 Dialog-box for for individual node input

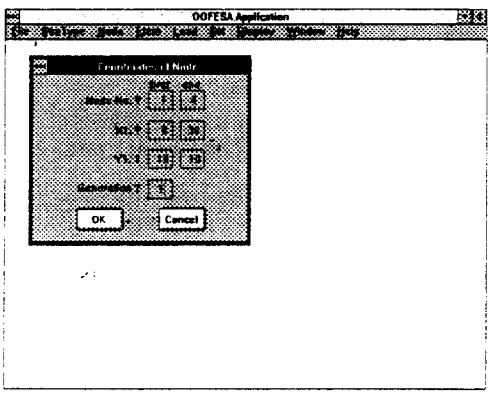


Fig.15 Dialog-box for node generation input

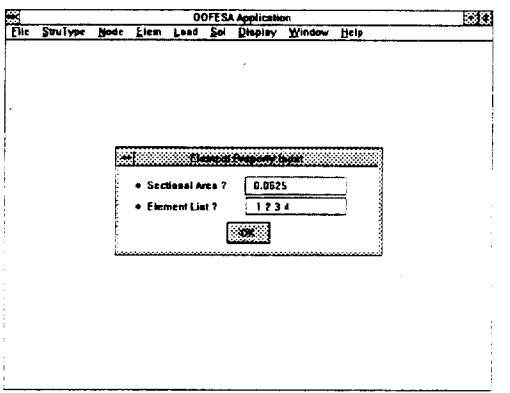


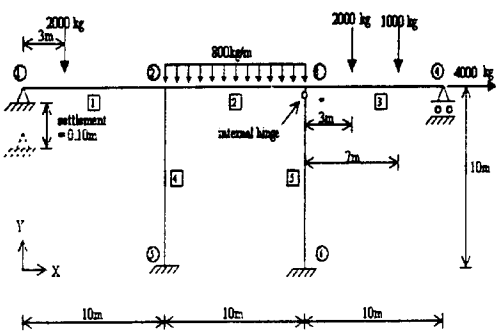
Fig.16 Dialog-box for material property input

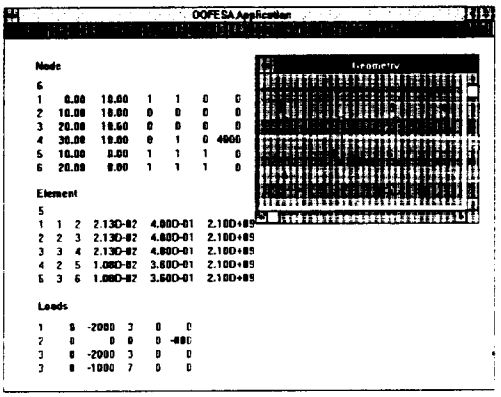
4. 예제해석

4.1 평면 뼈대구조의 정적해석

다음과 같은 평면 뼈대구조에 대하여 정적 해석을 수행하였다. Fig.17은 예제의 기하형상과 자료에 따라 입력된 화면을 보여주며, Fig.18은 윈도우 입출력 결과를 동시에 화면에 나타낸 것이다.

