建築構造解析을 위한 先後處理 프로그램의 開發

Developing A Pre-and Post-Procellor for Building Analysis

李 政 宰*
Lee, Jeong Jae

Summary

General concepts and overall procedures of interactive graphical user interface, a pre- and post- processor, for building analysis are introduced.
Attention is forcused on the data structures and the modeling operators which can ensure the intergrity of its database should have. An example of model building process is presented to illustrate its capability, its facilities for modifying, and for processing.

I. 緖 論

現代의 建築設計는 재료의 발달과 工法의 改善에 힘입어 모양이 다양해지고 복잡해지는 한편, 餘裕部材가 많은 高次不定靜 構造로 되고 있어, 全體 建物을 一體構造로 解析하는 有限要素 解析을 바탕으로 한 設計가 보편화 하고 있으나, 복잡한 構造物을 사실에 가깝도록 描寫하며, 有限要素 모델화 하고, 分析結果를 쉽게 判別하는데는 어려움이 있고, 특히 建築構造物을 動的解析한 경우, 시간에 따른 건물의 변화를 다른 도움없이 분석하는 것은 사실상 불가능에 가깝다고 한다[1]. 이런 문제점을 해결하기 위해 有限要素 解釋에는 資料의 입출력과 결과분석을 손쉽게 하는 先後處理 프로그램이 많이 이용되고 있다. 이와같은 先後處理 프로그램들은 제작자의 의도에 따라, 사용목적에 따라 서로 다른 資料構造와 시스템 구성을 가지게 된다.

建物은 기둥과 보 및 벽체를 조립하여 외부환경과 격리 또는 차단되는 공간을 유지하는데 그 목적이 있다. 따라서, 기둥과 벽체 및 보의 相互作用의 묘사는 건물의 構造解析에 중요한 변수가 된다. Srivastav[2]는 같은 建築物의 解

*(財)韓國農地開發研究所

키워드 : 先後處理 프로그램, 建築構造 解析, Boundary Modeling, Object Oriented Programming, Hierachical Data Structure, Radial Edge Data Structure, Modeling Operator, Euler Operator.

析에 있어서, 보와 슬래브의 결합상태에 따라, 동하중 상태에서 최대변위가 2배까지 다를 수 있음을 보인 바 있다.

본 論文에서는 筆者가 參與하여 작성한 建築構造解析 용 先後處理 프로그램 BASYS(Building Analysis System)를 中心으로 建物描寫에 필요한 유연성을 확보할 수 있는 資料構造, 先後處理 프로그램이 갖추어야 할 一般的인 기능 및 프로그램 작성시 有意點 등을 考察하고, 先後處理 프로그램의 운영예를 보이고자 한다.

Ⅱ. 先後處理 프로그램의 構成

1. 運營環境

Perquera[3]는 先後處理 프로그램이 갖추어야 할 一般的인 條件을 ① 使用者가 한가지 목적을 달성하는데 여러가지 기능을 이용할 수 있는 柔軟性, ② 한 作業이 잘못된 경우 그 복원의 容易性, ③ 컴퓨터의 고장등에 대처한 對備能力, ④ 작업의 진행과정을 使用者가 파악할 수 있도록 하는 明快性, ⑤ 작업상태를 계속 보여줄 수 있는 透明性, ⑥ 표시되는 영상의 簡潔性과 辨別能力, ⑦ 使用者를 위한 指針, ⑧ 기계간의 容易한 移轉能力으로 정의하였다. 이와같은 條件을 만족시키기 위해서는 우선, 대상 하드웨어와 운영환경을 적절히 考慮해야 될 것이다. 근래에 들어 DOS를 運營體制로 한 PC는 處理能力과 기계간의 호환성만을 고려할 경우, 先後處理 프로그램 개발의 基本環境이 될 수 있으나 實메모리(Real Memory)를 위주로 하므로 구성 모델에 한계가 있으며, 先後處理의 기본이 될 수 있는 Graphic Tool이 확정되지 않고 있다. 先後處理 프로그램은 使用者와 컴퓨터 간에 대화식으로 이용되고, 모델구성에 오랜 시간이 소요 되므로, 컴퓨터의 처리능력 이상으로 안정성과 높은 해상도 등이 필요하다. 또 有限要素 解析은 解析해야 할 대상이 尨大한 경우, 先後處理는 작은 기계에서 처리하고, 本 處理는 용량이 크고 빠른 기계를 이용할 수 있어야 하므로, 네트워크를 이용하기 용이 하여야 한다. X Window는 이와 같은 네트워크 환경을 전제로 하고 있으며, 다양한 함수들이 이미 개발되어 있어 使用者 중심의 프로그래밍을 용이하게 하고, Workstation의 산업표준 Graphic Tool로 이해되고 있으므로 기계간의 호환성을 높일 수 있다. UNIX는 가상메모리(Virtual Memory)를 이용하고 있으므로 構造모델의 크기에 큰 제한을 받지 않고, 대용량의 보조기억장치를 이용할 수 있으며, 이미 구축된 모든 네트워크에 잘 접속되어 X Window가 필요로 하는 Server-Client의 분리를 원활히 하는 등, 장래의 先後處理 프로그램의 기본환경으로 정착될 것으로 보인다. 先後處理에서의 使用者와 컴퓨터의 대화는 使用者의 편의를 높이기 위하여 GUI(graphic User Interface) 방법을 이용하는 것이 효율적이며, 사용단계에 따라 다음 단계가 제시되는 Pull Down 메뉴식은 프로그램에 대한 특별한 이해를 필요로 하지 않고, 간단한 설명으로 使用者의 의문점을 해소할 수 있는 등 장점이 있어 많이 이용된다. X Window를 바탕으로 한 GUI 기법을 이용한 프로그래밍은 근본적으로 C 또는 C^{++}언어를 前提로 하고 있다고 하여도 과언이 아니다. 이 두 언어를 이용함으로서 메모리의 관리와 컴퓨터 자원을 용이하게 이용하고 構造化 프로그래밍도 달성할 수 있다. 본 연구의 중심이 되는 BASYS는 C 언어로 작성되었다.

