

일관 구조설계 시스템 구축에 있어서  
객체지향 데이터베이스의 도입

Object-Oriented Database Application  
for An Integrated Structural Design Process

천 진호\* 김 홍국\*\* 이 병해\*\*\*  
Cheun, Jin-Ho Kim, Hong-Kook Lee, Byung-Hai

ABSTRACT

An integrated structural design system can define and process complex design information that occurs in each of the design stages for flexible cooperation. It can also reduce human error on sequential design steps; preliminary design, analysis and detailed design. Therefore, it's very important to have consistent semantic expression and procedure for structural design information that has complex relationships. In this study, we introduce the object-oriented concepts and object-oriented database technique that provides high level semantic expression in order to develop an integrated structural design system.

1 장 서 론

1.1 연구배경과 목적

구조물의 설계는 여러 제한조건을 만족하면서 시공시 최소의 비용과 최대의 안전성을 유지하게 하는 일련의 작업이다. 구조설계의 과정은 제시된 조건 및 제한사항을 고려하여 예비설계, 해석 및 상세설계를 거치게 된다. 이런 설계 과정에서 최적화된 구조설계를 얻기 위해서는 반복된 작업과 다양한 선택사항을 거쳐야 한다. 그 규모에 따라 설계단계 사이에는 다양한 상관 관계를 가지며 각 설계정보 사이에도 복잡한 관련성을 포함하고 있다. 그러므로 각 설계단계 사이의 긴밀하고 효율적인 협력은 반복되는 설계과정을 최소화시킬 수 있는 방법이다.

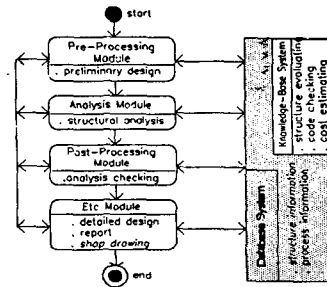
실제적으로 각 설계단계는 긴밀한 협조가 이루어지지 못하고 있으며, 또한 시공도면 작성을 시공 기일 내에 맞추기 위해서는 항상 시간에 쫓기게 된다. 더구나 구조설계하는 과정에서 관련 자료들이 수정되어 처음부터 재 설계하여야 하는 상황이 발생하게 된다. 이런 현실에서는 구조 설계자가 구조에 대한 기술과 창의력을 구조설계 단계에 충분히 발휘하는 것은 기대하기 어렵다.

구조설계에 대한 이런 문제점을 해결하고자 예비설계에서 시공도면 작성까지의 전 구조설계 작업을 일관되게 처리하는 일관 구조설계시스템에 대한 관심이 높아지고 있다. [1, 2] 즉 일관 구조설계시스템은 구조설계 문제에 연관된 설계정보를 적절히 구성하고 관리하며 설계정보 간의 상호 작용을 수행한다. 이런 일관 구조설계시스템 도입을 통하여 구조설계자는 다음과 같은 이득을 얻게 된다.

- . 각 설계단계 사이의 긴밀한 협력이 가능
- . 창의적인 문제에 집중할 수 있게 함
- . 전체적인 설계작업을 관리하고 판단할 수 있음
- . 설계작업 시 발생할 수 있는 인간적 오류를 줄임
- . 구조설계의 질 증진 및 시간 단축

\* 정회원, 한양대 건축공학과 대학원  
\*\* 정회원, 한양대 건축공학과 박사과정  
\*\*\*정회원, 한양대 건축공학과 교수

각 설계단계를 유연하게 연관시키기 위해서는 설계단계에서 발생하는 데이터를 적절히 변환하고 임의의 경험적인 판단을 내리는 등 많은 노력이 요구된다. 일관 구조설계시스템이 이런 다양하고 복잡한 작업을 수행하기 위해서는 효율적 데이터처리를 위한 데이터베이스 시스템, 경험적 판단을 위한 지식기반시스템 및 컴퓨터 그래픽 지원시스템이 긴밀하게 연관되어야 한다. [3] <그림 1>은 이런 일관 구조설계시스템의 상위 개념도이다. 본 그림에서와 같이 예비설계에서 상세도면 작성까지의 일련의 작업은 데이터베이스시스템 및 부분적 지식기반시스템 [18]과 연계되어 처리된다.



<그림 1> 일관 구조설계 시스템 구조

효율적인 일관 구조설계시스템은 설계정보를 일관되게 표현하는 능력을 가져야하며 설계정보를 능률적으로 처리할 수 있어야 한다. 이러한 설계정보는 데이터베이스 내에 저장되어 처리된다. <그림 1>과 같이 각 응용설계 모듈은 데이터베이스와 상호 밀접한 연관관계를 유지하면서 데이터베이스시스템을 중심으로 일련의 구조설계 작업을 수행한다. 이와같이 데이터베이스 기술은 각 설계작업을 통합하기 위한 도구이다. 그러므로 일관 구조설계시스템의 초과는 효율적인 데이터베이스시스템 구축에 달려있다.

데이터베이스는 정보데이터 사이의 關係性

(Relationship)을 정의하는 임의 구조로서 구축된다. 이런 구조를 데이터모델[4]이라 한다. 일반적으로 일관 구조설계시스템 개발을 위한 데이터베이스 기술은 주로 관계형 데이터 모델에 기초한 관계형 데이터베이스에 집중되었다. 관계형 데이터베이스와 데이터모델은 결합력, 단순성, 사용의 용이성 등의 특성 때문에 일반 사무 분야에서 널리 쓰여왔다. 그러나 공학 분야에서는 작업의 복잡성 중대에 따른 그 이상의 특성을 요구하게 되었다. 그 요구 사항 중 중요한 한가지는 實世界(Real World) 대상의 意味를 컴퓨터 내에서 다루어지는 데이터로 적절히 표현하는 높은 意味表現力(Semantic Expression)을 요구하게 되었다. 이런 특성은 일관 구조설계시스템을 위한 데이터베이스에 필수적이다. 즉 일관 구조설계시스템을 위한 데이터베이스 관리시스템(DBMS)은 설계정보를 효과적으로 관리하고 복합적인 설계정보를 표현할 수 있는 높은 수준의 抽象化(Abstraction) 방법론을 제공해야 한다. 이런 특성을 지원하지 못하는 관계형 데이터베이스 도입 노력은, 기대에 미치지 못하고 매우 부분적인 정보 저장에 그치게 되었다.

