

구조해석에서 객체지향 방법론의 도입 Application of Object-Oriented Methodology for Structural Analysis and Design

이주영* 김홍국** 이병해***
Lee, Joo-Young Kim, Hong-Kook Lee, Byung-Hai

ABSTRACT

This study presents an application of object-oriented methodology for structural design process. A prototype system of integrated structural design system is developed by introducing a structural analysis object model(SAOM) and structural design object model(SDOM).

The SAOM module, which is modeled as a part of structural member, performs structural analysis using FEM approach and the SDOM module checks structural members based on Korea steel design standard. Above mentioned models are abstract, encapsulatable and reusable.

1. 서론

본 연구는 통합 구조설계 시스템의 구축에서 객체지향적인 방법론을 도입하여 건축 구조물을 객체모델링하고, 구조해석 객체모델(Structural Analysis Object Model, SAOM)과 구조설계 객체모델(Structural Design Object Model, SDOM)을 개발하여 이를 도입한 통합 구조설계 시스템의 프로토타입(Prototype)을 제시한다.

1.1 배경

건축 구조물의 구조설계는 구조물 모델링, 예비설계, 구조해석, 부재설계, 구조도면 작성등의 과정을 거치게 된다. 이와같은 일련의 구조설계 단계 사이에는 다양한 상관관계를 가지고 있다. 이와 같은 이유로 통합 구조설계 시스템에서는 각 단계가 긴밀한 協力關係가 이루어 지도록 해야한다.[1]

현재의 통합 구조설계 시스템을 구축하는 여러가지 방법중 주요한 두가지를 언급하면, 하나는 混合的 방법이고, 다른 하나는 統一的 방법이다. 혼합적 방법은 블랙박스를 사용하는 방법으로서 안정되고 이미 성능이 인정된 구조해석 프로그램이나, CAD시스템을 사용한다. 開發者는 이 블랙박스를 연결시키는 데이터 베이스 시스템이나 연결용 프로그램을 작성하게 된다. 이는 개발 비용이 적게 드는 장점이 있으나, 전적으로 블랙박스에 의존하므로 입력되는 데이터와 출력되는 데이터가 한정되어 있기 때문에 개발자가 필요에 따라 다양한 데이터를 참조할 수 없다. 통합적 방법은 개발자가 모든 시스템을 개발하므로 데이터모델이나 모듈에 접근의 제한이 없는 장점이 있으나, 개발 시스템의 오류를 충분히 검증할 필요가 있기 때문에 개발시간과 비용이 비교적 많이 소요되는 단점이 있다.

통합 구조설계 시스템에서 구조설계 과정의 각 단계가 긴밀한 협력관계가 이루어 지도록 하기 위해서는 통합적인 방법을 사용해야 한다. 이에 따라 구조설계 시에 중요한 해석모듈의 개발이 필요하다. 현재 이 해석모듈은 유한요소법이 널리 사용되고 있다. 그러나, 이 해석모듈은 수만줄의 소스코드(Source Code)로 이루어져 있어 개발하려는 시스템에서 사용하기 위해서는 상당한 노력을 가

* 한양대학교 건축공학과 석사과정
** 한양대학교 건축공학과 박사과정
*** 한양대학교 건축공학과 교수

지고 분석하고, 검증하여야 한다.

통합 구조설계 시스템은 시간이 지남에 따라 요구되는 직업이나 방법이 변화되기 때문에 개발자는 이를 재설계 하거나 재구축하여야 한다. 이와같은 요구에 개발자는 현재의 수만줄의 소스코드에서 재고하여야 할 부분과 재사용할 부분을 구분하여야 한다. 그리고, 미래의 변경에 대한 재사용성을 극대화 시키도록 새로운 시스템을 구축해야 한다. 이때의 주요문제는 어플리케이션의 擴張性, 존재하는 소프트웨어의 再使用性, 과거 소프트웨어를 현재의 표준으로 바꾸는 도구, 프로그램 내부에서의 상호 정보교환등이다.[2]

객체지향 방법론은 최근에 시스템 개발을 위해 廣範圍하게 사용되어 지고 있다. 이것은 順次的인 방법론이나 構造的 방법론과 매우 다르다. 보통 구조해석 프로그램에서 사용되어지는 방법론은 구조적 방법론으로 개발자는 프로그램을 실행 가능한 여러가지의 모듈로서 나눈다. 이 모듈은 실행되어지는 프로시듀어(Procedure)로 실행될 때 일정한 형식이나 순서의 데이터를 입력시키고 출력시켜야 한다. 그러나, 이 데이터의 구조는 상당히 복잡한 것이 일반적인 경우이기 때문에 개발 비용을 증가시키는 원인된다. 재사용성이나, 캡슐화를 고려해 볼 때 프로시듀어와 데이터의 구조를 하나의 덩어리로 만들고, 이것에 정의한 신호를 보내 작업이 이루어 지도록 하는 것이 많은 이익을 제공한다. 즉, 어떤 모듈에 어떤 데이터의 형식을 가지고, 어떤 순서대로 입력을 주어야 하는 것을 생각할 필요가 없기 때문이다. 여기서 이 프로시듀어와 데이터의 구조가 하나의 덩어리로 이루어진 것이 객체의 개념이다.[3]

1.2 연구목적 및 절차

본 연구실에서는 구조해석 프로그램을 위한 전처리 작업과 후처리 작업을 위하여 통합 구조설계 시스템을 위한 객체지향 데이터 베이스를 개발한 적이 있는데, 이 연구를 바탕으로 구조물의 객체지향 데이터 모델을 개발하였다.[4] 본 연구는 구조물의 데이터와 코드가 함께 존재하는 객체지향적인 데이터 모델을 구축하고, 이 데이터 모델에 실제의 구조물과 유사한 舉動을 할 수 있도록 하기 위해 유한요소법을 적용시켜 재사용성이 높고, 캡슐화되고, 추상적인 의미 표현력이 높은 객체지향적 구조물의 데이터 모델을 제시한다.

