

유한요소 구조해석을 위한
객체지향 전처리 프로그램에 관한 연구
A Study on Object-Oriented Preprocessing Program
for Finite Element Structural Analysis

신 영식* 서 진국** 송 준엽*** 우 광성****
Shin, Young-Shik Suh, Jin-Kook Song, Joon-Yup Woo, Kwang-Sung

ABSTRACT

The pre-processor for finite element structural analysis considering the user-friendly device is developed by using GUI. This can be used on WINDOWS' environment which is realized the multi-tasking and the concurrency by object-oriented paradigm. Data input can be done easily through menu, dialog box, automatic stepwise input and concurrent representation with the structural geometry on multiple windows. It is designed to control integratedly the pre-processing, execution and the post-processing of the finite element structural analysis program on multiple windows, and input data can be seen with result outputs at the same time. In addition, the object-oriented programming environment makes convenient revision and addition of the program components for expanding the scope of analysis and making better user environment.

1. 서 론

유한요소법의 발전으로 복잡한 구조해석에 컴퓨터를 사용하게 됨에 따라 많은 범용 프로그램(package program)이 개발되었고, 지난 10여년 동안 개인용 컴퓨터(personal computer: 이하 PC)의 보급으로 PC용 프로그램들도 활발히 개발되고 있다. SAP90¹⁾, COSMOS/M²⁾, XETABS³⁾, MicroFEAP^{4),5)}, Micro-AT^{6),7)} 등이 기존의 널리 사용되는 PC용 프로그램들이라 할 수 있다. 이 프로그램중에는 정적해석만 가능하여 교육용 소프트웨어라고 할 수 있는 것도 있지만 동적해석을 포함한 다양한 해석범위를 가지는 것들도 있어 하드웨어의 프로세서(processor)의 발달로 인한 계산의 신속성 향상과 더불어 해석범위의 측면에서도 많은 진전이 있었다. 그러나 이와 같은 기존의 PC용 구조해석 프로그램들은 주로 매크로 명령어(macro-command)를 통한 파일 및 화면입력(Text User Interface: TUI)이 대부분이어서 사용자 환경이라는 측면에서는 문자(text)로만 주어지는 명령과 결과로 인해 정보의 인식 및 명령어 습득에 난점이 있는 등 사용자와 컴퓨터와의 유대가 결여되었었다. 이로 인해 지난 10여년간 대부분의 소프트웨어는 비대화식으로 부터 고도의 대화식으로 급속도로 발전되어 왔다. 사용자 대화 방식은 사용의 용이성, 정보의 표현력 등 여러가지 인간적 요소를 고려해서 구성되며, 그래픽 화면출력(display), 메뉴를 이용한 입력, 마우스(mouse)를 통한 선택 등 여러가지 대화식 유형을 말한다. 이러한 대화식 소프트웨어는 사용자 접속장치(user interface)를 응용 프로그램(application)으로부터의 독립적인 방법으로 사용하는 데 있어서 대화식 그래픽 접속장치 환경을 지원하기 위한 설계방법론과 개발환경인 객체지향적

* 영남대학교 공과대학 토목공학과 교수
** 영남대학교 공업기술연구소 연구원
*** 영남대학교 대학원 토목공학과 석사학위과정
**** 전남대학교 공과대학 토목공학과 조교수

설계에서의 재사용성(reusability), 캡슐화(encapsulation)의 원리와 신속한 원형제작(prototyping), 반복적 개발과 같은 기법들을 사용함으로써 사용자 접속장치를 함께 묶을수 있는 요소들로 다시 세분할 수 있고, 사용자는 대화식의 요구조건이 유사한 다른 응용 프로그램에서는 다시 사용될 수 없도록 신속하게 접속장치를 수정할 수 있으며, 응용 프로그램을 정의하는 코드(code)에 대해 어떠한 영향도 끼치지 않고 코드를 변경하거나 재위치시킬 수 있고, 반복적인 방법으로 접속장치를 개발할 수도 있는 등 몇가지 유리한 측면을 가지고 있다.