최근 공학 분야에서 관심이 집중되는 객체지향 개념은 관계형 데이터베이스 도입시 야기된 문제점을 극복할 수 있는 해결책을 제시하였다. [5,6] 즉 객체지향 개념과 데이터베이스기술은 객체 속에 관련 데이터와 행위를 함께 묶음으로써 설계 정보와 같은 복잡한 구조를 가진 데이터 처리가 용이하며 關係性(Relationship)과 制限規則(Constraint)을 통한 설계정보의 높은 意味表現力을 제공해준다. [7-9] 즉 객체지향 개념의 사용에 따른 높은 수준의 抽象化 방법론은 實世界의 객체와 컴퓨터 시스템 내에서 표현된 객체사이의 意味적 차이를 줄이는 意味表現力을 제공한다. 이런 특징을 갖는 객체지향 개념은 일관 구조설계시스템을 위한 설계정보 표현과 처리에 용이하다.

본 연구에서는 일관 구조설계시스템 개발 중진을 위한 환경 제시를 최종 목표로 하며, 객체지향 개념 및 객체지향 데이터모델링 방법론을 응용한 다음 일련의 항목을 하고자 한다.

1. 구조설계 정보와 설계 과정을 일관되게 표현 및 처리할 수 있는 방법론 제시.
2. 이를 통한 각 설계 단계사이의 긴밀한 협력을 가능하게하는 Design Object Model 개발.
3. 일관 구조설계시스템을 위한 객체지향 데이터베이스 관리시스템 고찰 및 개발.
4. 객체지향 데이터베이스시스템 환경 하의 응용설계 모듈 고찰 및 개발.

## 1.2 연구 방법

본 연구를 위해 강 구조물의 구조설계 과정을 분석하고 정리하여 설계정보, 설계정보 사이의 關係性 및 意味制限(Semantic Constraint)을 추출하였다. 이렇게 얻은 정보 데이터를 객체지향 개념과 모델링 방법론에 기초하여 체계적으로 객체화 시키고 모델링하여 Design Object Model을 작성하였다. 본 모델은 예비설계, 구조해석, 부재선정의 설계단계를 그 범위로 하며, 일관 구조설계시스템을 위한 객체지향 데이터베이스의 스키마(Schema)모델이 된다. 본 연구에서 개발된 객체지향 데이터베이스시스템은 ODAMIS로써 이는 Design Object Model을 구현하기 위해 Smalltalk 객체지향 프로그래밍 언어를 기반으로 개발되었다. 본 시스템은 구조설계 정보의 意味모델링에 대한 意味制限과 자동처리작업(Trigger)에 역점을 두었다. ODAMIS를 통하여 강 구조물의 형상모델링, 해석 및 수정 과정에서 발생할 수 있는 여러 설계작업 상황을 객체지향 데이터베이스 내에 구현

하여 일관되게 데이터를 정의하고 처리한다. 한편 본 연구에서 개발된 ORATUS는 ODAMIS를 직접 제어하는 응용설계 모듈이다.

## 2 장 객체지향 개념과 구조설계 정보

객체지향 개념의 근원은 처음 클래스 개념을 도입한 SIMULA(1966)라는 시뮬레이션 언어에서 찾을 수 있으며 프로그래밍 언어, 인공지능, 데이터베이스 등 여러 분야에서 새로운 요구에 의해 개발, 발전된 개념이다. 객체지향 개념의 가장 큰 특징은 實世界에 존재하는 객체를 쉽고 정확하게 표현하는 능력으로 이는 여러 공학 분야에서 그 적용성이 인정되고 있다. 건축, 토목 분야에서도 통합화를 위한 공학 데이터의 표현 및 처리와 구조해석 프로그램 개발 분야에서 그 적용성 연구가 활발히 진행되고 있다. [10-16]

### 2.1 객체(Object)

實世界(Real World)에 존재하는 모든 무형, 유형 의 個體(Entity)를 객체라고 한다. 한편 시스템 내에서의 객체는 實世界의 이런 존재의 모방을 일컫는다. 객체에는 그 존재의 성질을 나타내는 특성이 포함된다. 즉 객체는 데이터와 이에 연관된 Procedure를 함께 묶은 개념이다. 여기서 데이터와 Procedure를 각각 屬性(Attribute)과 메소드(Method)라고 부른다. 즉 모든 객체는 屬性과 메소드에 의해 그 상태(State)와 행동양식(Behavior)이 정의된다. 한편 객체지향 시스템 내에서, 각 객체는 유일하게 존재한다. 이를 객체의 Identity라 한다. 이런 객체와 Identity개념은 구조설계를 위한 높은 수준의 정보 데이터 표현력을 가능하게 한다.