- 객체지향적 접근

일반적인 유한요소법을 이용한 구조해석 프로그램은 구조물을 유한요소의 형태로 모델링하고, 요소별로 강성행렬을 계산하고, 전체 좌표계에 대하여 좌표변환시키고, 전체의 강성행렬로 치환한 후 연립방정식의 해를 구하여 변위를 구하고, 다시 변위를 이용하여 해당 요소에 대한 응력을 계산한다. 이와같은 방법을 사용할 경우에 유한요소법으로 프로그램을 작성하여 보았거나, 이론적으로 이해를 하고 있어야 이미 작성되어 있는 유한요소해석 프로그램의 소스코드를 재사용할 수 있다.

그러나, 소스코드의 덩어리도 시공현장에서 기둥이나 보를 인지하고, 조립하는 것처럼 意味의 표현이 뛰어나고 캡슐화할 수 있는 방법론을 사용할 경우에 유한요소해석 프로그램의 비전문가도 접근이 용이하다. 그래서, 구조물의 기둥과 보등을 객체지향적인 데이터 모델로 표현하고자 한다.

- 유한요소법을 이용한 구조해석 객체모델 구축

실제의 구조물은 하중을 견디는 데 내부적으로 유한요소법처럼 진행되지 않는 것은 자명한 사실이다. 이와같은 이유로 유한요소해석 프로그램의 개념을 意味적으로 실제 세계에 존재하는 구조물과 같이 표현하기가 어렵다. 많은 연구들은 개념적인 데이터 모델만 작성하고, 객체지향적인 프로그램기법을 유한요소법을 이용한 프로그램에서 경제조건이나 강성행렬을 구하거나, 좌표변환을 시키는 데 사용하도록 하고 있다.[5][6][7] 그러나, 본 연구에서는 이를 시공현장에서 사용하는 객체(기둥, 보)들과 같이 生成시키고, 結合시킬 수 있도록 하기 위해서 객체에 유한요소법의 해석부분을 內裝시키고, 객체들의 결합된 關聯性을 따라 부재응력을 구하도록 하였다.

- 구조설계 객체모델과 통합 구조설계 시스템의 모형 개발

독립적인 객체에 해석부분 뿐만 아니라, 부재가 설계기준에 적합한지의 여부를 판단할 수 있도록

설계부분도 추가한 구조설계 객체모델을 개발하고, 이 두모델을 사용하여 객체지향 데이터 관리 모델인 OMISAD(Object-Oriented Data Model for Integrated Structural Analysis & Design)와 이를 적용한 실험적인 통합 구조설계 시스템인 AISAD(Object-Oriented Application for Integrated Structural Analysis & Design)를 개발하였다.

2. 객체지향적 방법과 구조물 객체모델

2.1 객체지향 개념

객체지향개념은 처음으로 클래스 개념을 도입한 SIMULA(1966)라는 언어에서 찾을 수 있고, 그래픽 중심의 대화식 프로그래밍 환경을 갖춘 Smalltalk(1981)에서 그 기반을 확고히 하였다. 객체지향개념은 60년대 후반에서 시작하여 70년대에 중요한 정의가 이루어지고, 80년대 중반에 확장되어지고, 90년대에 데이터베이스, 운영체제, 인공지능, 하드웨어 및 소프트웨어 설계와 응용등에 중심연구과제가 되고 있으며, 건축, 토목에서도 건축물의 형상이나 공학 데이터의 표현 및 처리에 그 적용성의 연구가 활발히 진행되고 있다. 객체지향적인 분석에서는 재사용성과 캡슐화 그리고, 의미의 추상화를 위하여 객체, 클래스, 메시지, 메소드, 상속성등과 같은 개념들을 사용한다.

객체는 실세계에 존재하는 대상이다. 예를 들면, 구조물을 구성하는 기둥, 보, 하중등이 객체지향 시스템내에서 객체로서 표현된다. 객체는 屬性(Attribute)과 이에 연관된 메소드(Method)가 함께 뭉친 개념으로, 이는 데이터 구조와 행위방식을 말한다. 공통된 형태의 속성과 메소드를 가지는 객체들을 정의한 것을 클래스라 하며, 인스턴스(Instance)는 속성과 메소드를 정의한 클래스의 객체이다. 객체들은 메시지를 주고 받으면서 상호작용을 한다. 메소드는 객체에 메시지를 보냈을 때 그에 대해 일정한 작용을 수행하는 프로시저(Procedure)다. 유사한 클래스를 일반화시키고, 유사한 클래스에서 일반화된 특성을 그대로 물려받아 詳細化하여 정의할 수 있다. 이때 하나의 클래스가 다른 클래스로부터 속성이나 메소드에 대한 정의를 물려받는 것을 相續(Inheritance)이라고 한다.