2. 主要構成

先後處理 프로그램은 解析하고자 하는 모델을 開發하고, 本 處理 프로그램의 解析에 적합하도록 入力資料를 작성하며, 處理結果를 알기 쉽게 표현하여야 하므로 ① 다른 프로그램과 資料의 교환이 가능한 입출력 기능, ② 모델을 구성하기 위한 資料의 획득, 표현, 조작 및 저장기능, ③ 本 處理에 필요한 資料의 생산

및 저장기능, ④ 後處理에 필요한 표현기능 등이 기본적으로 필요하다. 이후 各章에서는 이들 기능을 합리적이고 효율적으로 보장하는 資料構造, 프로그램의 模型, 프로그래밍 方法 및 操作函數 등에 관하여 BASYS를 중심으로 논하고자 한다.

Ⅲ. 客體指向 프로그래밍(Object Oriented Programming)

一般的으로 圖形的 表現을 이용하는 프로그램은 尨大해지기 쉽다. 이들 대규모 프로그램들은 反復的이고 유사한 프로그램들의 집합이 될 가능성이 크고, 프로그램이 진행될 수록, 또 프로그램의 規模가 커질수록 급속히 誤謬가 늘게 된다. 이런 손실을 최소화 하기 위하여 최근에 客體指向 프로그래밍 기법이 제안되었고, 이 방법을 채택함으로서 고도의 함수독립성을 달성하게 되어, 프로그램의 수명을 대폭 연장하고 오류도 줄일 수 있게 되었다. 先後處理 프로그램은 대개 그 규모가 크고, 개발에 많은 시간과 노력이 소요되며, 使用者의 요구나 필요에 따라 계속 프로그램을 확장, 보완해야 할 필요가 매우 크므로, 客體指向 프로그램과 같은 프로그래밍 기법의 도입이 절실하다. 客體指向 프로그래밍 기법의 기본개념은 資料(Data)와 資料의 가공방법(Method)를 동시에 한 客體로 간주하는 Encapsulation, 같은 資料型을 이용하는 경우 그 기능이 상속되는 Inheritance 및 같은 함수라도 入力資料에 의해 서로 다르게 작용하는 Polymorphism으로 요약된다. Fig. 1은 BASYS에 구현된 한 客體로서, C 언어를 이용하여 有限要素解析의 Node를 하나의 資料型으로 선언하여 客體指向 프로그래밍을 달성하는 한 예를 도시한 것이다. 이 예에서 Node는 資料로서 절점의 ID, Type, Coordinates를 가지고 이 資料를 이용하고, 관리하는 방법으로 좌표값을 되돌려 주는 Get-Coord, 좌표값을

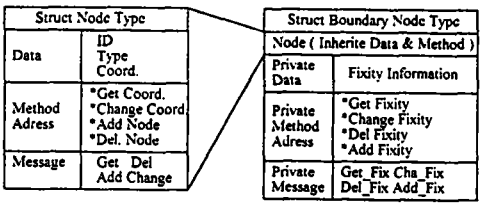

Fig. 1. A Example of Data Structure for Object Orient Programming With C

변경하는 Change-Coord, 새로운 절점을 추가하는 Add-Node 및 절점을 삭제하는 Del-Node 등 방법을 포인터 값으로 변형하여 내장한 客體를 구성함으로서 Encapsulation을 달성하고 있다. Boundary-Node는 이 Node형 변수를 선언함으로서 절점의 자료인 ID, Type 및 Coordinates와 방법들을 그대로 상속하고 Boundary-Node만이 가지는 절점의 고정여부(Fixity)에 관한 資料를 추가하며 이 고정여부 資料를 이용하는 방법들 즉, Get-Fixity, Add-Fixity만 정의하면 된다. 이 경우 Node part는 Boundary-Node part의 Super class가 된다. 또 절점의 좌표를 얻고자 하는 경우, Node, Method(Get, ID)와 같이 Message로 Get와 매개변수로 ID를 전달함으로서 Get-Coord 함수를 이용하고, 새로운 절점을 추가하는 경우 Node, Method (Add, x, y, z)와 같이 Message인 Add와, 매개변수인 x, y, z값을 전달함으로서 Add-Node 함수를 이용할 수 있는데, 이는 C 언어의 Overloading 기능을 이용하여 Polymorphism 현상을 구현한 좋은 예라 할 수 있다.

Ⅳ. 階層化 모델링

先後處理 프로그램은 本處理 프로그램을 손쉽게 이용할 수 있도록 하는데 그 목적이 있으므로, 本處理 프로그램의 성격에 맞추어 설계되고 작성된다. 先後處理 프로그램은 先後處理와 구별되는 本處理(Process)가 있으므로,

先後處理 중의 상태와 本處理 중의 상태가 분리될 수 있음을 암시한다. 建築構造에서는 여러 가지 부재들이 동일한 규격, 재질인 경우가 많고, 構造解析을 위해서는 많은 要素로 분할할 필요가 있지만, 건물을 설계하는 과정에서는 要素를 분할할 필요가 없는 것이 보통이다. 만약, 本處理를 염두에 두고 설계과정에서부터 많은 要素로 분할하여 모델을 구축한다면, 要素의 수가 많은 만큼 부수처리가 많아지고, 도형要素와 함께 屬性資料를 유지해야 하므로 비능률적이 된다. 이와 같은 비능률을 줄이고, 작은 컴퓨터에서도 빠르게 운용될 수 있는 프로그램을 구성하기 위해서는, 설계과정 중의 資料構造와 분석과정의 資料構造를 분리할 필요가 있다. BASYS는 이들을 모델의 구축과정(STR : Structural Level)과 要素의 분할 과정(SDV : Subdivade Level) 및 Mesh 생산과정(MSH : Mesh Level)로 분리하여 그 효율성을 높이고 있다.

1. STR-Level

設計過程에 해당하며, 모델의 규모와 형태, 제반 재질과 규격 및 경계조건을 확정한다. 이 과정에서는 모든 부재가 가장 적은 수의 要素로 분할되어 있게 함으로서 메모리를 줄이고, 프로그램의 속도를 올릴 수 있다. STR Level에서 정해진 모든 속성들을 하위계층인 SDV, MSH Level이 이어받게 하면, 資料의 중복도 피할 수 있고 설계오류도 방지할 수 있다. 실제 프로그램에서는 공통적인 屬性資料들을 참조자료(Lookup Tables)로 만들고, 같은 STR Level의 하위요소(SDV, MSH 要素)는 같은 참조자료를 이용함으로서 이를 구현한다.