### 2.2 클래스(Class)와 인스턴스(Instance)

공통된 형태의 屬性과 메소드를 가지는 객체들은 클래스 안으로 그룹화 된다. 클래스의 정의는 기존의 데이터베이스 관리시스템의 스키마와 유사하다. 클래스는 구조설계 같은 응용분야에서 중요한 특성을 취하고 나머지는 무시하는 抽象化 개념이다. 각각의 구조설계 정보는 클래스 내로 정의 될 수 있다. 클래스는 그들 자신의 행위를 통제할 수 있다는 점에서 스키마와 다르다. 하나의 클래스에 속하는 유사한 객체들은 그 클래스의 인스턴스가 된다. 인스턴스 객체의 상태는 그의 인스턴스 변수에 의해 구성되며, 인스턴스의 행위는 그가 속하는 클래스의 메소드에 의하여 제어된다.

### 2.3 메시지(Message)와 메소드(Method)

객체들은 서로 메시지를 주고 받으면서 상호 작용을 한다. 한편 메소드는 클래스와 인스턴스에게 메시지를 보냈을 때 그에 대한 응답으로 일정한 작업을 수행하는 Procedure이다. 객체에 메시지를 보냈을 때 그 객체가 그의 메소드를 부르는 행위를 메시지 전달(Message Passing)이라 부른다. 이는 객체지향 시스템에서의 유일한 의사소통 수단으로 객체는 메시지 전달을 통해 다른 객체와 협력하여 구조설계 같은 소기의 목적을 수행하게 된다.

### 2.4 相續性(Inheritance)과 클래스 계층 구조

클래스는 표현상 편의를 위해 계층적으로 구성된다. 새로 생성되는 클래스는 上位클래스(Superclass)의 下位클래스(Subclass)로 만들어지며, 上位클래스의 특성을 공유하게 된다. 上位클래스의 메소드와 屬性은 相續性에 의해 下位클래스에

서도 이용할 수 있게 된다. 클래스계층은 一般化(Generalization)와 詳細化(Specialization)의 意味關係性(Semantic Relationship)으로 표현된다. 즉 上位클래스는 下位클래스의 一般化이며, 下位클래스는 上位클래스의 詳細化이다. 上位클래스는 넓은 의미의 정보를 정의하고 下位클래스는 上位클래스의 모든 정의를 합병, 相續 받고 자신의 특성을 추가시킨다. 이렇게 一般化 관계는 유사한 설계객체를 한 단위로 통합 관리하게 하는 利點이 있다.

## 2.5 객체지향 데이터모델

모델은 實世界에 존재하는 대상의 意味를 抽象化하는 작업이다. 즉 데이터모델은 현실 세계에 존재하는 個體 意味(Entity Semantic), 個體들 사이의 關係性 및 制限規則을 통하여 데이터 정보를 기술하기 위한 개념적인 도구이다. 이런 데이터모델은 데이터베이스의 내용과 意味를 결정한다. 즉 데이터베이스시스템은 데이터모델을 구현한 것이라 할 수 있다. 구조설계 분야에서 데이터모델은 설계정보 데이터 구조를 정의하고 그 설계정보 데이터사이의 意味制限을 설정하며, 그 행위를 기술하는 등 이런 일련의 작업에 대한 일정한 규칙을 정의하는 것이다. 따라서 다른 규칙의 데이터모델은 다른 조합의 데이터 모델로 생성되어진다.

객체지향 데이터모델은 기본모델(Relational, Network, Hierarchical Data Models)에 비해 實世界 意味(Real World Semantic)를 표현하는데 利點이 있다. [17, 20] 객체지향 데이터모델링 기술은 객체로써 구조설계를 생각하고 표현하고 제어한다. 즉 목적하는 응용분야의 객체와 객체 Identity, 그들간의 關係性, 制限規則 및 屬性, 행동방식을 객체지향 모델링방법론에 입각하여 구성하는 것이다. 객체지향 데이터모델에서 기본단위는 클래스이다. 객체지향 데이터모델은 인스턴스 화, 클래스, 一般化, 詳細化, Aggregation 과 Association 關係性 같은 높은 수준의 抽象化 방법론을 사용하여 복잡한 實世界의 데이터와 그 關係性을 표현한다.

객체지향 데이터모델에서 Association과 Aggregation은 가장 중요한 개념 중 하나로 객체간의 關係性을 정의한다. Association 關係性은 객체사이의 의지 관계(Dependent Relationship)를 표현하며 Aggregation은 'isPartOf'와 같은 한 객체가 다른 객체를 포함하는 關係性을 표현한다. Aggregation 關係性은 특히 여러 공학 분야에 풍부히 존재한다. 예를 들어, 한 Frame은 기둥, 보, 슬라브 등의 기본 구조적 요소의 집합체이다. 이와같이 Aggregation은 설계정보 표현에 유용하다.

## 3 장 Design Object Model

### 3.1 구조설계정보

Design Object Model은 본 연구에서 제시된 객체지향 데이터베이스시스템의 물리적 스키마 구성을 위한 구조설계 정보 데이터모델이다. 본 모델은 객체지향 데이터모델로써 구조설계 정보를 계층적이고 일관성 있게 표현, 정의한다.

구조설계는 구조물의 기본 설계정보를 토대로 관련 자료를 수집하여 구조물의 형상을 결정하는 예비설계 단계, 구성된 형상모델에 대한 구조해석 작업을 거쳐, 구조물의 안전성 검사 및 부재설계, 접합부설계, 기초설계와 같은 상세설계를 행한다. 구조설계자는 이런 일련의 구조설계 작업에서 다양한 설계 정보데이터를 다루고 그 關係性에 의해 일정한 설계작업을 행한다. 예로써 일련의 강 구조 설계작업 시 처리되는 설계정보에는

- . 구조물 형상 데이터와 시스템 구성 데이터
- . 구조물 부재 재료
- . 각 부재간의 연결 상태
- . 각 종류별 하중, 하중조합 상태
- . 각 요소에 대한 최대 모멘트, 전단력
- . 각 절점의 처짐, 부재 선정, Code Check

등이 포함될 수 있다. 이런 설계정보는 본 Design Object Model에서 설계객체로 적절히 모델링 된다.