이와 같이 객체지향 프로그래밍 방식(object-oriented approach)⁸⁻¹⁰⁾은 자료 추상화(data abstraction)를 통한 오류수정의 용이성과 시스템 설계시 객체(object)와 클래스(class)별 분산으로 신속한 원형제작을 가능하게 하는 등 점진적(incremental) 프로그래밍 방식을 가지며, 상속성을 이용한 반복적 과정의 특성을 가진다. 또한 사용자 접속장치 생성 시스템에서 흔히 사용되는 입출력 루틴(routine)을 상속성(inheritance) 및 캡슐화의 개념으로 효과적으로 정의 가능케 하며, 표준화된 처리방법(method)들을 라이브러리(library) 형태로 미리 제공하여 이러한 상속성에 근거한 일련의 변형에 의해 프로그램 작성을 용이하게 하는 등 프로그래밍의 용이성과 향상된 기능의 접속장치를 지원한다.

사용자 접속장치 개발은 기본적인 그래픽 입출력 요소들을 조절하는 기능을 가지는 그래픽 패키지(graphic package), 그래픽 패키지를 확장시켜 하나 이상의 응용 프로그램 대화를 만들어 내고 관리할 수도 있게 하는 윈도우잉 시스템(windowing system), 프레임워크(frameworks), 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)¹¹⁻¹³⁾는 화면안에서 아이콘(icon)이나 윈도우(window)를 통해 프로그램을 수행하고, 화면 축소, 확대, 이동이 가능하며, 메뉴를 사용한 명령입력, 대화상자(dialogue box)를 이용한 자료입력을 통하여 현재상태나 명령이 화면에 그림으로 표시되어 사용자가 이러한 내용을 기억할 필요 없이 PC를 사용할 수 있게 한다. 그 중 윈도우즈 응용 프로그램은 모든 응용 프로그램이 동일한 윈도우즈 메카니즘을 이용하여 풀다운(pull-down) 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 다중처리(multi-tasking) 기능을 제공한다.

기존의 PC용 구조해석 프로그램중 대화식 유형을 취하여 비교적 편리한 전처리(preprocessing)를 제공하는 것중 MicroFEAP은 화면을 통한 대화식 입출력으로 입력자료를 문자자료와 그래픽 자료로 파일화한다. 그러나 이 프로그램은 정적 해석만 가능하며, 각 부메뉴(sub-menu)가 각각 독립적인 화면에 나타나는 단점이 있다. 또한 SAP90은 매크로 명령어를 사용하여 파일입력을 하거나, 전처리기인 SAPIN의 대화상자를 통하여 자료를 개별 입력하고 구조물의 기하적 형태를 그래픽 화면을 통해 동시에 표현하는 등 GUI를 이용한 좀 더 진보적인 입력장치를 사용하고 있으나, 단일화면을 사용함으로써 문자자료와 그래픽자료를 동시에 비교하여 인식할 수 없으며, 전처리, 실행 및 후처리(postprocessing)가 별도로 운영되고 있어 아직 사용자 측면에서 개선, 개발해야 할 점이 많다고 할 수 있다. 이에 반해 본 프로그램은 다중 윈도우즈를 이용하여 입력자료, 구조물의 기하적 형태, 문자와 그래픽 정보 등 여러가지 정보를 입력과정중 동시에 확인가능하게 하여 입력중 오류를 방지하며, 이러한 전처리기와 함께 실행 및 후처리를 윈도우 상에서 통합 운영함으로써 입력자료와 여러가지 출력결과를 동시에 비교가능케 함으로써 사용자 편의를 고려하는 좀 더 편리한 유한요소 구조해석용 전처리기를 GUI를 사용하여 개발하고자 한다.