객체와 클래스간의 관계성을 객체의 구조와 함께 기술한 모델을 데이터 모델이라 한다. 실세계에 존재하는 대상을 객체로 抽象化하고 이를 一般化(Generalization), 詳細化(Specialization), 複合化(Aggregation) 등의 관계성(Relationship)을 가지고 복잡한 데이터를 표현한다.[8]

2.2 구조물 객체모델(Structure Object Model)

통합 구조설계 시스템의 목적은 구조물 모델링에서 시공도면의 작성까지를 유연하게 연계시키는 데 있다. 이는 데이터 베이스를 중심으로 설계정보를 관리함으로써 가능하다. 그러므로 데이터베이스내의 설계정보의 표현과 처리방법의 효율성이 중요하다. 이를 위해서는 실세계의 정보를 유사하게 시스템의 데이터 베이스에서 표현하고 처리되도록 하는 것이 복잡한 시스템을 처리하는 데 유리하다. 이에 본 연구에서는 구조물 객체모델을 제시한다.

이 구조물 객체모델은 실세계의 구조부재를 실세계에서 존재하는 유사한 형태로 모델링하므로 모델의 표현력이 높다. 그리고, 구조부재의 기둥을 부재모델의 객체에 내장하고 있어 복잡한 유한요소 해석모델을 이해하지 않아도 자유롭게 사용이 가능하다.

- 구조설계 정보

건축 구조물의 구조설계는 구조물 모델링, 예비설계, 구조해석, 부재설계, 구조도면 작성등의 과정을 거치게 된다. 일반적으로 위의 일련의 과정은 구조설계 시 순차적으로 반복하여 이루어진다. 예를 들면 부재설계 과정에서 예비설계 시에 가정한 부재가 만족스럽지 못하면, 예비설계 과정으로 되돌아가서 순서에 따라 위의 과정을 다시 실시해야 한다.

위 과정을 고찰할 때, 구조물에서 하나의 부재를 설계하기 위해서는 구조부재의 형상 데이터와 구조 시스템 구성 데이터, 재료, 각 부재간의 상호 관계성, 하중, 경계조건 등에 의한 그 부재의 응력 계산 및 설계 기준의 적절성을 판별하는 것이 주요 문제이다. 현재 일반적인 시스템에서는 위의 과정을 각각 별도의 모듈에서 데이터를 읽고 데이터를 작성하고 하는 작업을 수행해야 한다. 그래서,

시스템 개발자는 각 모듈 뿐 아니라, 데이터 구조도 잘 알고 있어야 한다. 그러나, 위 과정은 하나의 부재에 작용하는 행위이기 때문에, 데이터 구조는 각 부재의 속성으로, 모듈은 각 부재의 메소드로 객체모델링을 할 경우는 실세계에 존재하는 부재와 유사하게 사용할 수 있기 때문에 쉽게 재사용하는 것이 가능하고, 미래의 변화에도 능동적으로 대처할 수 있다. 그래서, 본 연구에서는 구조물의 객체모델을 주요하게 부품(Component), 하중(Load), 지지조건(Support)으로 나누었다.

- 구조물 객체모델의 구조

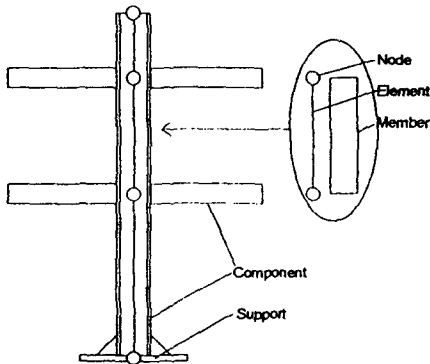


그림 2.1 실세계의 구조부재에서 객체의 구성

구조물 객체모델의 부품(Component)클래스는 보(Beam)클래스나 기둥(Column)클래스를 一般化 시킨것으로 이것은 구조부재(StructuralMember)클래스와 절점(Node)클래스를 複合化 시킨것이다. 구조부재 클래스는 요소 클래스와 부재 클래스의 속성을 多衆相續하여 詳細化한 것이다. 그림 2.1은 실세계에 존재하는 부재를 概念的으로 나타낸 것이고, 그림 2.2는 이것을 구조물의 객체모델로 나타낸 것이다.

요소(Element) 클래스는 1차원 유한요소해석이 이루어지도록 구조부재 클래스를 一般化시킨 객체이다. 요소 클래스는 강성행렬과 자유도를 정의하여 2차원 트러스에서 3차원 라아멘까지 詳細化시킬 수 있다. 절점(Node) 클래스는 요소 클래스와 항상 connect_to 관계성으로 연결되어 있고, 유한요소법에 의한 구조물의 해석을 위하여

모델되어져 있다. 이 절점 클래스가 구조설계 객체모델의 核心이다. 부재(Member) 클래스는 구조부재 클래스가 설계 기준식에 만족하는 지를 검토하는 클래스로서 구조부재를 一般化시킨 것이다. 이는 설계기준의 기본적인 식만을 갖추고 있다. 구조부재(Structural Member)클래스는 요소 클래스와 부재 클래스를 詳細化시킨 것으로 유한요소해석과 부재설계를 한다. 부품(Component)클래스는 기둥, 보, 압축 및 인장재등의 실제적인 구조부재를 一般化시킨 클래스로 구조부재 클래스와 절점 클래스를 複合化시킨 것이다. 부품 클래스는 다른 부품 클래스와 connect_to 관계성을 가지고 결합되어 질