(1) 기하형상 및 입력자료





$E=2.1\times10^{9}kg/m^2$

Beams : $A=0.40m^2$, $I=0.0213m^4$

Columns : $A=0.36m^2$, $I=0.0108m^4$

Fig.17 Geometry and input screen of a plane frame example

(2) 해석결과

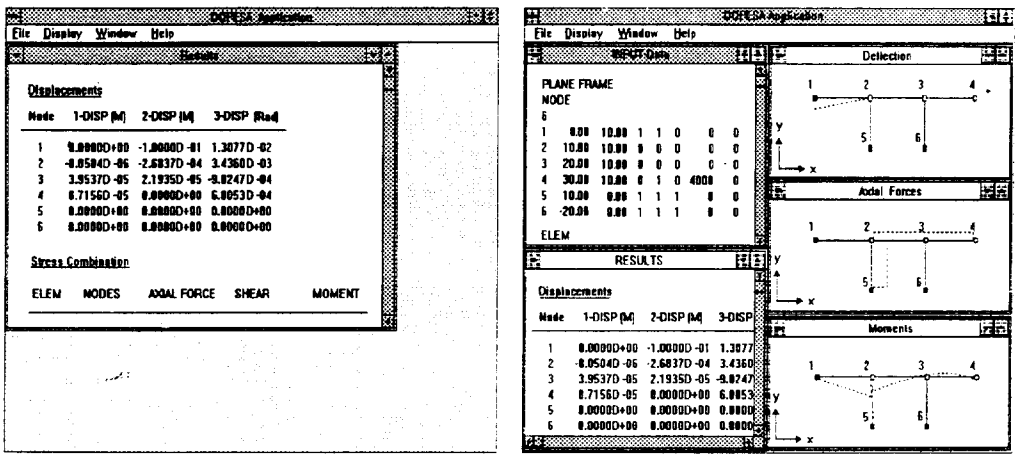
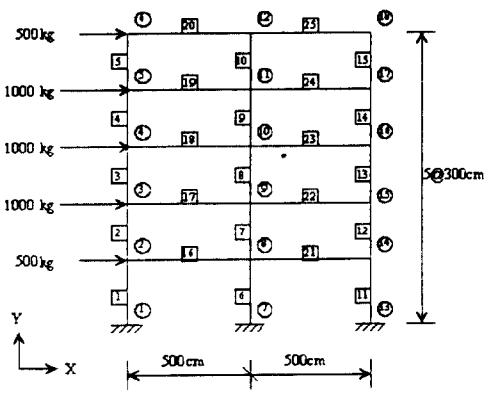


Fig.18 Text and graphic windows for input data and for result output of a plane frame

4.2 5층 2경간 빌딩구조의 정적해석

자유도가 54개인 다음과 같은 빌딩구조물을 평면 뼈대 구조로 해석하였다. 예제의 기하형상에서 '○'로 표기된 숫자는 절점 번호, '□'로 표기된 것은 요소 번호를 각각 가리킨다.

(1) 기하형상



$E=2.1\times10^5 kg/cm^2$, $A=1,250cm^2$, $I=2.6\times10^5 cm^4$

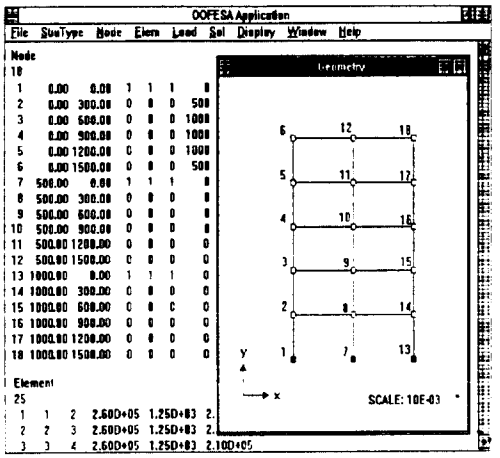


Fig.19 Geometry and input screen of a 2-bay, 5-story building frame example

(2) 해석결과

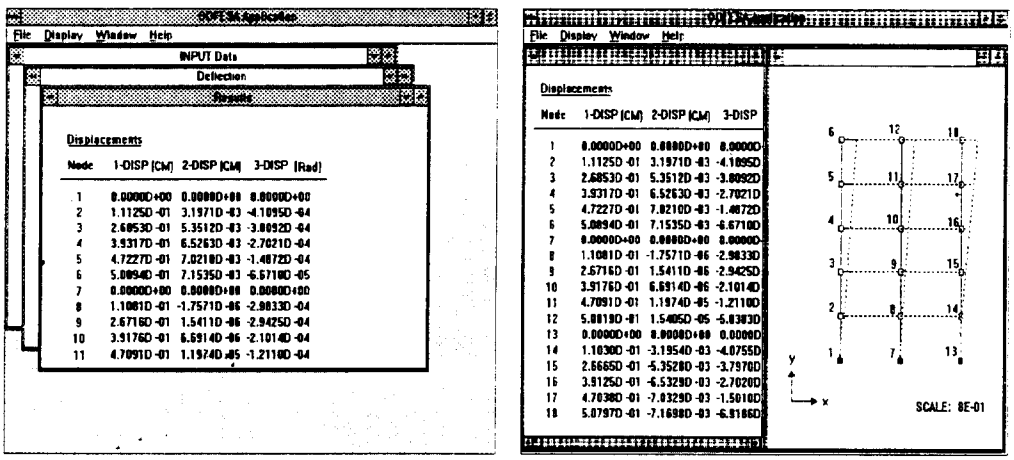
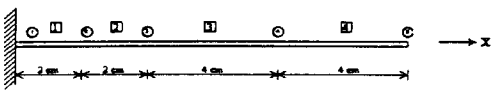


Fig.20 Text and graphic windows for input data and for re sult output of a building frame

4.3 캔티레버 보의 자유진동해석

다음과 같은 캔티레버 보의 자유진동해석을 수행한 결과 여러가지 자유진동 모드를 다중 윈도우를 통해 비교할 수 있으며, 각 모드별 고유진동수와 고유벡터들을 동시에 검증할 수 있다.