2. 객체지향적 전처리기의 개발환경

2.1 사용언어

최초의 GUI자체가 객체지향 언어인 Smalltalk¹⁴⁾로 개발된 것이기 때문에 객체지향 언어로 응용 프로그램을 작성하는 것이 좋으며, 특히 C++언어^{15),16)}는 객체지향적 특성과 윈도우 프로그래밍 양쪽 모두에 적합한 언어라고 알려져 있다. C 기반언어인 C++는 Simula, Smalltalk 등과 달리 복합언어(hybrid language)로서 간결(compact)하고 실행속도가 빠르고 이식성이 강하며 어셈블러(assembler)와 연관하여 미세한 제어를 할 수 있는 등 기존의 C언어의 모든 기능들을 그대로 사용할 수 있으며, 동시에 자료 추상화, 연산자 중복, 다중상속 지원 등 객체지향적 개념들을 지원

하고 있다.

2.2 컴파일러(compiler) 및 프로그램 구동환경

본 프로그램의 컴파일러는 ObjectWindows¹⁷⁾, TurboVision 등과 같은 CASE 툴¹⁸⁾을 갖추고 있고, 윈도우용 프로그램을 완성하기 전에 먼저 수행해 봄으로써 오류를 찾아 프로그램을 수정할 수 있는 디버깅 툴인 WinSpector가 제공되며, 또한 윈도우에서 사용되는 단축키(accelerators), 그림(bitmap), 커서(cursor), 대화상자, 글자체(font), 아이콘, 메뉴, 문자열(string table) 등을 쉽게 설계, 편집할 수 있는 리소스 워크샵(Resource Workshop)을 제공하는 등 객체지향적 특성과 많은 클래스 라이브러리를 가지고 있는 볼랜드 C++ 3.1¹⁹⁾을 사용하였다.

본 프로그램은 IBM호환기종인 2MB이상의 메모리를 가진 386급 이상의 하드웨어와 WINDOWS 3.1²⁰⁾ 환경하에서 구동된다.

2.3 윈도우용 클래스 라이브러리(ObjectWindows Library; OWL)

Object는 ObjectWindows의 루트(root) 클래스이기 때문에 ObjectWindows에 존재하는 클래스중에서 TStreamable을 제외한 모든 클래스들은 Object의 파생 클래스로 존재한다. 그런데 Object는 추상 클래스이기 때문에 윈도우 프로그램에서 직접 Object의 인스턴스(instance)를 정의하여 사용할 수는 없다.

클래스 TModule은 윈도우 DLL(Dynamic Linking Library)로 작성되는 것들이 바로 프로그램에서 클래스 TModule의 인스턴스로 정의되어 있다. TModule은 주 윈도우(main window)를 생성하는 작업과 사용자로부터 전달되는 메시지(message)를 처리하는 작업 등과 같이 윈도우 프로그램 인스턴스를 초기화 시키는 작업을 수행하는 클래스인 TApplication의 기반 클래스로 존재하고 있는데, 클래스 TModule에서는 윈도우 DLL과 윈도우 프로그램에서 공통적으로 사용되는 정보들이 정의되어 있다.

클래스 TWindowsObject는 윈도우 프로그램에서 사용되는 모든 그래픽 자원의 기반 클래스로 동작한다.

TDialog는 대화상자를 표현하기 위해 제공되는 클래스이다. 윈도우 프로그램에서 대화상자는 정형 다이얼로그(dialogue)와 비정형 다이얼로그 등 두 가지로 구분되는데, 정형 다이얼로그를 생성하기 위해서는 클래스 TModule에 정의되어 있는 멤버 함수(member function) ExecDialog()를 사용하게 되고, 비정형 다이얼로그를 생성하기 위해서는 MakeWindow()를 사용한다.

TWindows는 윈도우 프로그램에서 사용되는 윈도우를 표현하기 위하여 제공되는 클래스인데, Fig.1에서와 같이 파생 클래스인 TMDIClient, TMDIFrame, TEditWindow, TControl을 가지고 있다.