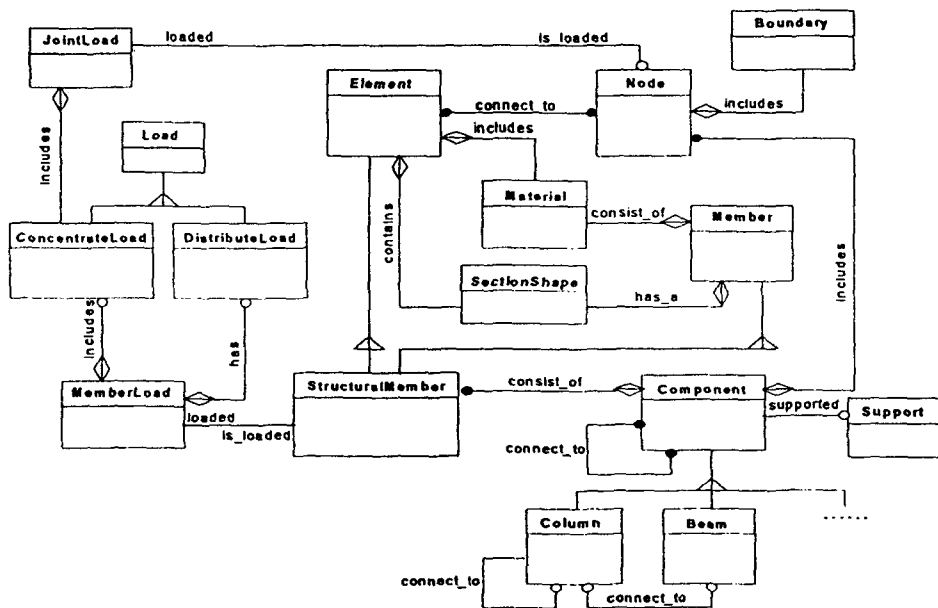


그림 2.2 구조물 객체모델 (Structure Object Model)

수 있다. 그리고, 절점 클래스와 구조부재 클래스를 내재하고 있고, 부품 클래스의 결합시에 필요하면 구조부재 클래스와 절점 클래스를 생성하거나 제거한다. 지지조건(Support)클래스는 부품 클래스와 supported관계성으로 연결되어지며, 절점 클래스의 경계조건인 경계조건(Boundary)클래스를 정의한다. 하중(Load)클래스는 외력을 정의하며 해석시, 절점의 경계조건을 정의한다. 구조시스템(StructuralSystem)클래스는 부품 클래스를 詳細化한 기둥(Column)클래스, 보(Beam)클래스등과 하중 클래스, 지지조건 클래스를 複合化하여 이루어진다.

3. 구조해석 객체모델과 구조설계 객체모델

3.1 정식화

- 골조요소의 절점력과 절점변위

유한요소법에서는 구조물을 유한개의 간단한 형상을 가진 요소의 집합체로 생각한다. 골조구조는 보나 기둥의 유한개의 부재를 유한요소로 생각하고, 여기서는 골조요소라 부르기로 한다. 골조요소 중 트러스부재는 트러스 요소, 라아멘재는 라아멘 요소로 부르기로 한다.

여기서는 입체 라아멘 요소를 주요 모델로 하기 때문에, 입체 라아멘 요소에 대해서 언급하겠다. 골조요소는 절점을 통하여 힘을 전달하고, 절점을 통하여 변위를 나타낸다. 입체 라아멘 요소의 절점변위벡터의 성분은 절점*i*(시작점)에 대해서 x, y, z방향의 변위 u_i, v_i, w_i 와 x, y, z축 회전의 회전각 $\theta_{xi}, \theta_{yi}, \theta_{zi}$ 절점*j*(끝점)에 대하여 x, y, z방향의 변위 u_j, v_j, w_j 와 x, y, z축 회전의 회전각 $\theta_{xj}, \theta_{yj}, \theta_{zj}$ 의 총 12개로 이루어진다. 절점력 벡터의 성분도 위의 각 방향으로 총 12개로 이루어진다.

- 골조요소의 강성행렬

유한요소법으로 구조물을 해석하기 위해서는 그 요소에 대해 강성행렬이 필요하다. 입체 라아멘 요소에 대한 강성행렬은 직접법이나 에너지법으로 유도되는 과정을 여러 유한요소법 교재에서 다루었기 때문에 여기서는 생략한다. 이 강성행렬은 부재 좌표계에서 절점력과 절점변위의 관계를 표시한 것이다. 이것을 식으로 나타내면

$$\{F_e\} = [k_e]\{\delta_e\} \quad \dots\dots\dots(1)$$

$\{F_e\}$: 절점력벡터, $[k_e]$: 강성행렬(stiffness matrix), $\{\delta_e\}$: 절점변위벡터

이다. 이 강성행렬은 각 요소별로 계산되어야 한다.[9]

- 구조전체의 강성방정식

부재 좌표계에 대한 강성방정식은 전체 좌표계에 대한 강성방정식으로 전환시키는 것이 필요하다. 전체 좌표계에 대한 강성방정식은

$$\{\bar{F}_e\} = [\bar{k}_e]\{\bar{\delta}_e\} \quad \dots\dots\dots(2)$$

이고, 여기서

$$[\bar{k}_e] = [T_e]^T [k_e][T_e] \quad \dots\dots\dots(3)$$

$[T_e]$: 변환행렬

이다.