2. SDV-Level

STR Level에서 구축된 모델의 部材는 構造解析에 용이하도록 要素로 구분하는 과정이다. 따라서, 예상되는 應力의 集中地域이나 필요한 곳을 좀 더 세밀하게 분할하여 검토할 수 있도록 要素의 크기를 조절한다. 이 과정은 모든 모델을 wire Frame 모델로 간주함으로서 효율을 높일 수 있다.

3. MSH Level

SDV Level의 분할을 기초로, 要素의 형태에 따라 메쉬를 생성하는 과정이다. 모델의 構築過程에서는 최소한의 圖形要素를 이용하는 것이 유리하므로, 모든 Edge는 2절점으로, 모든 Face는 4절점으로 이루어진 것으로 간주한다. 따라서, 이 과정에서는 Face 要素가 있을 경우에만 필요한 절점을 추가한다. 만약 使用者가 모델의 구성을 완료하고 本處理를 위하여 入力資料를 생성하는 경우, 使用者가 요망하는 要素에 부합하는 새로운 절점을 추가하도록 하면 매우 효율적이 된다. 入力資料 생성과정에 메모리를 최소화하는 알고리듬이나, 자동 메쉬 알고리듬 등을 구현함으로서 프로그램의 유연성을 높일 수 있다. Fig. 2는 9 Node Lagrangian Shell 要素와 Quadratic Beam 要素를 이용한 경우로, STR, SDV, MSH Level의 Node 수가 각각 9, 21, 23개인데 반하여, 실제 解析 과정에서는 63개로, 설계과정과 入力資料 생성과정이 분리되어야 하는 타당성을 보여주고 있다.

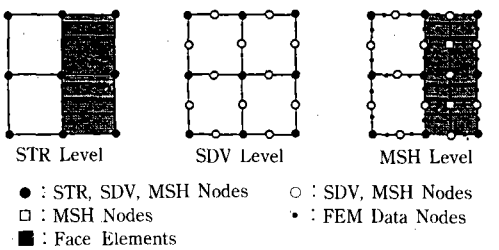

Fig. 2. Hierachy of Models for a 2D Framed Structure

V. 圖形要素와 데이터베이스

有限要素 解析에 필요한 先後處理 프로그램은 Solid 모델링 기법을 이용하지만 무게, 밀도,

크게, 강도 및 상호관계를 표시할 수 있도록 제작되어야 한다. 즉, 先後處理 프로그램에서는 모든 要素가 길이, 두께에 상관없이 각각 서로 다른 資料를 가지고 있게 된다. 이와 같은 先後處理의 특성을 나타내기 위해서는, 모델의 형태를 보여주는데 이용되는 圖形要素와, 圖形要素가 가지는 성질들을 표현하는 屬性資料를 분리하여 정규화(Normalized)된 데이터베이스를 구성함으로서 데이터베이스의 무결성을 높일 필요가 있다. 데이터베이스는 채택하는 Solid 모델링 기법에 따라 그 資料型과 구성이 결정되는데, Zeid[4]는 Solid 모델링을 ① Half Spsac, ② Boundary 모델링, ③ Constructive Solid Geometry(CSG), ④ Sweeping, ⑤ Analytical Solid 모델링, ⑥ Spatial Enumerating, ⑦ Octree Encoding, ⑧ Gell Decomposion, ⑨ Primitive Instancing 등으로 분류하고 이들 중 CSG와 Boundary 모델링이 가장 많이 쓰인다고 하였다. BASYS는 Boundary 모델링 기법을 채택하고 있는데, Boundary 모델링은 Solid를 점과 선으로 구성되는 면에 의해 경계지어지는 위상요소로 간주하고, 이 면들의 방향-안(Inside)와 밖(Outside)-에 따라 결합상태를 결정지어 가며, 경계 이외의 곳은 서로 연결된 연속체로 가정한다. 따라서, 건물과 같이 선, 면의 집합을 묘사하는데 장점이 있으며, 모든 圖形要素들이 독립적으로 존재하므로 屬性資料를 연결하여 저장하기 용이하다. 모든 圖形要素 간의 관계를 엄밀히 정의하고, 저장하기 때문에 資料의 양이 많아지므로, 처리속도가 떨어지는 단점은 Double linked list 構造를 이용하여 극복하였다.

1. Boundary 모델링을 위한 圖形要素

임의의 Solid의 경계를 점과 선으로 묘사하기 위해서는 잘 정의된 要素가 필요하다. BASYS는 Radial Edge 資料構造를 채택하고 있는데 이 資料構造는 Baumgart가 발표한 "Wing Edge" 資料構造로부터 비롯된다.[5] Weiler는 이 Wing Edge 構造에 use의 개념을 도입함으로서 3次元 non-manifold 물체를 일반적으로 표시할 수 있게 하였는데, Edge를 기본요소로 하고 있으며, use는 要素의 사용 역할을 뜻하는 것으로서, 이를 이용하여 圖形要素간의 연결상태를 적절히 표현할 수 있다. 이 構造에서는 한 Edge가 여러 Face와 접할 수 있게 되므로, 그 단면형이 방사형으로 되어 "Radial Edge" 資料構造로 이름하게 되었다.

2. Redial Edge 資料構造

가. 基本要素

Redial Edge 資料構造에 이용되는 기본 圖形要素는 Model, Region, Shell, Face, Loop, Edge, Vertex로서 그 정의는 아래와 같으며, 모든 물체는 이들 기본요소의 집합으로 표현되고, 이들 要素는 서로 계층적 構造를 이루고 있다.

Model : 물체를 나타내며, 모든 要素들의 집합
Region : 3차원의 한 지역
Shell : Region의 면 경계요소로 Face의 집합
Face : Shell의 경계이며, Loop로 그 규모가 표시됨.
Loop : 면의 닫혀진 경계, 선들의 집합
Edge : 두 점간을 이은 선분
Vertex : 공간의 한 점

이들 요소들의 결합상태는 use要素로 표현되며 그 종류는 다음과 같다.