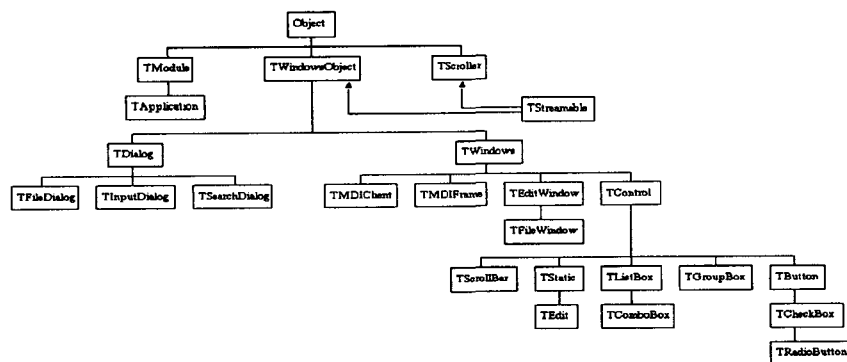


Fig.1. Hierarchy of Components of Input Window Classes

3. 객체지향 유한요소 구조해석용 전처리기의 구성

3.1 설계개요

입력은 하나의 프로그램만 받게 되지만 여러 개의 프로그램이 CPU를 나누어 쓰면서 동시에 여러 개의 응용 프로그램이 수행되는 선취형 다중처리(preemptive multi-tasking)를 구현한다. 여기서 입력을 받지 않고 주어진 프로그램을 수행하는 후면 처리(background processing)는 입력 그래픽 윈도우(input graphic window)가 되며, 입력이 이루어지는 전면 처리(foreground processing)는 입력 텍스트 윈도우(text input window)가 된다. 이 두 윈도우들은 객체지향적 특성인 객체 연결 및 결합기법(Object Linking and Embedding; OLE)을 통하여 두 개의 독립적인 응용 프로그램이 메시지 전달(message passing)에 의해 자료를 교환한다. 이를 위해 공통점이 있는 클래스들에 대한 기반 클래스(base class)를 만들어 놓고 여러가지 방향으로 진화시킴으로써 다양한 클래스를 새로이 만들어 낼 수 있다. 이때 가상함수의 개념은 기존 클래스에 있었던 기능중 일부를 개선하는데 사용하게 된다. 이와 같이 C++프로그램에서 가장 중요한 클래스간의 계층적 구조를 정하고, 클래스들이 가지고 있는 멤버 함수에 대한 정확한 정의를 내린다.

3.2 객체지향 유한요소 구조해석 프로그램과 입출력 윈도우즈와의 상관관계

객체지향 유한요소 구조해석 프로그램인 OOFESA²¹⁻²⁴⁾는 평면 및 공간 뼈대구조의 정적 해석 및 자유진동해석이 가능하며, 격자 및 판구조 해석을 위한 클래스들도 현재 개발중이다. Fig.2는 이 OOFESA프로그램과 본 전처리기, 앞으로 개발될 후처리기와의 상관관계를 나타낸 것이며, Fig.3은 OOFESA의 계층구조를 나타낸 것이다.