보통 유한요소해석 프로그램에서 구조전체에 대한 강성방정식을 만드는 방법은 절점력벡터와 절점변위벡터의 성분을 절점번호순으로 열거한다. 그리고, 각 요소의 강성행렬을 계산하고, 그것을 전체 좌표계에 대해 전환한 후 관련있는 절점력벡터와 절점변위 성분의 위치에 조합한다.[10]

본 연구에서는 구조전체의 강성방정식을 따로 만들지 않으며, 각 절점 클래스가 연관된 요소의 강성행렬을 찾아 각 절점에서 미지의 변위를 계산하도록 하였다.

- 절점변위의 계산

구조전체의 강성방정식은 미지의 절점변위와 기지의 절점변위를 가진다. 이 방정식을 나타내면

$$\begin{pmatrix} P_A \\ \dots \\ P_B \end{pmatrix} = \begin{bmatrix} K_{AA} & \dots & K_{AB} \\ \dots & \dots & \dots \\ K_{BA} & \dots & K_{BB} \end{bmatrix} \begin{pmatrix} d_A \\ \dots \\ d_B \end{pmatrix} \dots\dots\dots(4)$$

이고, 여기서 $\{d_A\}$ 는 미지의 절점변위이고, $\{d_B\}$ 는 기지의 절점변위이다. 이 식을 분해하면, 다음의 식을 얻을 수 있다.

$$\{P_A\} = [K_{AA}]\{d_A\} + [K_{AB}]\{d_B\} \dots\dots\dots(5)$$

$$\{P_B\} = [K_{BA}]\{d_A\} + [K_{BB}]\{d_B\}$$

여기서 미지의 절점변위 $\{d_A\}$ 를 구하기 위해서는 위 식에 대한 연립1차방정식을 풀면 된다.[11] 본 연구에서는 구조전체의 강성방정식을 만들지 않기 때문에 위 식을 이용하여 미지의 변위의 절점에서 Gauss-Seidel법을 사용하여 반복해법으로 변위를 구한다.

- 응력계산

절점변위가 구하여 지면, 각 요소의 응력을 계산하게 되는 데, 요소의 절점에서 응력은 요소의 부재 좌표계에 대한 절점력이므로 전체 좌표계에 대한 변위 $\{\bar{\delta}_g\}$ 에 의해

$$\{f_g\} = [k_g][T_g]\{\bar{\delta}_g\} \dots\dots\dots(6)$$

이다.

3.2 구조해석 객체모델

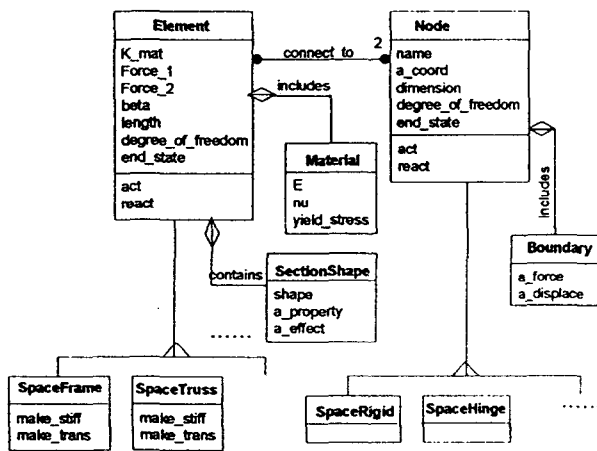


그림3.1 구조해석 객체모델

구조해석 객체모델은 그림3.1과 같은 관계성을 가지고 있다. 요소(Element) 클래스는 요소에 대한 강성행렬, 응력, 회전각, 길이와 자유도를 포함한 屬性을 가지고 있고, act와 react라는 메시지를 받아 수행한다. 요소 클래스는 절점(Node) 클래스와 connect_to 관계성으로 연결되는 데, 이때에 재료적 특성과 단면성능 그리고, 좌표를 각 관련된 부재로부터 읽어 강성행렬과 전환행렬을 계산하고, 전체 좌표계에 대한 강성행렬을 자신에 보관한다. 또, 요소 클래스는 상속성을 이용하여 자신의 속성을 입체 라아맨(Space Frame)이나 입체 트러스(Space Truss)와 같은 선형요소로 상속시킬 수 있다. 그림3.1에 보여 주고 있는 입체라아맨(SpaceFrame)클래스는 요소 클래스로부터

상속받은 것이다. 입체라아맨 클래스에서는 단지 자신의 강성행렬과 전환행렬만 만들어 주면 된다. 이것은 입체트러스 클래스나 다른 선형요소에서도 마찬가지이다.