Faceuse : Face의 한쪽면으로 Shell의 내외부를 나타낼 수 있다.
Loopuse : Face를 구성하는 Edge의 순서와 Vertex의 순서를 나타낸다.
Edgeuse : Vertexuse와 함께 Edge의 방향을 나타낸다. 한 Edge는 동시에 여러 要素-Face, Shell, Re-

gion－의 하위계층 要素로 이용될 수 있다.

Vertexuse : 한 Vertex가 여러 개의 상위요소의 구성 인자가 될 경우 이 Vertex를 소유하는 상위요소를 나타낸다.

BASYS에 구현된 이들 要素들의 계층적 연결상태는 Fig. 3과 같다. 즉, 모든 圖形要素의 屬性資料는 기본요소가 가지고 있으며 use 要素들은 기본요소의 쓰임새와 연결상태만을 가지게 함으로서 資料의 유일성을 보장한다.

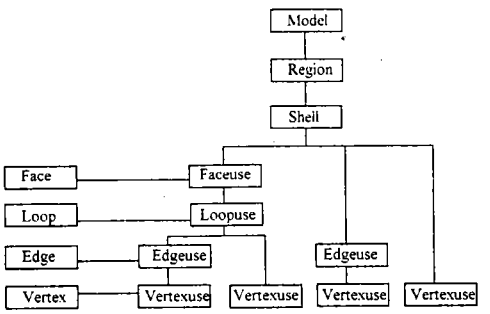

Fig. 3. A Schmatic of the Topological Entities of Radial Edge Structure.

Fig. 4는 한 Edge를 여러 개의 Face가 이용하는 경우, 그 연결상태를 표시하는 방법을 보여주고 있다. 즉, 동일한 Face의 앞뒤를 구성하는 방향이 반대인 두 개의 Edgeuse는 Mate Edgeuse로 불리며, 서로 다른 Face를 구성하는 이웃 Edgeuse는 Face는 이 Edge에 속하는 모든 Edgeuse를 모두 검색함으로서 알 수 있다.

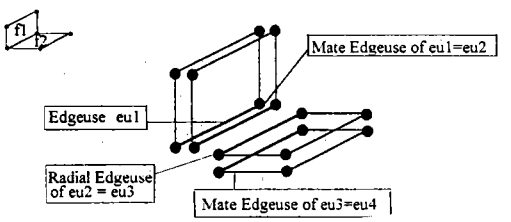

Fig. 4. The Relationship of Radial Edges and Mate Edges

나. Redial Edge 資料構成

Fig. 5는 Fig. 3 및 Fig. 4의 개념을 Doubly linked list 構造와 Union을 이용하여 계층적으로는 상하의 기본요소와 연결되고, 동일한 要素끼리의 수평적으로 연결되어 양방향으로 참조할 수 있도록 구현한 資料構造의 예이다.

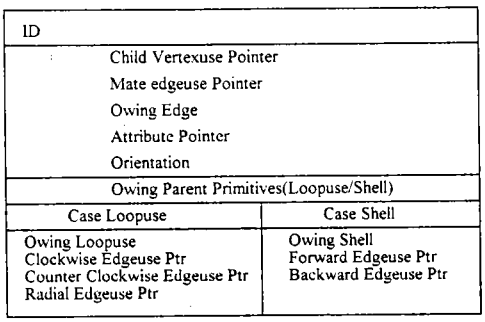

Fig. 5. An Example of Radial Edge Structure : Edgeuse Data Structure

3. 屬性資料 데이터베이스

先後처리 프로그램은 모든 圖形要素가 가지는 屬性資料로 구성된 데이터베이스를 가져야 한다. 이 屬性資料는 개개의 圖形要素를 통하여

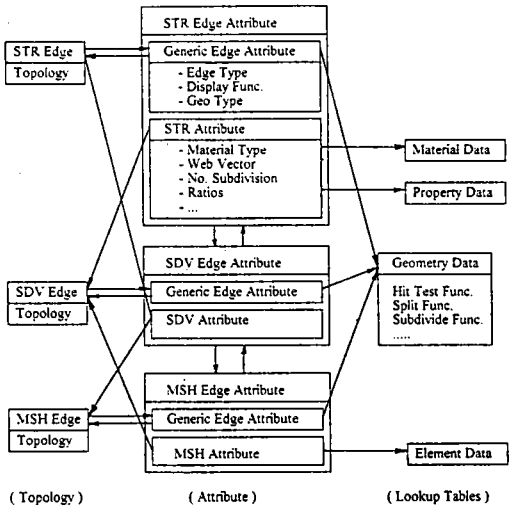

Fig. 6. Relationship Between Data Structures

참조할 수 있고, 圖形要素와 마찬가지로 자유로이 변경, 추가, 삭제가 가능하여야 하며, 모든 圖形要素는 하나의 屬性資料를 가지므로, 屬性資料와 圖形要素는 서로 양방향으로 참조가능 하여야 한다. Fig. 6은 BASYS에서의 資料構造간의 관계를 묘사한 것으로, 모든 Edge는 같은 資料와 方法을 쓰며, Generic Attribute는 서로 다른 계층간에 상속되는 등, 客體指向 프로그래밍 기법이 도입된 것을 알 수 있다.

VI. 資料의 管理

모델의 구성을 위한 資料의 추가, 검색, 변경 및 삭제 등 조작은 資料構造나 계층적 특성과는 관계없이 使用者에게는 투명(Transparency)하여야 하며, 使用者가 스스로의 의사에 따라 작업을 수행할 수 있도록 높은 유연성을 확보해야 한다. 이와 같은 유연성은 잘 정의된 조작함수(Operator)에 의해 이루어지는데, BASYS에서는 이 조작함수를 基本圖形要素의 단위처리(Transaction)과 관련된 위상함수(Generic Topology Operator : Euler Operator)와 이들을 조합하여 실제모델을 구성하는데 이용되는 모델링함수(Modeling Operator)로 구분하고 있다. 이 조작함수들은 先後處理를 실제로 가능하게 하며, 使用者가 직접 이용하는 부분으로, 가능한 使用者의 편의를 도모하고, 유연하게 되도록 구현하고 있다.