### 3.2 설계객체(Design Object) 모델링

#### 1) 클래스 내로 설계정보 객체화

구조설계 정보는 객체로 정의된다. 즉 부재, 하중, Frame 등이 객체로써 다루어질 수 있다. 설계객체는 클래스를 통하여 구성되며 그 특성과 행동 방식이 정의된다.

#### 2) 屬性을 통한 설계객체 특성 정의

각 설계객체의 특성은 屬性과 메소드를 통하여 정의된다. 예를 들어 기둥 객체에 대한 부재 길이나 최대 모우멘트 값 등은 屬性으로 정의될 수 있다.

#### 3) 인스턴스를 통한 설계객체 처리

클래스 속으로 객체화된 설계객체들은 인스턴스를 통하여 구체적인 정보가 처리된다. 즉 인스턴스는 설계정보의 특성을 정의한 클래스에 대한 구체적인 實體가 된다.

#### 4) 相續性을 이용한 설계객체 표현

유사한 설계정보는 相續性을 통하여 체계적으로 구성할 수 있다. 相續性에 의한 설계객체 표현은 유사한 설계 정보를 단순하고 통합적으로 표현 관리하는 방법론을 제공한다.

#### 5) 關係性을 통한 설계객체 사이의 意味性 정의

설계객체 사이에는 복잡한 意味關係性이 존재한다. 이런 關係性은 일종의 정보 추적 경로가 된다. 즉 關係性에 의해 정의된 경로를 통하여 설계객체는 상호 작용을 하며 이런 일련의 상호 작용으로 구조 설계 작업이 수행되어 진다. 關係性에는 크게 Aggregation과 Association 關係性이 존재한다.

#### 6) 意味制限(Semantic Constraint)을 통한 설계객체 관리

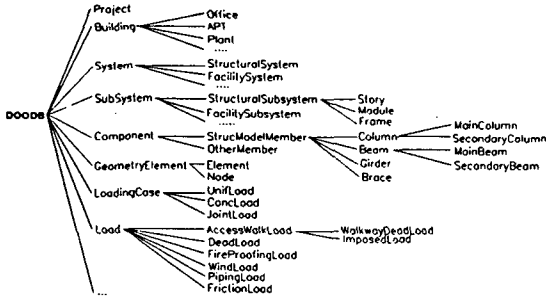
설계객체나 객체 사이에는 다양한 制限規則이 존재할 수 있다. 이런 制限規則은 意味制限을 통하여 정의되고 유지된다. 意味制限은 설계객체 모델의 意味保全性(Semantics Integrity)을 유지, 설계 데이터에 대한 규정치를 정의 및 설계객체에 대한 자동작업 처리를 정의하고 수행한다. 본 意味制限은 ODAMIS에서 Rule Trigger, Rule Constraint, Method 들 통하여 구현된다.

### 3.3 Design Object Model 구조

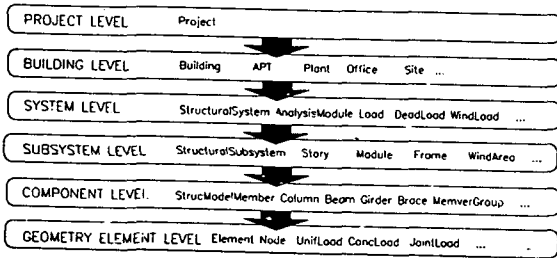
Design Object Model은 실제 구조설계 작업과정을 분석하여 추출한 설계객체와 그 意味關係性(Semantic Relationship)에 기초하여 구성된 객체지향 데이터모델이다. 본 모델의 구성은 Top-Down 접근법, Botton-Up 접근법으로 전체 설계객체 단계를 구성하였으며 각 설계객체와 객체사이의 關係性 표현을 위해 객체지향 개념과 모델링방법론에 그 이론적 기초를 두고 있다. 본 모델을 통하여 표현되는 주요 설계정보는 다음과 같다.

- . 단계별 구조물 정보와 關係性
- . 종류별 하중 정보와 關係性

- . 설계객체 간의 意味制限
- . 각 하중 산출프로세스
- . 부재 재료 데이터베이스
- . 구조해석 용역값



<그림 2> Design Object 클래스 계층



<그림 3> 단계별 Design Object 클래스 구성

구조설계를 위한 Design Object Model에서, 설계정보는 몇 가지 抽象化(Abstraction) 단계로 표현된다. <그림 2> 는 설계정보를 객체화하고 相繼性에 의해 표현한 Design Object Model의 클래스 계층도이다. <그림 3> 과같이 본 클래스는 다시 구조적 설계객체를 기준으로 여섯 단계로 Top-Down배치된다. 즉, 계층별 설계 클래스들은 相繼性을 통한 수평적 關係性을 형성하며, 단계별 설계 클래스들은 Association과 Aggregation을 통한 수직적 關係性을 형성한다.

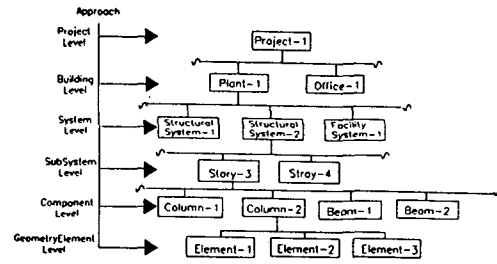
### 3.4 Design Object Model을 통한 설계객체 접근

Top-Down 및 Bottom-Up 방법론에 기초한 단계별, 계층별 설계객체 구성은 실질 설계정보 처리 시 사용되는 자연스러운 방법이다. 본 Design Object Model에서, 설계객체의 관리의 이런 구성 경로를 따라 제어된다. <그림 4> 는 구조적 설계객체의 Top-Down 경로 구성도이다. <그림 4> 내 화살표와 같이 'Project' 설계객체에서부터 'Element'까지 단계별, 계층별로 다양한 접근법이 가능하다.