절점(Node)클래스는 전체 좌표계에 대한 좌표를 가지고 있고, 역시 act와 react의 메시지를 받아 수행한다. 실제적인 구조부재의 해석은 절점 클래스에서 이 절점과 includes의 관계성으로 연결되어 있는 경계조건(Boundary)클래스의 미지의 변위가 만족되어질 때까지 반복하여 계산한다. 변위계산의 수행은 connect_to관계성으로 연결된 요소 클래스의 강성행렬과 기지의 외력벡터를 사용한다. 강절(S

paceRigid) 클래스와 활절(SpaceHinge) 클래스는 이 절점 클래스에서 모든 속성을 상속받으며, 자유도만을 정의하면 된다.

3.3 구조설계 객체모델

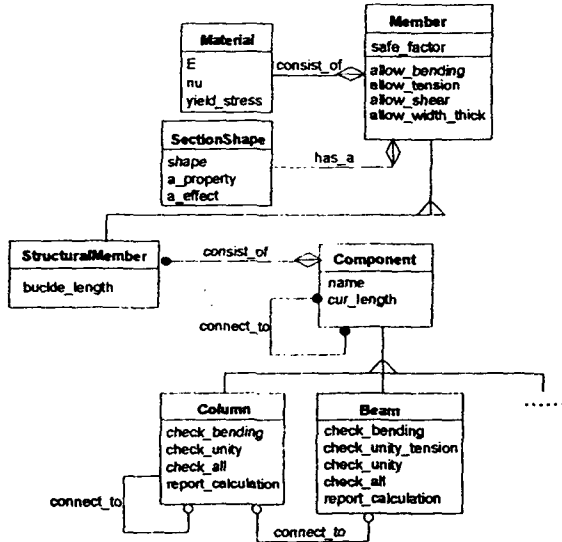


그림3.2 구조설계 객체모델

구조설계 객체모델은 한국강구조기준에 의한 부재설계를 위하여 설계했다. 그림3.2는 구조설계 객체모델의 클래스간의 관계성을 보여준다. 부재(Member)클래스는 기둥(Column), 보(Beam)등의 구조부재를 一般化한 것으로 부재의 허용응력을 계산하기 위한 메소드를 가지고 있다. 구조부재(StructuralMember)클래스의 메소드는 허용압축응력도를 계산하기 위한 좌굴길이를 계산한다.

기둥(Column)클래스는 압축응력과 휨응력을 각각 허용압축응력도 및 허용휨응력도로 나눈 다음과 같은 설계식을 사용한다.

$$\frac{\sigma_c}{f_c} + \frac{\sigma_{bx}}{f_{bx}} + \frac{\sigma_{by}}{f_{by}} \leq 1.0 \quad \dots\dots\dots(7)$$

σ_c : 압축응력도, σ_b : 휨응력도,

f_c : 허용압축응력도, f_b : 허용휨응력도

그리고, 보(Beam)클래스는

$$\frac{\sigma_x + \sigma_y}{f_t} \leq 1.0, \quad \frac{\sigma_x}{f_b} + \frac{\sigma_y}{f_t} \leq 1.0 \quad \dots\dots\dots(8)$$

σ_x : 압축응력도, f_t : 허용압축응력도의 식을 사용하여 부재를 검토한다.[12]

4. 통합 구조설계 시스템의 구현

4.1 설계(Design)

본 연구에서는 통합 구조설계 시스템을 구현하기 위하여 OMISAD와 AISAD를 개발하였다. OMISAD는 구조물 객체모델로 구조해석 객체모델과 구조설계 객체모델을 포함하는 객체지향 데이터 관리 모델이며, AISAD는 OMISAD를 사용자가 제어하도록 만든 객체지향적인 통합 구조설계 시스템의 모형이다.

이 통합 구조설계 시스템의 설계에서 주요한 목표는 캡슐화되어 재사용성이 높은 데이터 모델을 개발하여 제한된 자원과 짧은 시간에 유한요소해석 프로그램과 같은 해석모듈을 사용하는 것이다. 모형개발을 위해 입체 라아멘 요소의 보와 기둥을 Beam 클래스와 Column 클래스로 캡슐화시키고, 이를 생성하고 管理하는 StructuralSystem 클래스를 구현하였다.

4.2 구현

- 프로그램 언어

본 연구에서 사용한 C++는 어떤 요소는 객체의 개념을 사용하고, 어떤 요소는 객체의 개념을 사용하지 않을 수 있는 혼합형 언어이다. C++는 C에 객체지향적인 개념을 추가하고 단점을 보완한 C언어의 확장판이다. 이것은 인라인 서브루틴과 함수의 오버로딩(overloading), 多衆 相續性등의 객체

지향 프로그램의 많은 특징을 지원한다. 특히, 柔軟性(Flexibility)과 單純性(Simplicity)을 갖도록 설계하는 데 비용면에서 실행시간에서 효율적이다.[13] 본 연구에서 C++를 사용한 이유는 이상과 같은 객체지향적인 특징을 지원할 뿐 아니라 유한요소해석 프로그램이 수치해석을 위해 많은 수행시간을 소요하기 때문에 수행시간 면에서 다소 효율적인 C++를 사용했다.