1. 位相函數

基本圖形要素를 조작하는 位相函數는 오직 한 단위 작업만을 목표로 작성되며, Euler 공식에 의해 수학적 唯一性이 보장되나, 階層的 특성이나 屬性資料에 대해 고려하지 않는다. Wawr-

Table-1. Euler Operators for Non-manifold Solid Modeling

Generic Topology Operator		Actions
m-mr		make model, region
m-sv		make shell, vertex
m-vl		make vertex, loop
m-ev		make edge, vertex
m-e	m-e	make edge
	m-eks	make edge, kill shell
m-f	m-fl	make face, loop
	m-flrs	make face, loop, region, shell
k-f	k-flrs	kill face, loop, region, shell
	k-fl	kill face, loop
k-e	k-e	kill edge
	k-ems	kill edge, make shell
	k-eml	kill edge, make loop
	k-efl	kill edge, face, loop
k-v	k-vfle	kill vertex, face, loop, edge
	k-vl	kill vertex, loop
	k-vrsfl	kill vertex, region, shell, face, loop
	k-vs	kill vertex, shell
esplit		slpit edge
esqueeze	esqueezek-ev	edge squeeze, kill edge, vertex
	esqueezek-e	edge squeeze, kill edge

yzynek는 Non-manifold Solid를 묘사하는 데는 Table-1에 열거된 11개의 位相函數가 필요하다고 하였다[7].

2. 모델링 函數

모델링 함수의 종류는 先後處理 프로그램의 목적에 따라 달라진다. 일반적으로 데이터베이스의 운영, 관리를 위해서는 資料의 추가, 변경, 삭제 및 검색의 네가지 기능이 기본이 되는 것으로 알려져 있다. 또, 간단한 入力資料로 목적하는 모델에 유사한 초기화면을 구성하도록 하면 使用者에게 대단히 유리하다. BASYS는 주요 모델링 함수로 資料의 追加, 變更, 削除 및 檢索技能과 初期畵面의 構成機能을 포함하고 있다.

가. 初期畵面의 構成

Boundary 모델은 간단한 構造에도 많은 Vertex와 Edge가 필요하므로, 초기에 대략적인 모델을 컴퓨터에 의해 자동생성 하게 하면 使用者의 노력을 크게 절감시킬 수 있다. 건물의 경우, 층수와 경간의 크기로 대략적인 初期畵面을 손쉽게 구성할 수 있으며, 使用者는 제공된 초기화면을 이용하여, 마치, 조각과정처럼 資料의 첨삭을 통해 목적하는 모델을 구성한다. Fig. 7은 BASYS에 의해 생성된 初期畵面의 예이다.

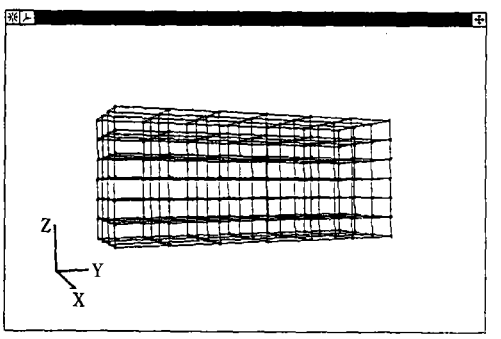

Fig. 7. An Examples of Auto Generated Grid
X : 3@5m, Y : 6@7, Z : 6@3m

나. 資料의 蒐集

모델을 構築하는 과정이나 변화시키는 과정에서는, 대상이 되는 圖形要素와 작업방법 등 資料가 필요하다. BASYS에서는 시스템의 單純性을 높이고자 우선 對象要素(Operand)를 選定하고, 그 다음에 作業方法(Operator)을 選擇하도록 하고 있다. 이 資料의 習得은 Mouse와 Keyboard를 이용하고 있는데, Mouse에 의한 資料의 획득에는 Fig. 8에서 圖示한 바와 같이 假想의 3次元 透視圖를 이용한다.

Key board를 이용하는 경우, 기존 데이터 구조와 부합하는 資料의 入力이 보장되어야 하므로, 제한적으로 資料를 입력하도록 하는 장치가 필요하다. Fig. 9는 이런 목적으로 작성된 資料 입력용 박스의 예를 보이고 있다.

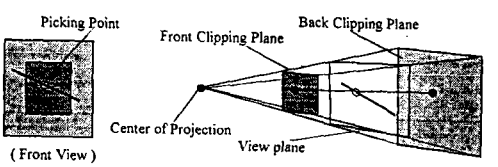

Fig. 8. Schmatic Description of Picking an Obejct

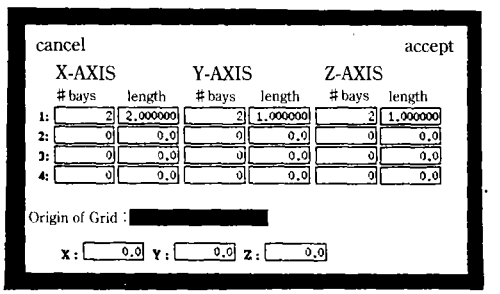

Fig. 9. An Example of Dialog Box

다. 圖形資料의 追加

圖形資料를 追加하는 방법은 낱 圖形要素를 하나씩 追加하는 방법과 旣往에 입력되어 있는 모델의 일부를 複寫하는 방법이 있다. 圖形資料가 추가되면, 이에 종속되는 屬性資料도 추

가되어야 하고, 도형간의 階層關係도 定立되어야 한다. BASYS는 ① 任意의 벡터 방향으로, ② 임의의 회전축을 중심으로, ③ 임의의 평면상의 원주방향으로, ④ 임의의 평면에 대한 反射 이미지를 이용하는 방법을 구비하고 있다. 아울러, 새로이 추가되는 資料가 기왕에 있는 資料와 중복되거나 모델의 一貫性에 영향을 미치는 일이 일어나지 않도록 점검하는 기능도 구비하고 있다.