다음은 해석 요소 Element2을 얻기 위한 한 접근 경로이다.

Project-1 → Plant-1 → StructuralSystem-2 → Story-3 → Column-2 → Element-2

한편 비 구조적인 설계객체(절점, 단위하중) 접근 경로는 구조적 설계객체(Floor, Column, Element)에 접근한 후 그 설계객체에 關係性 경로를 통하여 도달하게 된다.



<그림 4> 설계객체 접근 경로

### 3.5 단계별 Design Object Model

#### 1) PROJECT 단계

PROJECT 단계는 본 모델의 최상위 단계이다. 설계 작업은 프로젝트 단위로 처리되며 <그림 5> 에서와 같이 한 프로젝트에는 여러 건물이 'contains' Aggregation 關係性으로 포함 관계를 형성한다. 또한 Project 클래스는 프로젝트의 정보를 저장 관리한다.

#### 2) BUILDING 단계

본 단계는 건물에 대한 구조설계자의 관점에서 정보를 처리한다. Building 클래스는 건물을 기능별 下位클래스로 詳細化(Specialization)시키고 한 건물에 포함되는 기능별 부시스템인 System 클래스와 'isPartOf' Aggregation 關係性을 형성한다.

#### 3) SYSTEM 단계

건물은 여러 기능별 시스템으로 구성된다. 본 단계는 이런 시스템을 정의하며 연관된 설계정보를 처리한다. 본 모델에서는 이런 시스템 중 구조설계에 관련된 Structural System 특히 Steel Frame에 관련된 정보를 그 범위로 한다. 본 StructuralSystem 클래스는 구조 해석되는 단위 구조물을 정의하고 그에 관련된 설계 하중을 처리한다. <그림 7> 은 이런 關係性을 보여준다.

#### 4) SUBSYSTEM 단계

구조물의 시스템은 여러 부시스템의 모임일 수 있다. SUBSYSTEM 단계는 이런 부시스템의 구조적, 기능적 관련된 정보를 저장 관리한다. <그림 5> 는 SUBSYSTEM 단계에서의 설계객체와 그 객체사이의 關係性을 보여 준다.

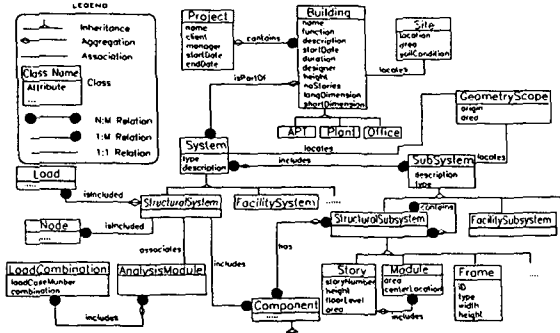
#### 5) COMPONENT 단계

본 단계는 부시스템을 구성하는 단위부재의 설계 정보를 정의한다. 각 단위부재는 기능상 기둥, 보 등으로 詳細化되며 부재재료의 屬性이 정의된다. 구조 해석상 각 부재는 하나 이상의 요소(Element)와 둘 이상의 절점을 포함한다. <그림 6> 은 COMPONENT 단계의 설계객체 정의와 관련성을 보여준다.

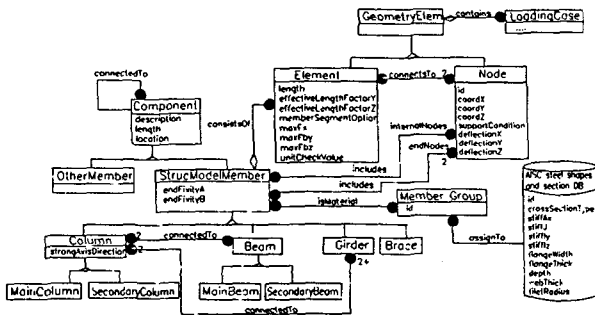
#### 6) GEOMETRY ELEMENT 단계

본 단계는 구조물의 해석 시 요구되는 설계객체인 요소(Element), 절점(Node) 및 그 곳에 작용되는 단위 하중을 정의하고 관리한다. 기둥, 보같은 실질 부재는 구조 형상모델에서 하나 이상의 요소로 구성된다. 또한 각 요소는 양단에 두개의 절점을 소유하게 된다. 이런 關係性은 <그림 6> 에서와 같이 Element 클래스와 Member 클래스 사이의 'consistsOf' Aggregation 關係性, Element 클래스와 Node 클래스 사이의 'connectsTo' Association 關係性으로 표현된다. 요소와 절점을 함께 관리하는

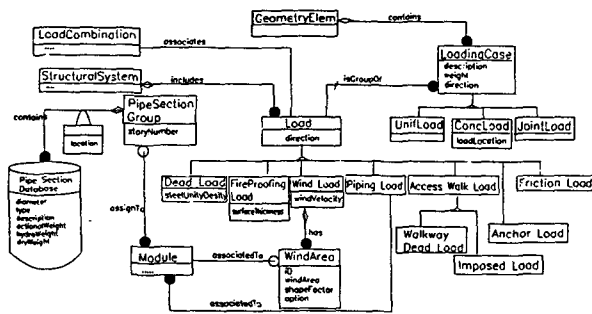
上位클래스인 'GeometryElem' 클래스는 구조물 전체의 하중을 산출 관리하는 'Load'와 그의 下位클래스에 의해 생성되는 단위하중을 포함하게 된다. 이는 'contains' Association 關係性으로 관리된다.