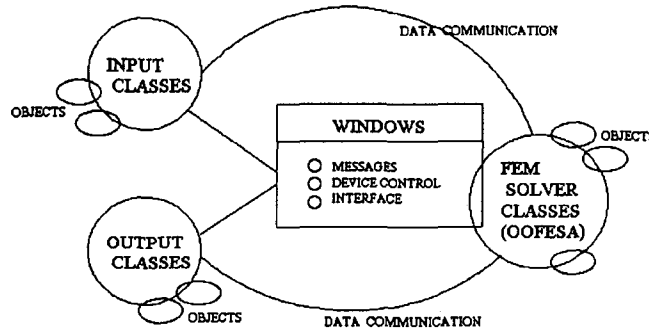


Fig.2. Interaction of Pre- & Post-processors and OOFESA Program

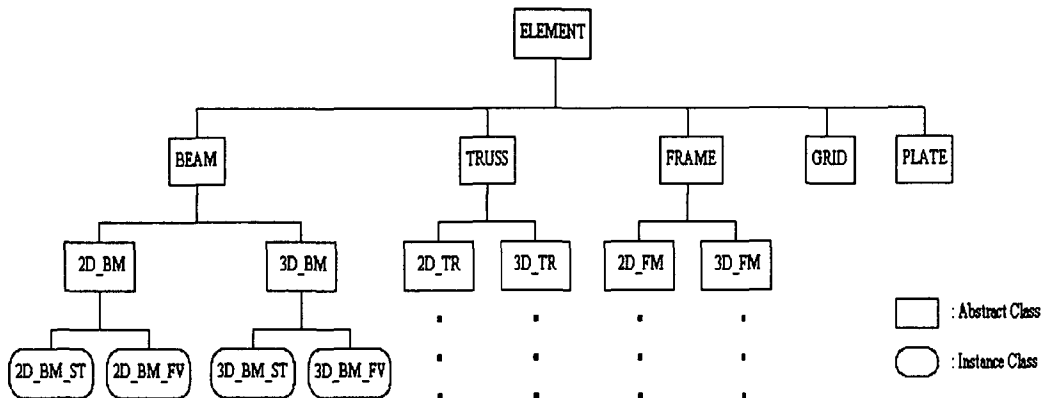


Fig.3. Hierarchy of ELEMENT Library Classes(OOFESA)

다음 Fig.4는 입출력 윈도우의 메뉴바(menu bar)에 나타나는 프로그램 메뉴와 각 메뉴별 부메

뉴에 대한 계층구조를 나타낸 것이다.

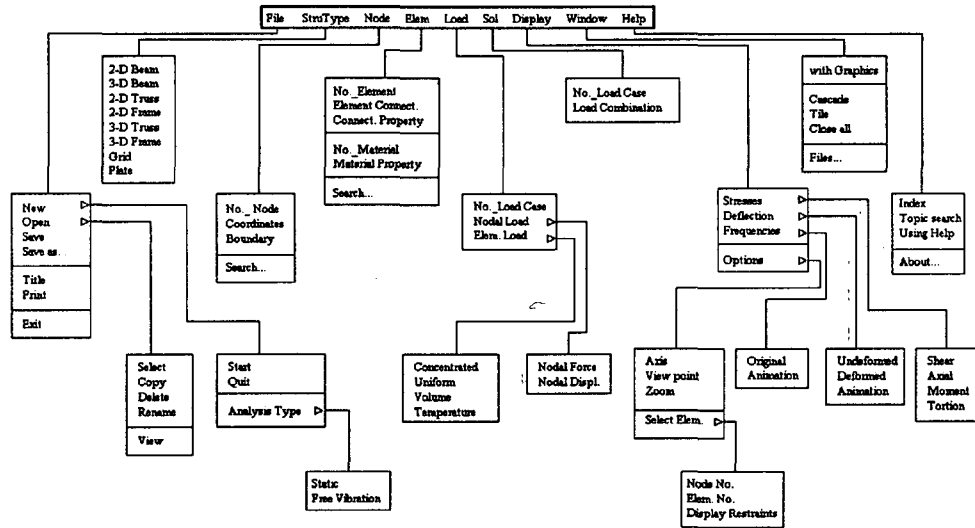


Fig.4. Architecture of Window's Menus for Input & Output

3.3 입력단계 및 기능

입력형식은 "Sequential Mode"와 "Editing Mode"로 나뉘는데, Sequential Mode는 윈도우의 메뉴바중 "File"메뉴의 "New"항목을 선택함으로써 도입되며, 이는 각 입력단계에서 대화상자를 통하여 자료를 입력하면 자동으로 다음 단계로 전개되는 새로운 프로젝트(project)를 생성하는 경우를 말한다. 이때 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 그래픽화면에 해당 좌표값이 동시에 그림으로 표시되는 것을 알 수 있다. Editing Mode란 기존의 프로젝트 파일을 열어 문서 편집기상에서 편집, 수정하거나 입력 윈도우의 해당 메뉴를 풀다운한 후 대화상자를 통해 수정할 수 있는 형식을 말한다.