- 개발 환경

하드웨어는 Workstation(HP Model 715/50)를 사용했고, 운영체제는 HP-UX 9.01 상의 X Window 시스템인 VUE환경을 사용했다. 사용자 인터페이스를 구현하기 위하여, 객체지향적인 인터페이스 도구인 OSF/Motif를 사용했다. UNIX는 32bit 운영체제로 하나의 세그먼트로 4GB의 메모리를 사용할 수 있고, 물리적인 메모리의 부족을 가상 메모리로 대체할 수 있기 때문에 많은 메모리를 요구하는 객체지향적인 프로그램에서 유용하게 사용되어질 수 있다.[14][15]

- OMISAD의 구조

OMISAD는 실세계에 존재하는 구조부재 즉, Column클래스, Beam클래스, Support클래스등을 포함하고 있으며, 이들 클래스를 생성하고 관리하는 StructuralSystem클래스를 가지고 있는 데 다음은 그것의 일부를 보여준다.

```
class StructuralSystem{
public:
    StructuralSystem();
    Column& seek_column( const char* va ) const;
    Beam& seek_beam( const char* va ) const;
    Support& seek_support( const char* va ) const;
    ConcentrateLoad& seek_con_load( const char* va ) const;
protected:
    Column** columns;
    Beam** beams;
    Support** supports;
    ConcentrateLoad** con_loads;
};
```

실세계에 존재하는 기둥이나 보와 같이 StructuralSystem내에서 생성된 Column클래스나 Beam클래스에서 인스턴스를 생성하고, 연결하고, 구조해석을 수행하고, 부재가 설계 기준식에 만족하는 與否를 檢討한다. 다음은 이를 수행하는 일련의 분장을 보여 주고 있다.

```
a_column = new Column();
a_beam = new Beam();
a_support = new Support();
a_load = new ConcentrateLoad();
a_column->connect_to ( a_beam , 400 );
a_column->supported ( a_support );
a_column->put_con_load( a_load );
a_column->act();
cout << a_column->check_all();
cout << a_column->report_calculation();
```

- AISAD의 개발

使用者가 프로그램과 相互作用에 의해서 데이터를 생성시키는 것은 통합 구조설계 시스템과 같이 데이터의 適合性 與否를 확인하고 수정하고 하는 일련의 반복작업을 수행하는 시스템에서 절대적으로 필요하다. AISAD는 구조물의 구조부재를 OMISAD내에 생성하고, OMISAD의 구조물 정보를 관찰하고, 변경시키고, 구조해석을 위한 메시지를 전달하고, 구조해석이 완료된 OMISAD의 부재가

기준식을 만족하는 與否를 출력한다. 부재가 기준식을 만족하지 못할 때, 사용자는 AISAD를 통하여 OMISAD내의 부재 데이터를 변경하고, 다시 구조해석을 수행하게 된다.

이와같은 수행은 기존의 수행방법이 구조물 데이터 베이스에서 각 부재의 구조해석을 위한 데이터를 抽出하고, 이에 대한 강성행렬을 모든 구조부재마다 구하고, 이를 구조전체에 대한 강성행렬에 조합하고 연립방정식을 구하는 일련의 작업을 수행하는 것과는 다르다. 구조물 객체모델인 OMISAD는 부재가 변경이 되면 그 부재의 강성행렬만을 다시 계산한다. 이때 다른 부재는 관계성에 의해 정의되어 있기 때문에 강성행렬등의 다른 속성에 영향을 받지 않는다.

4.3 실행

본 실행에에서는 5층 규모의 철골 라아멘구조를 AISAD를 사용하여 데이터를 생성하고, 구조해석을 수행하여 AISAD에서 객체지향적인 해석방법의 유용성을 실험하였다. OMISAD는 Node클래스에서 적합한 변위를 구하기 위하여 Element클래스와 연결된 각 Node클래스를 반복하여 追跡한다. 이때 Node클래스에서 사용되는 방법은 연립방정식의 수치해법인 Gauss-Seidel법을 사용한다. 그래서, 실행실험에서는 Gauss-Seidel법의 收斂에 영향을 미치는 relaxation factor와 해의 正確度를 조절하면서 Node클래스가 적당한 변위를 구하고, 부재내력을 구하는 시간을 測定하였다.

이 구조물의 절점의 수는 102개이고, 요소의 수는 197개이다. 그리고, 각 절점의 자유도는 6이다. <표1>의 수행결과를 볼때, 이 구조물에서는 relaxation factor가 1.9일때가 수렴에 소요되는 시간이 가장 짧았고, 초기에 전체 구조물의 미지의 변위를 0.0으로 가정하고, 구조해석을 수행한 시간보다 구조해석을 수행한 후 각 부재가 적당한 변위를 가지고 있는 상태에서 구조물의 일부 부재를 수정하였을 때는 최초의 구조해석에 소요되는 시간에 비해 1.6~2.6배가 빨라지는 것을 알 수 있다. 부재를 추가했을 때도 이와같은 현상이 나타난다. 이것은 변경되거나 추가된 부재의 전체 구조물에 대한 기여도가 크면 소요시간이 많이 걸리고, 적으면 적게 나타나는 것으로 판단된다.