<그림 5> Design Object Model 1



<그림 6> Design Object Model 2



<그림 7> Design Object Model 3

#### 4 장 객체지향 개념의 실현화

본 연구에서 개발한 ODAMIS(Object-Oriented Database Management System for Integrated Structural Design)는 Design Object Model의 물리적 스키마 구현을 위해 개발된 실험적 객체지향 데이터베이스 관리 시스템(OODBMS)이며, ORATUS(Object-Triggered Application for Structural Design)는 ODAMIS로 구축된 설계객체를 직접 제어하는 응용설계 모듈이다. ODAMIS는 구조설계 정보와 意味(Semantic)를 저장 관리하고 설계 단계에서 수행되는 설계 작업을 적절히 구성하는 기능을 제공한다.

다. 즉 구조설계 정보와 구조설계 작업을 위한 意味 모델링, Rule Constraint, 그리고 Rule Trigger의 정의 및 관리를 주목적으로 한다.

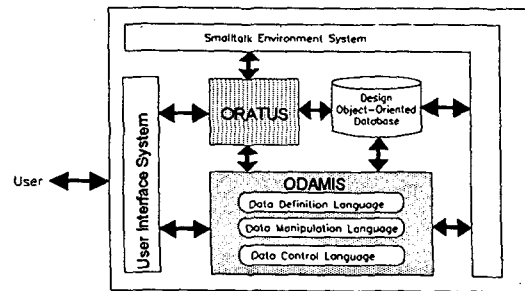
#### 4.1 본 시스템 개발 언어(Smalltalk)

Smalltalk [19]는 ODAMIS와 ORATUS 개발을 위한 기반 프로그래밍 언어로서 주요 객체지향 개념인 '객체', '클래스', '相繼性'을 가장 충실히 실현한 객체지향 언어이다. 이는 최초의 객체지향적 언어로서 언어로서뿐 아니라 사용자에게 윈도우 시스템을 기반으로 하는 프로그래밍 환경과 객체지향적 프로그래밍 패러디임(Paradigm)을 제공한다. 즉 기존 구조적 프로그래밍에 비해 프로그램 제작에 있어서 상당한 효율성이 있으며 유지 보수가 편리하다는 이점이 있다.

#### 4.2 시스템 특성

ODAMIS의 구성은 객체지향 개념과 데이터베이스 관리시스템 기술을 기반으로 하여 개발되었으며 일관 구조설계시스템에 맞게 적절히 보안되었다. 시스템 개발에 사용된 언어는 Smalltalk로써 본 언어로 객체지향 데이터베이스 관리시스템의 데이터 정의언어(Data Definition Language), 데이터 유지언어(Data Manipulation Language), 그리고 데이터 제어언어(Data Control Language)를 개발하였다. <그림 8>은 통합된 구성을 이루고 있는 본 시스템 구조를 보여준다. 즉 기존 Smalltalk Environment System과 데이터 정의언어(DDL), 데이터 유지언어(DML), 데이터 제어언어(DCL), 설계 객체지향 데이터베이스(DOODB), 그리고 응용설계 모듈인 ORATUS 및 사용자 인터페이스시스템(User Interface System)들이 같은 환경속에 위치하게 된다. 이와같이 동일한 환경에서 본 시스템이 구축됨으로 일관 구조설계시스템의 통합화에 근본적인 환경을 제공하게 된다.

본 시스템의 주요 기능은 자동처리되는 Rule Constraint, Rule Trigger 이다. 이는 각 인스턴스의 屬性값 및 객체형태(Object Type)에 대한 意味制限을 정의하고 유지하는데 그 유용성이 있다. 즉 이는 임의 설계정보 변경에 따른 연관된 설계정보의 수정 및 데이터의 제한 규정과 같은 설계정보의 자동적 관리에 적합한 도구이다. 한편 본 시스템의 意味 모델링 기능은 객체지향 개념의 가장 큰 의미를 가지는 것으로, 屬性과 關係性을 통하여 설계개체들 사이의 意味 모델을 합리적으로 구성할 수 있게 한다.



<그림 8> ODAMIS와 ORATUS의 구성

#### 4.3 ODAMIS 클래스 구조

<그림 9>에서와 같이 ODAMIS의 각 모듈들은 기존 클래스인 'Behavior'와 'Object'의 메소드 내에 구축된다. 즉 두 클래스내에 데이터베이스를 유지

관리할 수 있는 대부분의 모듈이 포함된다. 특히 'Rule' 클래스는 설계객체 클래스의 유지와 자동관리를 정의하는 클래스로써, 규정(Rules)을 표현하기 위한 객체 인스턴스를 생성한다. 설계객체의 데이터베이스는 'DOODB'클래스의 下位클래스로 스키마를 형성하고 그의 인스턴스를 만들어 저장 관리된다.

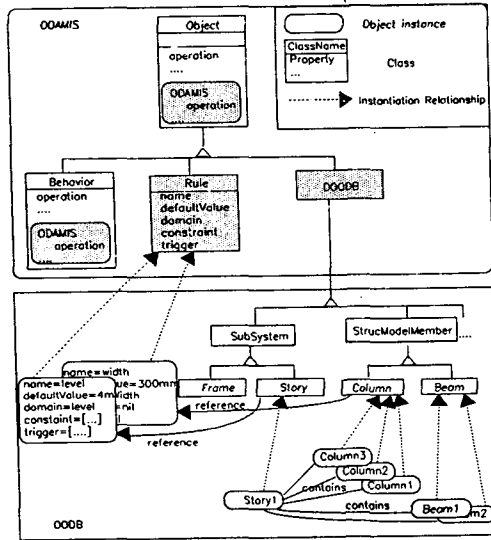
### 1) Rule 클래스 구조

설계정보 데이터베이스 구축 시, 각각의 정보를 저장하기 위해서 클래스를 구성하고 그의 屬性 및 關係性을 정의한다. 정의된 이런 클래스에서 생성된 인스턴스는 그 자신의 屬性과 關係性을 가지게된다. Rule 클래스는 이런 각각의 클래스에 포함된 모든 인스턴스들의 屬性과 關係性을 관리 할 수 있는 규칙을 설정하고 관리하는 편의를 제공한다.