(1) 기하형상



$I=3.25\times10^{-4}cm^4$, $A=0.0625cm^2$

$E=30\times10^6N/cm^2$, $\rho=7.35\times10^{-4}N^2/cm^4$

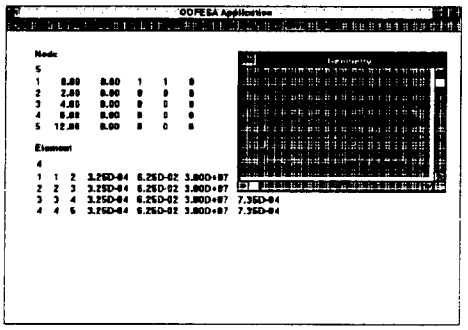


Fig.21 Geometry and input screen of a cantilever beam example

(2) 해석결과

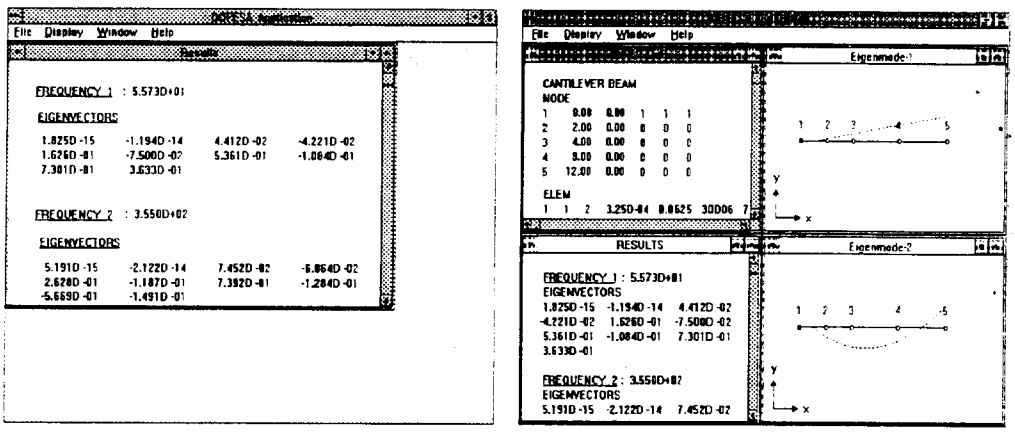
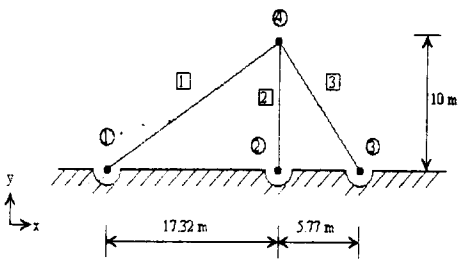


Fig.22 Text and graphic windows for input data and for result output of a cantilever beam