Sequential Mode에서의 입력흐름에

- STEP 1: Select Starting Menu : New
- STEP 2: Select Analysis Type : Static
- STEP 3: Select 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

3.4 전처리의 객체, 클래스 정의 및 확장성

GUI객체들은 '이벤트(event)'라는 개념의 외부 입력에 대해서 동작한다. 프로그래머와 객체, 객체와 객체, 입력장치와 객체, 입력장치와 프로그래머 등의 사이에서 하고자 하는 일에 대한 의도(message)를 나타내는 것을 이벤트라고 할 수 있다. C++의 다형성(polymorphism)을 이용하여 서로 다른 윈도우들을 연결할 수 있도록 하며, 객체지향 프로그래밍 기법에서는 모든 윈도우들이 통합적으로 관리되어 윈도우의 수에 무관하게 외부 이벤트를 객체 내부에서 직접 처리하므로 외부 입력에 대한 윈도우들의 외부 처리가 가능하다. 메시지 루프(loop)와 이벤트에 의한 처리는

수 많은 윈도우들이 각각 자신에게 주어진 이벤트에 대해서 행동할 수 있도록 해 준다.

GUI에서 입력을 위한 비디오 화면은 다양한 그래픽 객체를 아이콘, 버튼, 스크롤바(scroll bar)와 같은 입력도구의 형태로 보여준다. 마우스 같은 지정도구(pointing device)를 사용하여, 사용자는 화면에서 이들 객체들을 직접 조작할 수 있다. 따라서 사용자와 프로그램 사이의 상호 작용은 좀 더 직관적이 되며, 정보의 일방적 흐름보다는 사용자는 화면에서 객체와 직접 상호 동작한다. 이와 같은 윈도우를 화면상에서의 객체로서 볼 수 있고, 이들 객체들과 버튼을 누르거나 스크롤바를 스크롤함으로써 직접 상호 작용할 수 있다. 항상 윈도우 클래스(window class)에 기반하여 만들어 지는 모든 윈도우는 그와 연관된 윈도우 처리방식(procedure)을 가지고 있으며, 메시지를 윈도우에 보내는 이 윈도우 처리방식은 프로그램 그 자체 또는 DLL에 있는 함수이다. 또한 윈도우 클래스는 동일한 윈도우 클래스에 기반한 여러 개의 윈도우를 허용하여 동일한 윈도우 처리방식을 사용하게 된다.

3.5 입력에

다음 Fig. 5, 6은 입력 윈도우의 초기화면과 부메뉴 등을 보여주며, Fig. 7의 윈도우들은 대화상자를 통해 입력된 자료들이 입력 윈도우(child window)상에 저장되어 축적되는 과정과 문자자료 파일 및 그래픽자료 파일이 생성되는 과정을 사용자가 동시에 인식할 수 있음을 보여준다.