<그림 9> 는 설계객체 데이터베이스와 Rule 클래스 및 그의 인스턴스와의 관계를 보여준다. Rule 인스턴스들은 屬性 이름(Property Name), 初期屬性값(Default Value), Rule Domain, Rule Constraint, Rule Trigger를 정의할 수 있는 슬롯(Slot)을 가지고 있다.

### 2) Rule Domain을 통한 屬性 및 關係性 구현

구조 설계자들이 다루는 데이터들은 그 구조가 단순한 수치값에서부터 상당히 복잡한 형의 데이터를 다루게 된다. 객체지향 데이터베이스 관리시스템은 구조설계 정보를 충분히 구축 관리할 수 있는 데이터형을 정의하고 관리할 수 있다. 이때 각 인스턴스 屬性에 들어가는 다양한 정보의 형태(Type)을 조사하는 기능을 Rule Domain이 처리하게 된다. 이런 기능은 다양한 종류의 데이터를 처리해야 하는 건물 정보 데이터베이스에서 적절치 못한 데이터를 넣는 사용자의 실수를 막을 수 있다.



<그림 9> OODAMIS 클래스 구조

### 3) Rule Constraint를 통한 屬性값 제어

Rule Constraint는 각 屬性에 정의되어 그 屬性값의 허용 여부를 체크한다. 즉, 그 屬性에 대한 허용 조건문을 서술하여 Rule Constraint 슬롯에 할당한다. 그러면 이 Rule Constraint는 屬性값이 입력될 때마다 그의 허용여부를 체크하게 된다. 이는 여러 규정치가 많은 설계정보 관리에 적합하다. 구조설계

자는 설계정보 처리 시 본 Rule Constraint를 통해 자신의 설계문제 해결에 더 집중할 수 있다.

### 4) Rule Trigger를 통한 屬性값 제어

Rule Trigger는 설계정보를 자동유지 관리하기 위한 도구이다. Rule Trigger도 Rule Constraint와 유사하게 각 屬性에 할당된 Rule 인스턴스 내 Rule Trigger 슬롯에 Trigger제어문을 정의한다. Rule Constraint와 다르게 이는 입력 屬性값이 Trigger 조건에 합당하면 그 정의문이 작동되는 메커니즘이다. Rule Trigger는 설계정보 관리와 연관된 설계 작업 자동실행을 위한 적절한 도구이다.

### 4.4 ODAMIS 지원 데이터베이스 언어

1. 데이터정의언어(DDL)
  - 클래스, 屬性 및 매스드 정의어
  - Rule Domain, 초기값 정의어
  - Rule Constraint, Rule Trigger 정의어
2. 데이터유지언어(DML)
  - 인스턴스 생성어, 屬性값 할당어
  - 설계객체 質疑(Query)와 소멸(Remove)어
3. 意味모델 정의언어
  - Association, Aggregation 정의어

### 4.5 구조 설계 작업을 위한 응용설계모듈(ORATUS)

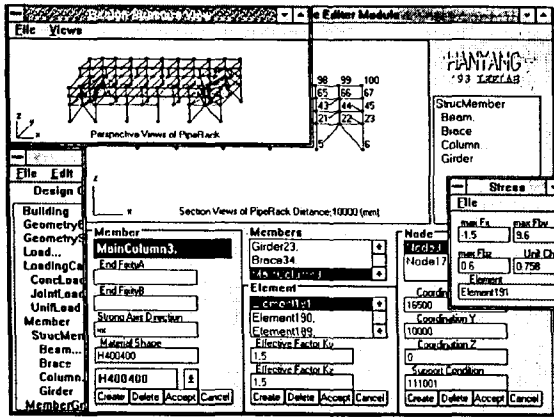
본 연구에서 개발된 ORATUS는 구조설계를 위한 응용설계모듈로써 ODAMIS에 의해 구현된 구조설계 객체를 직접 제어하여 목적하는 일련의 설계작업을 행한다. 이런 일련의 응용설계모듈(ORATUS)들은 일관 구조설계시스템을 구성하게 된다. ORATUS에 포함된 모듈들은 다음과 같다.

1. Structure Modeler: 구조설계 작업 시간의 상당 부분을 차지하는 구조물 형상모델링 지원 도구이다. 구축된 설계정보 데이터베이스(DOODB)에서 구조물 형상, 부재 단면 재료 및 구조적 시스템정보 등에 관련된 설계객체를 ODAMIS를 통하여 제어한다.
2. Analysis InputData Generator: 구축된 설계객체를 토대로 구조해석 프로그램[22]의 형식에 맞게 해석 입력 데이터를 자동 작성하여 주는 모듈이다.
3. Analysis Results Processor: 해석 결과에 의한 부재 검사를 행하고, 본 결과를 그래픽적으로 출력하여, 설계자에게 해석 결과값에 대한 이해를 돕는다.
4. Design Object Builder/Monitor: 설계객체를 데이터베이스 내에 용이하게 구현할 수 있는 사용자 인터페이스 모듈과 설계객체 데이터베이스 구축 상황을 화면 처리하여 구조 설계자에게 정연하게 제시하는 모듈이다.