4.4 평면 트러스의 자유진동해석

(1) 기하형상



$A=0.0625m^2$, $E=30\times10^6N/m^2$, $\rho=7.35\times10^{-4}N^2/m^4$

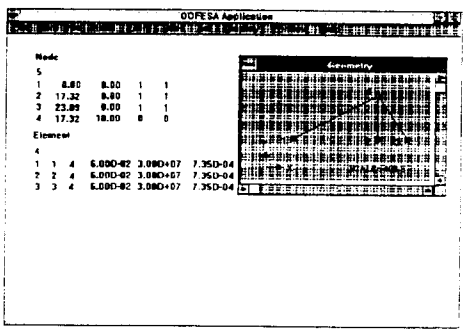


Fig.23 Geometry and input screen of a plane truss example

(2) 해석결과

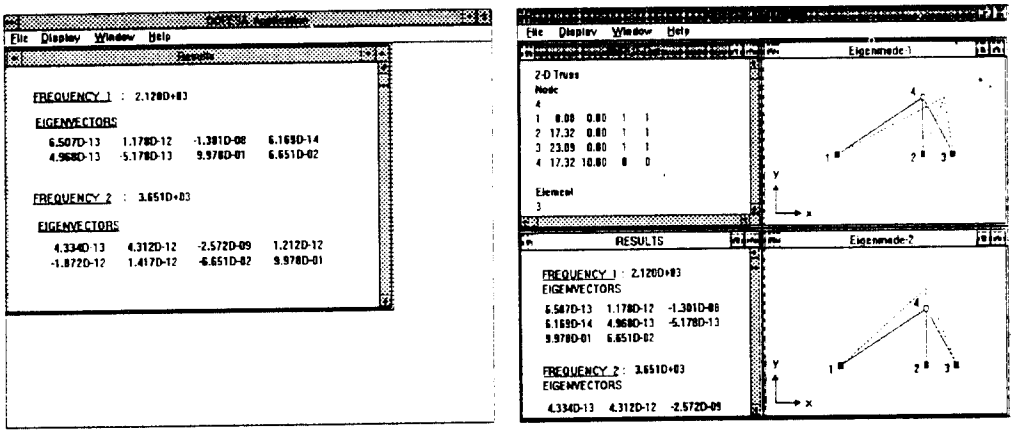


Fig.24 Text and graphic windows for input data and for result output of a plane truss

5. 결 론

본 논문에서는 C++언어를 사용한 객체지향 프로그래밍 기법으로 윈도우 환경에서 구동되는 유한요소 구조해석용 전후처리 프로그램을 개발하여 주프로그램인 OOFESA와의 통합운영 시스템에 관해 연구하였다. 이러한 통합운영 시스템은 GUI구현으로 사용자가 입력정보를 인식하는데 용이하고, 윈도우 환경의 사용으로 멀티태스킹과 동시성을 구현하며, 객체지향적 특성으로 프로그램 확장 및 수정시 전처리 메뉴의 구성이 편리하도록 설계하였다. 본 프로그램은 기존의 파일입력뿐만 아니라 사용자에게 친숙한 대화식 입출력 형태를 제공하며, 입출력 정보를 그래픽 처리하여 그 인식을 용이하게 하였고, 여러가지 문자정보와 그래픽 정보를 다중 윈도우를 통해 연계 운영할 수 있으며 구조해석을 위한 입출력에 적합한 메뉴를 구성하여, 입출력 윈도우 클래스, 그래픽 클래스 등의 라이브러리와 유한요소 구조해석 클래스들을 상호 연관시켜 전후처리 및 해석을 윈도우상에서 통합 운영할 수 있는 환경을 구현하였다. 해석결과가 다중 윈도우를 통하여 각 항목별로 표현됨으로써 기존의 프로그램에 비해 입출력 자료 인식이 현저하게 용이해졌고, 입력 오류의 발생도 감소시킬 수 있게 되었다. 본 프로그램에서는 객체지향적 기법의 적용 가능성에 중점을 두었기 때문에 이 기법의 핵심적인 특징인 확장성, 재사용성 등을 이용함과 동시에 수행속도, 저장방법 등에 좀 더 효율적인 알고리즘을 가진 유한요소 및 계산모듈들을 첨가한다면, 수행 효율성 면에서도 기존의 재래식 프로그램과 비교할만한 수준의 프로그램이 될 것으로 사료된다.

參 考 文 獻

1. Wiener, R.S. and Pinson, L.J., An Introduction to Object-Oriented Programming and C++, Addison-Wesley, MA(1989).
2. Microsoft WINDOWS 3.1, Guide to Programming, Microsoft Press, WA (1992).
3. Wilson, E.L. and Habibullah, A., SAP90: Computer Programs for the Static and Dynamic Finite Element Analysis of Structures, Computers & Structures Inc., Berkeley, CA (1989).
4. COSMOS/M, USER GUIDE, Structural Research and Analysis Corporation, CA (1986).
5. XETABS 1993; Three Dimensional Analysis of Building Systems, MICRO-ACE CLUB, Bangkok (1993).

6. Kanok-Nukulchai, W., Somporn, A and Sarun, U., MicroFEAP II P1-Module A module for static analysis of 2D truss, frame and shear wall structural systems, The MICRO-ACE CLUB, Bangkok (1987).
7. Kanok-Nukulchai, W. AIT 1993, MICRO-ACE CLUB, Bangkok (1993).
8. Stroustrup, B., The C++ Programming Language, Addison-Wesley, MA (1986).
9. 신영식, 서진국, 박영식, 최회욱, "PC용 객체지향 구조해석 프로그램의 개발", 한국전산구조공학회지, 제5권 제4호, pp.125-132 (1992).
10. 신영식, 서진국, "뼈대구조물의 자유진동해석을 위한 객체지향 C++ 프로그램", 대한토목학회 논문집, 제14권, 제1호, pp.119-129 (1994).
11. Borland C++ 3.1, Borland International Inc., CA (1992).

(접수일자 : 1994. 11. 30)