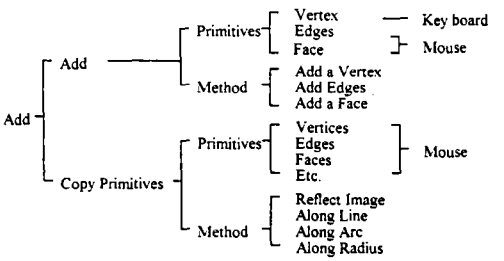

Fig. 10. Menu Tree for Add Operation

라. 資料의 變更

모델의 변경은 Topology에는 영향이 없으므로 좌표값을 가지고 있는 유일한 圖形要素인 Vertex의 위치를 옮김으로서, 즉, 좌표값을 변경시킴으로서 이루어진다. Vertex의 좌표변경에 따라 발생할 수 있는 문제점은, 기존 자료와 중복되거나, 위상학적으로 발생할 수 없는 경우 및 圖形要素의 특성을 변경시키는 경우로서, Fig. 11은 BASYS에서 사전에 점검하는 주요 문제점을 圖示한 것이다. 이들 중의 일부는 複數의 Vertex를 옮기는 작업 도중에도 발생할 수 있으므로, 模擬作業을 통하여 데이터베이스의 무결성을 확인하는 과정이 필요하다. Vertex의 좌표를 변경하는 방법은 ① 기존 Vertex를 삭제하고 새로운 Vertex를 追加하거나, ② 변경되는 Vertex와 관련된 모든 要素를 검색하여 중간 절점의 좌표는 비례적으로 변경하는 등, 필요한 조치를 취하는 방법이 있는데, BASYS에서는 후자의 경우를 채택하여 시스템의 一貫性을 중시하였다.

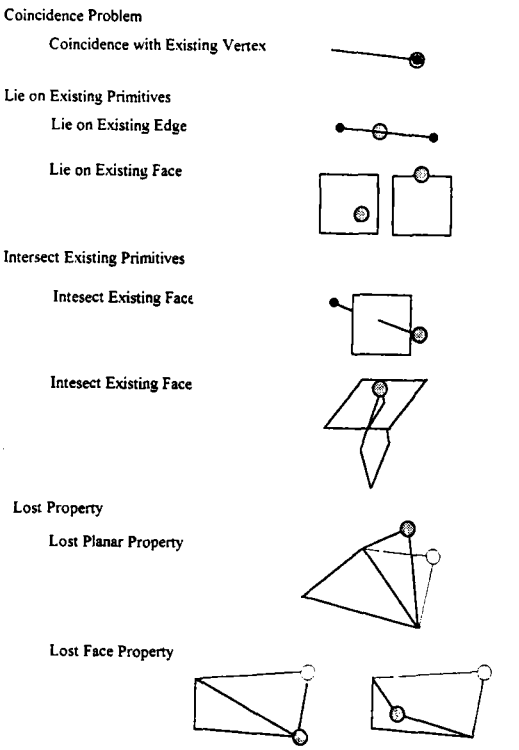

Fig. 11. An Illustration of Possible Errors Could be Happened While Move(or Add) Vertices

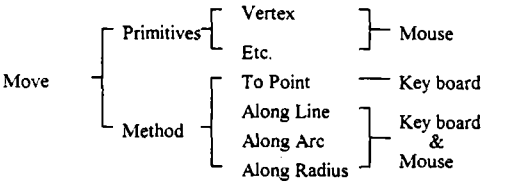

Fig. 12. Menu Tree for Move Operator

마. 資料의 削除

圖形要素를 삭제하는 때에는 階層構造 전체에 걸쳐 그 영향을 검토하여야 한다. 하위요소가 삭제되는 경우, 이 要素를 소유하는 상위요소는 그 영향을 받아야 하지만, 상위요소를 삭제하는 경우, 하위요소는 삭제 되어서는 안된다. Wire Frame 모델이 아닌 경우, 삭제된 要素의 제원을 알고 있다고 하여도, 종전과 같은 位相構造와

屬性資料 및 SDV, MSH Level의 상태를 복원하는 것은 매우 어렵다. 따라서, Undo등 기능은 사용에 상당한 제한이 있더라도, 削除 전에 從前의 전체 데이터베이스를 臨時로 저장하고, 삭제작업이 완료된 후에 使用者의 확인을 받아 임시 저장된 내용을 삭제하도록 구성한다. BASYS는 삭제할 要素를 화면에서 선택하도록 하여 誤謬를 줄이도록 하고 있다.

바. 資料의 檢索

理想的인 모델을 구축하기 위해서는 많은 Feed Back이 필요하다. 따라서, Feed Back 과정에 가능한 많은 資料를 제공하기 위해, 圓形要素의 Topology 와 Geometry 및 屬性資料를 동시에 검색할 수 있도록 모델링 함수를 作成한다. BASYS는 모델을 구성하는 동안 使用者가 필요한, 총 부재의 수난 전체 構造物의 무게, 부재의 길이, 두 점간의 거리, 주요 재질, 要素의 종류 등을 주요 검색 대상으로 한다. 또, 이 검색으로 얻어진 값을 다른 函數의 初期값으로 설정하도록 하여 작업과정을 단축하고 있다.

사. 屬性資料의 管理

대부분의 屬性資料는 Lookup Table로 저장함으로서, 데이터베이스의 무결성을 용이하게 유지할 수 있다. 실제로 BASYS에서는 이 테이블들을 STR Level에서만 조작할 수 있도록 함으로서, 계층구조의 이점을 살리고 있다. 주요 속성 資料로는 ① 재질 資料 ② 재료의 규격 資料 ③ 하중 資料 ④ 하중의 조합 상태 ⑤ 경계조건 등 有限要素 解析에 필요한 제반자료가 포함된다. 각 屬性資料는 작업의 균일성을 위해, 부문별로 관리하도록 메뉴를 구성하되, Table-2에 예시되어 있듯이, 각부문에서 일정한 운영패턴을 유지함으로서 使用者가 혼란을 느끼지 않도록 하였다.

또, 屬性資料 테이블을 생성하거나 추가하는 경우, 산업표준이나 기타 공공표준으로 사용되는 資料를, Scroll 화면 등을 이용하여 제공함으로서, 使用者의 편의를 돕고, 입력 오류를 예방하였다.