그 외에 다음과 같은 모듈들이 포함될 수 있다.

- Material Estimator( 물량산출 모듈 )
- Shop Drawing( 시공도면 자동산출모듈)
- Report Generator( 보고서 및 구조계산서 자동작성모듈 )

<그림 10> 은 ORATUS로 작업하는 일부과정을 보여준다.



〈그림 10〉 ORATUS 실행화면

### 5 장 결론

객체지향 개념, 객체지향 모델링 및 데이터베이스 기술로 구조설계 문제를 생각하고 표현하는 것은 컴퓨터를 이용하는 응용 공학자들에게 새로운 개념이며 다시 학습해야 하는 부담을 줄 수 있다. 그러나 객체를 통한 설계정보의 표현은 기존방법론[21]에 의해 개발된 일관 구조설계시스템에 비교하여 더 자연스러운 표현법이며 다양한 요구와 변화에 대응할 수 있는 방법론이다.

본 연구에서는 일관 구조설계시스템 개발을 위한 객체지향 개념, 모델링, 프로그래밍, 그리고 데이터베이스 기술의 적용성을 고찰하여 보았다. 이런 일련의 작업과정을 통하여 다음의 결론을 얻었다.

1. 구조설계 정보를 표현하는데 있어서 높은 수준의 意味抽象化, 즉 實世界에 존재하는 설계정보에 대한 컴퓨터내로의 자연스러운 표현이 가능하며 일관되게 정보를 처리할 수 있게 되었다. 이는 각 설계단계 사이의 정확한 설계정보 전달에 필수적 요건이다.
2. 설계정보는 예비설계, 해석, 상세설계와 같이 각 설계단계가 진행됨에 따라 새로운 설계정보가 창출된다. 이에 객체지향 데이터베이스는 새롭게 추가 갱신되는 설계정보에 능동적으로 대응할 수 있었다.
3. 설계정보를 단계별로 표현, 처리함으로써 다양한 관점으로 설계정보에 접근, 관리할 수 있었다. 이는 데이터 관리와 응용설계모듈 작성에 단순성과 처리의 편의를 제공하여 주었다.
4. 응용설계 모듈은 일관 구조설계시스템의 중앙 데이터베이스를 제어한다. 그러므로 자신의 데이터 관리 기능을 갖출 필요가 없어 프로그래밍 단축과 일관된 설계정보 처리방식을 택할 수 있었다.
5. 응용설계모듈 개발 시 객체지향 데이터베이스 언어와 응용모듈 개발언어(예: Smalltalk)를 같은 종류의 언어로 구현함으로써 표현의 충돌을 막을 수 있었다.

### 참고 문헌

1. 김종현, 이병해, "철근 콘크리트 구조물의 설계 자동화", 대한건축학회 춘계 학술발표회 논문집, 제11권, 제 1호, 1991
2. 김홍국, 이병해, "철골 구조물의 통합 설계시스템", 전산구조공학회 추계 학술발표회 논문집,

- 제11권, 제2호, 1991
3. C.Y.Lin and L.T.Lin, "General Structural Data Model in an Integrated Structural Design System", The Fourth International Conference on Computing in Civil and Building Engineering, pp 267, 1991
4. J.Martin, Managing The Database Environment
5. W.Kim, Introduction to Object-Oriented Databases, The MIT press, 1991
6. J.J.Kim and C.W.Ibbs, "Comparing Object-Oriented and Relational Data Models for Project Control", Journal of Computing in Civil Engineering, 1992
7. R.Gupta and E.Horowitz, Object-Oriented Databases with Applications to CASE, Networks and VLSI CAD, Prentice-Hall, 1991
8. J.J.Kim and C.W.Ibbs, "An Experimental Object-Oriented DBMS", TRB Paper, 1991
9. J.G.Hughes, Object-Oriented Databases, Prentice-Hall, 1991
10. R.Sause and G.H.Powell, "A Design Process Model for Computer Integrated Structural Engineering: Design Phases and Tasks", Engineering with Computers 7, pp.145-160, 1991
11. G.R.Miller, "An Object-Oriented Approach to Structural Analysis and Design", Computers & Structures Vol.40, No.1, pp.75-82, 1991
12. K.H.Law and M.K.Jouaneh, "Data Modeling for Building Design", COMPUTING IN CIVIL ENGINEERING, pp 21-36
13. H.C.Howard, J.A.Abdalla, and D.H.Douglas Phan, "A Primitive-Composite Approach for Structural Data Modelling", 1990
14. J.Madden and R.Sause, "A Design Product Model for Computer Integrated Structural Engineering", 8th Conference on Computing in Civil Engineering, 1992
15. Gregory L.Fenves, "Object Representations for Structural Analysis and Design", pp.502-511
16. K.H.Law, "A Formal Approach for Managing Building Design Information in a Sharabe Relational Framework", The Fourth International Conference on Computing in Civil and Building Engineering, pp 266, 1991
17. B.Bruegge, J.Blythe, J.Jackson and J.Shufelt, "Object-Oriented System Modeling with OMT", OOPSLA'92 Conference, pp 359-376, 1992
18. 김진수, 이병해, "철골 구조 접합부에 대한 전 문가 시스템의 적용", 대한건축학회 추계 학술발 표회 논문집, 제12권, 제1호, 1992
19. Smalltalk/V Windows Tutorial and Programming Handbook, DIGTALK Inc, 1991
20. J.Rumbaugh, M.Blaga, W.Premierlani, F.Eddy and W.Lorenson, Object-Oriented Modeling and Design, Prentice-Hall, 1991
21. E.Yourdon, Modern Structured Analysis, Prentice-Hall, 1989
22. SACS III User's Manual, Engineering Dynamics Inc.