번호	Relaxation Factor	Error (%)	최초의 해석시 소요시간 (sec)	C01 부재의 변경후 해석시 소요시간(sec) H300x300x10x15 H350x350x12x19	부재추가후 해석시 소요시간(sec) H350x350x12x19	C02, C04, C05부재의 변경후 해석시 소요시간(sec) H300x300x10x15 H350x350x12x19
1	1.4	0.01	1615	657 (2.46)	-	-
2	1.7	0.01	815	370 (2.20)	363 (2.25)	-
3	1.8	0.01	680	256 (2.66)	348 (1.95)	-
4	1.9	0.01	282	173 (1.63)	128 (2.20)	-
5	1.9	0.01	288	174 (1.66)	131 (2.20)	109 (2.64)

<표1> AISAD의 실행시간 비교

이와같은 현상은 구조물 객체모델이 캡슐화돼 있고 상호 독립적이어서 구조해석이 완료되면 계산된 변위를 항상 가지고 있기 때문이다. 이와달리 일반적인 절차중심의 구조해석에는 재해석시 강성행렬을 재조합하기 때문에 변경시마다 항상 일정한 시간을 소요하게 된다. 또, 각 데이터로의 접근이 용이하기 때문에 부재의 적합성을 구조해석이 수행되면 바로 확인할 수 있을 뿐 아니라, 變更이 쉽게 이루어 졌다.

5. 결론

본 연구에서 제안한 구조해석 객체모델과 구조설계 객체모델을 통합 구조설계 시스템에 도입하여 다음과 같은 결론을 얻었다.

1. 구조해석 객체모델과 구조설계 객체모델이 의미표현력이 높고, 캡슐화되어 있기 때문에, 미래의 변경에 재활용성이 높고, 유한요소해석 프로그램을 전문으로 프로그램을 하는 전문가가 아니더라도 쉽게 인식할 수 있는 개념으로 모듈화되어 있어, 본 연구 수행중의 응용모델인 AISAD의 개발시 單純性과 便宜性을 제공하였다.

2. 구조해석 객체모델은 보통의 유한요소해석 프로그램이 처음부터 각 부재가 변경될 때마다 다시 계산을 시작하는 것과 달리 초기 계산해를 가지고 있기 때문에 반복적인 變更을 요구하는 구조설계 작업에서 상대적으로 시간을 절약할 수 있었다.
3. 구조물의 각 데이터를 실세계의 구조물과 유사하게 모델했고, 데이터 모델이 구조적인 거동을 模寫했기 때문에 시공되어 지는 것과 유사하게 데이터 모델을 표현할 수 있어, 통합구조설계 시스템에서 일괄성 있게 시공에 관련된 데이터도 처리할 수 있었다.
4. 절점변위를 구할 때에 반복해법을 사용하므로 구조해석에 적합한 해를 구하는 데, 많은 시간을 요구 하였다. 또, 실행시간 중에 C++로 생성된 코드가 메모리의 動的連結을 수행하므로 실행속도의 低下의 원인이 되었다. 현재, 이 모델을 실무에 적용하기 위해서는 모델에서의 반복해법을 실행할 때에 실행 알고리즘의 속도향상을 위한 最適化 방법이 연구되어야 한다.

통합 구조설계 시스템에서 구조물 해석과 데이터 처리를 위하여 구조해석 객체모델과 구조설계 객체모델을 사용하는 것은 다음과 같은 기대효과를 가진다. 첫째, 컴퓨터 하드웨어의 飛躍的인 발전으로 실행속도를 극복할 수 있으므로 통합 구조설계 시스템과 같은 소프트웨어의 생산성을 향상시킬 것이다. 둘째, 건축의 여러분야의 데이터를 일괄되고 유연하게 처리한 필요가 있는 곳에 標準化의 방법론을 제공할 것이다.

참고 문헌

- [1] 김홍국, 이병해, "철골 구조물의 통합설계 시스템", 대한건축학회 논문집, 제8권12호, 1992
- [2] B.W.R Forde, R.O.Foschi and S.F.Stiemer, Object-oriented finite element analysis, Comput. Struct. 35, 355-374, 1990
- [3] G. Booch, Object-oriented Design with application, The Benjamin/Cummings, 1991
- [4] 천진호, 김홍국, 이병해, "객체지향 데이터베이스를 도입한 통합구조설계 시스템의 구축", 대한건축학회 논문집, 제9권9호, 1993
- [5] R.I.Mackie, Object oriented programming of the finite element method, Int.J.Numer.Meth.Engng. 35, 425-436, 1992
- [6] T.Zimmermann, Y.Dubois-Pelerin and P.Bomme, Object-oriented finite element programming: I. Governing principles, Comput.Methods.Appl.Mech.Engng, 98, 291-303, 1992
- [7] S.P.Scholz, Elements of an object-oriented FEM++ program in C++, Comput.Struct. 43, 517-529, 1992
- [8] J.Rumbaugh, M.Blaga, W.Premertani, F.Eddy and W.Lorensen, Object-oriented Modeling and Design, Prentice-Hall, 1991
- [9] W.McGuire and R.H.Gallagher, Matrix structural analysis, Wiley, 1979
- [10] 崔暢根, "有限要素解析", 集文堂, 1992
- [11] C.A.Brebbia and A.J.Ferrante, Computational methods for the solution of engineering problems, Pentech press, 1979
- [12] 金尙植, "建築鐵骨構造", 文運堂, 1987
- [13] B.Stroustrup, The C++ Programming Language, Addison-Wesley, 1991
- [14] D.A.Young, Object-oriented programming with C++ and OSF/Motif, Prentice-Hall, 1992
- [15] A.A.Williams, DOS and Windows protected mode: programming with DOS extenders, Addison-Wesley, 1993