3. 要素의 分割

有限要素 解析을 위한 요소분할은 Edge 요소를 이용한다. 즉, ① 분할할 Edge를 선택하고, ② Edge당의 요소수를 확정한 후, ③ 분할할 비율을 입력함으로서 이루어진다. 분할비율을 임의로 조정할 수 있도록 하면, STR Level에서 관리해야 할 부재수를 줄일 수 있을 뿐 아니라, 要素의 형태를 조절할 수 있어 매우 유익하므로 BASYS는 방향을 구분할 수 있는 use 要素를 이용하여 첫번째 要素의 크기를 기준으로 다른 要素의 길이를 상대적 크기로 입력하도록 하고 있다. 또, 방향의 손쉬운 전

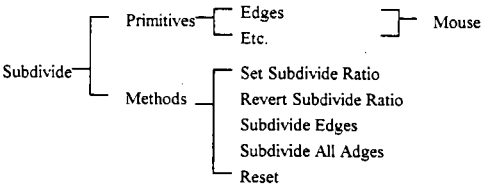

Fig. 13. Menu Tree for Subdivide Operator

Table-2. A Paradigm for Managing Attribute Data

Attribute Categories	Handling Operators
Material Property Data Physical Property Data Load Instances Load Combinations Boundary Informations	· Creat A New Table · Select A Table · Attach The Selected Tables to The Selected Primitives · Modify The Selected Tables · Delete The Selected Tables

환과 전체 構造物을 동일한 要素로 자동 분할하는 보완적 기능을 구비하고 있다.

4. 메쉬의 生成

메쉬는 SDV Level에서 정해진 분할비율에 의해 생성된다. Face 要素를 가진 모델의 경우, 별도의 Vertex 삽입이 필요하다. 이 부문에서는 有限要素 解析과의 Feed Back을 위한 부분적 메쉬기능, 자동메쉬 및 강도행렬의 규모를 줄이기 위한 Renumbering과 메쉬상태를 점검 할 수 있는 기능이 있다.

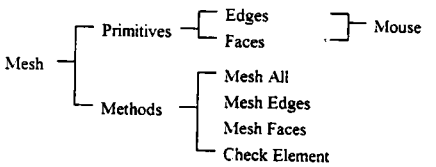

Fig. 14. Menu Tree for Mesh Operator

5. 後處理 프로그램

선처리 프로그램에 비하여 후처리 프로그램은 구성이 간단하다. 후처리의 목적은 解析된 결과를 판독하는 것이므로, 계층 등의 구별이 필요없게 된다. 有限要素 解析의 종류를 크게 ① Static, ② Modal Analysis, ③ Time History로 구분할 수 있으므로 이에 적절한 표현방법이 필요하다. BASYS는 Static 및 Modal Analysis의 경우에는, 변형후의 결과를 보여주도록 하고, 변형이 미소할 경우 과장하여 표현하는 방법을 제공하고 있다. Time History의 경우에는 시간별로 변형상태와 응력의 분포를 보여주는 방법과, 특정 要素의 시간별 응력, 변위를 X-Y 평면에 보여줄 수 있는 방법을 병행하였다.

Ⅶ. 運用 例

1. 모델의 構成

先後處理 프로그램의 효용성을 보이기 위하여

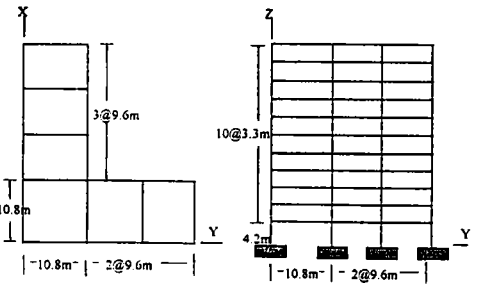

Fig. 15. A L Shaped Building

Fig. 15와 같은 L형 건물에 대한 解析 과정을 일례로 모델을 구성하였다.

모델의 構成順序는 다음과 같다.

① 初期畵面을 生成한다.(Fig. 16)
② 불필요한 Vertex와 Edge를 삭제한다.(Fig. 17)
③ 기초층을 이용하여 첫층을 복사한다.
④ 기초의 묘사를 위해 기초층의 Edge를 삭제한다.(Fig. 18)
⑤ 첫층을 Z방향으로 복사한다(복사수량 10).(Fig. 19)
⑥ 구조재료, 부재규격을 선택한다.
재질 : ASTM A 36 Steel
규격 : 보(W16×100), 기둥(W30×116)
⑦ 기초부분을 모든 방향으로 고결시킨다.
⑧ 하중조건을 작성한다.
Vertex 하중 : 모든 기둥의 수직하중 80톤
지진하중 : Eℓ Centro
⑨ 모든 부재를 2 要素로 분할한다.
⑩ 분할된 要素를 메쉬한다.

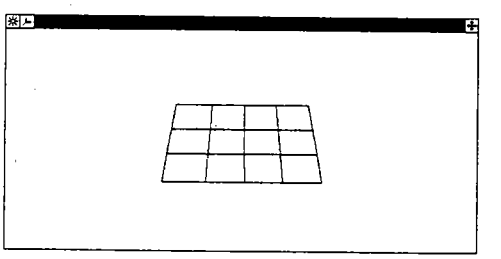

Fig. 16. After Grid Generating

⑪ 有限要素 解析을 위한 入力資料를 작성한다.

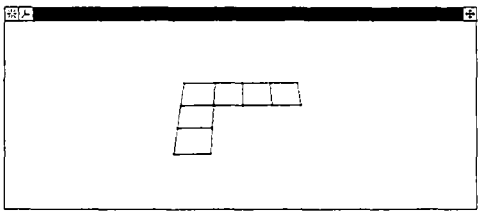

Fig. 17. After Delete Redundent Elements

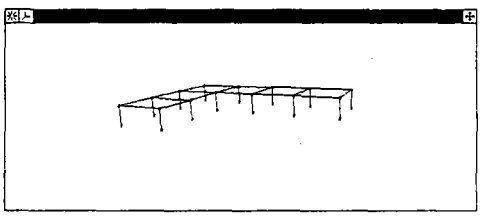

Fig. 18. After Delete Ground Level Edges

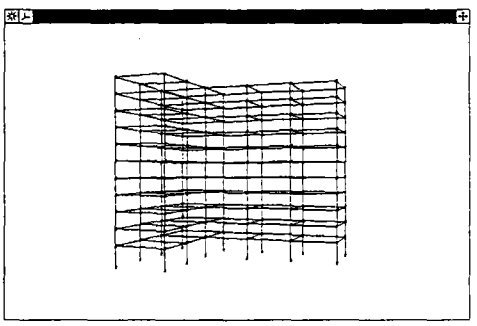

Fig. 19. After Copy the First Floor Image to Remaining Floors

2. 後處理

구성이 완료된 모델은 有限要素 解析 프로그램에 의해 처리되면 그 결과를 읽어들여 화면에 표시한다. 본 예제는 ABREAST에 의해 解析되었다. Fig. 20은 예제 構造物의 Modal Analysis 결과 중 한 모드를 보인 것이다.