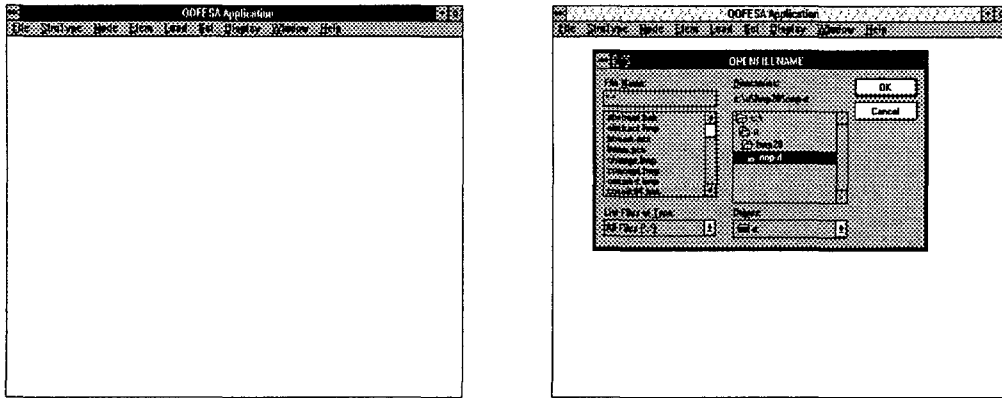


Fig.5. Initial and File Managing Demo Screen for Input

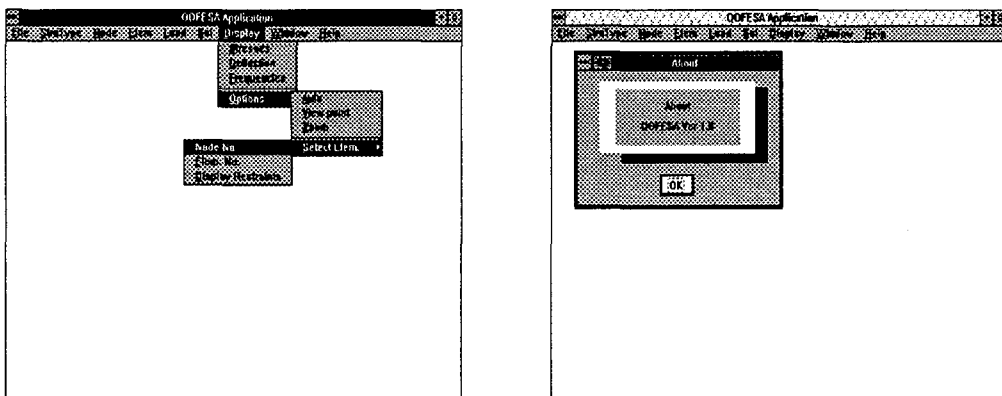
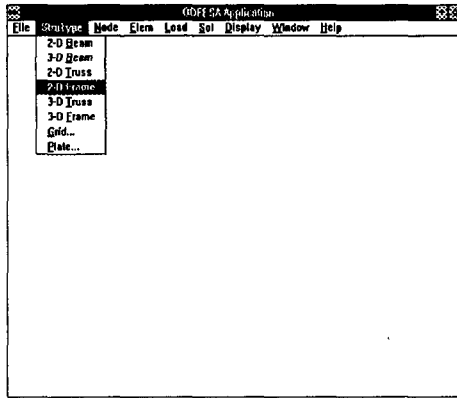
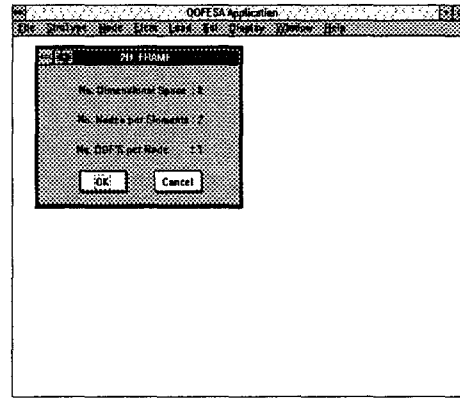


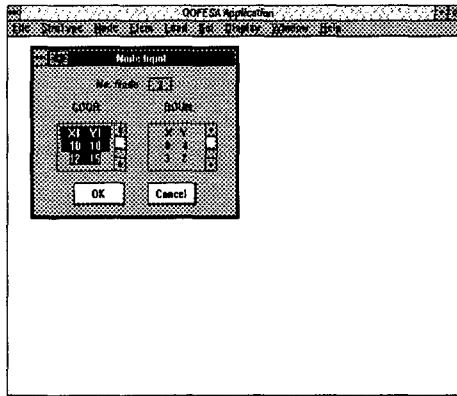
Fig.6. Sub-menu of Display and Help Screen