構造物을 잘 이해하기 위해서는 構造物의 변화를 실제시간의 흐름에 따라 시뮬레이션 해

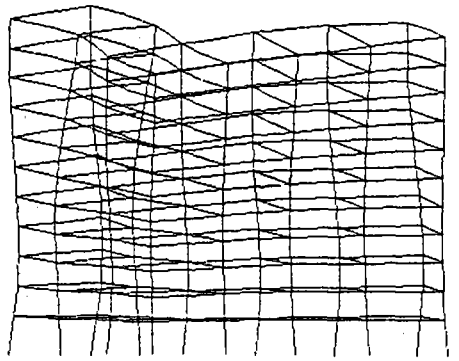

Fig. 20. A Displaying Example of Modal Analysis(Mode No. 20)

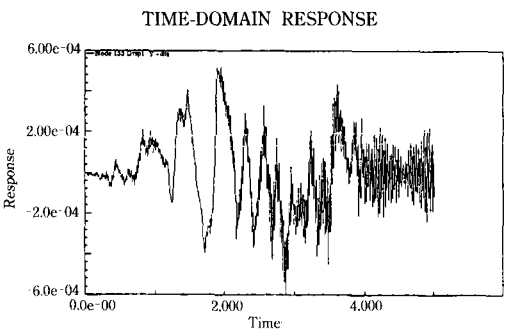

Fig. 21. A X-Y Ploting Example of Time History Analysis

보는 것이다. Fig. 21은 使用者가 선택한 임의의 한 점이 하중작용 기간동안 어떻게 거동하는지 알 수 있도록 X축을 시간, Y축을 변위로 표시한 것이며, 참고로 본 例題 解析過程에서 만들어진 화일의 크기와 소요시간은 HP-730 Workstation에서 Table-3과 같았고, 모델의 구성과 解析 전 과정에는 대략 2시간 내외가 소요되었다.

Ⅷ. 結 論

現代의 構築構造物은 그 技能과 施工上의 편의성을 고려하여 一體式, 대규모화하는 傾向이 있으며, 이와 같은 趨勢에 맞추어 構造解析도 有限要素 解析이 활발해지고 있다. 有限要素 解析의 올바른 이용을 위해서는 視覺裝備를 通

Table-3. Summary of File Size and Elapse Time

File Name	File Size	Elapse time
	K bytes	sec
Modeling File	812	
FEM Data File	141	
Analysis Results		
• Mode Shape(20 Mode)	1,222	299
• Time History(500 sec)	47,792	760

한 資料의 입출력이 강력히 요청되고 있으므로, 많은 先後處理 프로그램이 개발되고, 또 응용되고 있다. 본 논문을 통하여 先後處理 프로그램의 資料構造, 프로그램의 흐름과 유의사항 및 주요 조작함수에 대하여 검토하였던 바 다음과 같은 결론을 얻었다.

1. 建物의 設計에 이용할 先後處理 프로그램은 Topology와 Attribute 데이터베이스를 분리함으로서 部材의 規格과 材料特性을 잘 描寫할 수 있었다.

2. Attribute Data는 Lookup Table 構造를 利用함으로서 데이터베이스가 갖추어야 하는 無缺性 原則을 容易하게 具現할 수 있었다.

3. 建物은 주로 線, 面材로 이루어지므로 Non-manifold Boundary 모델로 能率的으로 그 構成狀態를 描寫할 수 있었다.

4. 프로그램의 運營 能力을 向上하기 위해서는 設計過程(STR Level)과 解析 過程(SDV & MSH Level)으로 區分하는 것이 유리함을 알 수 있었다.

5. 모델의 구성은 基本的으로 11개의 位相函數로 가능하며, 이 位相函數를 이용하여 모델을 구축하는 모델링 函數는 일반 데이터베이스의 操作 基準인 資料의 追加, 變更, 削除 및 檢索 기능을 만족시키면 충분함을 확인할 수 있었다.

6. 先後處理 프로그램의 제작을 위해서는 客體指向 프로그래밍 기법이 프로그램의 製作이나 改善 및 補完에 매우 유리함을 확인할 수 있었다.

이 硏究의 基礎가 된 프로그램 BASYS는 National Center for Earthqueake Engineering Research의 지원으로 미국 CORNELL 대학의 Abel 敎授의 指導아래 S. Srivastav, W. Christoper 및 筆者에 의해 開發되었다.

參 考 文 獻

1. Robert Haber & Abel F. John, 1982, Discrete Transfinite Mapping for the Description and Meshing of Three Dimensional Surfaces Using Interactive Computer Graphics, International Journal for Numerical Method in Engineering, Vol. 18, pp 41-46.
2. Srivastav S, 1991, Three Dimensional Modeling and Simulation of building for Seismic Anlysis, Ph. D. thesis, Cornell University.
3. Carlos Z. Pesquera, Willian McGuire & Abel F. John, 1083, Interactive Graphical Processing of Three Dimensional Framed Structures, Computer and structures Vol. 17, No. 1 pp1-12.
4. Ibrahim Zied, 1991, CAD/CAM Theory and Practice, McGrow hill.
5. Weiler K., The Radial Edge Structure : A Topological Representation for nonmanifold Geometric Bounding Representation, Geometeic modeling for CAD Application, 1988, pp. 3-36.
6. Srivastav S. & Abel F. John, 1991, Generalized Computer Modeling of Structures, IASS symposium 1991.
7. Wawrzynek P., 1987, Interactive Finite Element Analysis of Fracture Process : An Intergrated Approch, Ph. D. thesis, Cornell Univ.