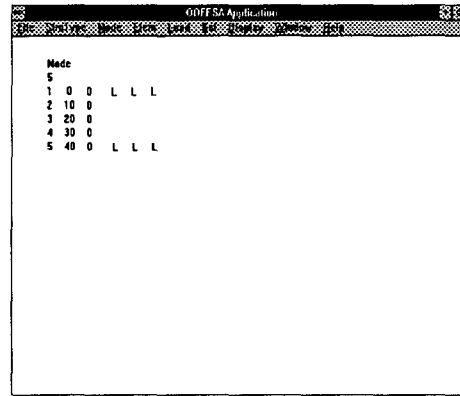
(a) Selecting Structural Type



(b) Starting Data Input



(c) Input Various Data
for Structural Analysis



(d) Data File Generating
by Stepwise Input

Fig.7. Demo Screens for Every Step of Sequential Input Mode

4. 결론 및 앞으로의 연구과제

GUI구현으로 사용자가 입력정보를 인식하는데 용이하고, 윈도우 환경의 사용으로 멀티태스킹과 동시성(concurrency)을 구현하였으며, 객체지향적 특성으로 프로그램 확장 및 수정시 전체리 메뉴의 구성이 용이할 것이다. 나아가 구조해석용뿐만 아니라 통합 설계 시스템 구축시에도 적용이 가능하리라 본다. 또한 윈도우상에서 실행 및 후처리를 통합 관리할 수 있게 하며, 다양한 그래픽 표현기법을 응용한 후처리를 설계하고, 확장성을 이용하여 구조해석 범위를 점차 넓혀 가야 할 것이다.

5. 참고문헌

- 1) 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
- 2) *COSMOS/M, USER GUIDE*, Structural Research and Analysis Corporation, CA, 1986

- 3) *XETABS 1993; Three Dimensional Analysis of Building Systems*, MICRO-ACE Club, Bangkok, 1993
- 4) 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, AIT, 1987
- 5) Kanok-Nukulchai, W., Somporn, A. and Sarun, U., *MicroFEAP II P4-Module A module for static analysis of space truss, frame and structural systems*, The MICRO-ACE CLUB, AIT, 1987
- 6) Chung, C.F., "Development of an Extended Databased Analysis Interpretive Treatise for Micro/Mainframe Computers.(Micro-AIT)", *M.S.Thesis*, AIT, Bangkok, Thailand, 1987
- 7) Kanok-Nukulchai, W., *AIT 1993*, MICRO-ACE Club, Bangkok, 1993
- 8) Wiener, R.S. and Pinson, L.J., *An Introduction to Object-Oriented Programming and C++*, Addison-Wesley, MA, 1989
- 9) Pinson, L.J. and Wiener, R.S., *Applications of Object-Oriented Programming*, Addison-Wesley, MA, 1990
- 10) Pohl, I., *Object-Oriented Programming Using C++*, The Benjamin/Cummings Co., Redwood City, 1993
- 11) Ezzell, B., *Graphics Programming in Turbo C++*, Addison-Wesley, MA, 1990
- 12) Faison, E. W., Jr., *Graphical User Interfaces with Turbo C++*, SAMS, Carmel, 1991
- 13) Goodwin, M., *User Interfaces In C and C++*, MIS press, New York, 1992
- 14) Goldberg, A. and Robson, D., *Smalltalk-80, The Implementation*, Addison-Wesley, MA, 1983
- 15) Stroustrup, B., *The C++ Programming Language*, Addison-Wesley, MA, 1986
- 16) Radd, S.R., *C++ Techniques & Applications*, Prentice-Hall, Redwood City, 1990
- 17) Shammass, N.C., *Windows Programmer's Guide to OBJECTWINDOWS LIBRARY*, SAMS, 1992
- 18) McCord, J., *Windows Programmer's Guide to Borland C++ Tools*, SAMS, 1992
- 19) *Borland C++ 3.1*, Borland International Inc., CA., 1992
- 20) *Microsoft WINDOWS 3.1, Guide to Programming*, Microsoft Press, WA., 1992
- 21) 신영식, 고영배, "PC용 객체지향 매트릭스 구조해석 프로그램의 개발", 영남대학교 공업기술 연구소 연구보고, 18권 2호, 1990, pp.11-19
- 22) 신영식, 서진국, 박영식, 최희욱, "PC용 객체지향 구조해석 프로그램의 개발", 한국전산구조 공학회지, 제5권 제4호, 1992, pp.125-132.
- 23) Shin, Y.S., "Object Oriented Free Vibration Analysis of Structures", *Proceedings of EASEC-4, Fourth East-Asia and Pacific Conference on Structural Engineering and Construction*, Seoul, Vol.3, 1993, pp.1809-1814.
- 24) 신영식, 서진국, "뼈대구조물의 자유진동해석을 위한 객체지향 C++ 프로그램", 대한토목학회 논문집, 제14권 제1호, 1994, pp.119-129