

## 구조설계 통합 시스템에서 중앙 데이터베이스를 위한 데이터 모델에 관한 연구

### A Study On Product Data Model for Central Database in an Integrated System for Structural Design of Building

안 계 현\*      신 동 철\*\*      이 병 해\*\*\*  
Ahn, Kye-Hyun      Shin, Dong-Cheol      Lee, Byung-Hai

#### ABSTRACT

The purpose of this study is to propose data models for central database in integrated system for structural design building. In order to efficiently express data related to structure, I analyzed the structure design process and classified data considering design step. I used an object-oriented modeling methodology for logical data model and relational modeling for physical data model.

Based on this model, we will develop an integration system with several applications for structure design. Each application will communicate through the central database.

#### 1. 서 론

건축 구조설계 과정은 일반적으로 초기 구조설계, 구조해석, 부재설계 그리고 상세설계 등의 일련의 과정을 거치며 각 단위 과정에서 개별 시스템의 사용을 통하여 작업을 수행한다. 이 때 구조설계 관련 데이터들은 단위 작업에서 독립적으로 사용되는 것이 아니라 여타 단위 작업과 유기적으로 연결되어 있다. 선행 작업의 데이터들은 대개 후행 작업의 입력 데이터로서 사용된다. 이 과정에서 기존의 단위 작업에 국한된 구조설계 전산화는 불필요한 데이터의 중복과 작업의 중복이라는 문제가 발생하였다.

기존 구조설계 전산화에서 발생된 문제를 해결하기 위해서 데이터 공유의 필요성이 인식되었고, 그 일환으로 시스템 통합화가 연구되어왔다. 특히 철골 구조 설계 분야는 데이터의 정형화가 철근콘크리트 구조 설계보다 용이한 이유로 더욱 활발히 진행되고 있는 실정이다. 외국의 경우 CIMSteel 표준화<sup>(5)</sup>, COMBINE 프로젝트<sup>(11)</sup> 등 대규모 프로젝트가 진행되고 있으나, 현재 국내에는 이러한 시도가 미흡한 것이 사실이다. 이러한 이유로 철골 구조설계 시스템 통합화를 위한 연구를 하였다.

기존의 통합시스템에 관한 연구는 개발자가 각 구조설계 과정별 모듈을 개발하고 이를 통합하는 방식으로 하나의 시스템을 개발자가 모두 개발하는 방법을 시도하여왔다. 이러한 통합 방법은 개별 모듈과 데이터베이스 간 개념이 같으므로 데이터베이스의 안정성이 보장되고 데이터 교환도 자유로운 장점이 있으나 현업에서 이미 다양한 응용 프로그램들이 사용되고 있는 실정에 비추어 본다면 그 의미를 찾기 어렵다고 볼 수 있다. 또 개발 단계에 있어 기술적인 문제들의 한계에 부딪힐 수도 있다.<sup>(3)</sup> 이러한 문제점을 보완하기 위해 본 연구실에서는 구조설계 각 모듈은 기존의 상용 소프트웨어를 그대로 이용하고 각 소프트웨어 상호간의 데이터를 교환할 수 있는 중앙 데이터베이스 시스템을 개발함으로써 시스템 통합을 이루는 방법을 시도하였다. 이 방법은 개별 시스템의 데이터 개념과 중앙 데이터베이스의 데이터 개념이 일치하지 않을 수 있으므로 데이터베이스의 안정성이 떨어지고, 개별 시스템들이 변화하게 되면 데이터베이스의 스키마에 수정이 가해져야 하는 단점이 있다.<sup>(3)</sup> 그러나 이미 상업적으로 보편화되어 있는 응용 프로그램들을 사용함으로써 엔지니어가 응용프로그램을 다루는 데 있어서 능숙한 기술을 보유할 수 있고, 수년간 다져진 높은 질의 기술을 그대로

\* 한양대학교 건축공학과 석사과정

\*\* 한양대학교 건축공학과 박사과정, 경원전문대학 건축과 교수

\*\*\* 한양대학교 건축공학과 교수

사용할 수 있다는 장점에 비중을 두어 이 통합방법을 적용하고자 한다.

본 논문은 제시된 통합화 방법을 구현하는데 있어서 기반을 이루게 될 중앙 데이터베이스의 데이터 모델링과 데이터베이스의 구현에 연구 목적을 두고 있다. 통합 시스템에서 데이터베이스는 데이터 공유를 위한 데이터 관리장치로서 개별 시스템이 요구하는 데이터를 제공하고 생성된 데이터를 저장하는 역할을 하게 될 것이다. 구조설계와 관련된 데이터를 전반적으로 다루는 것이 데이터베이스이므로 적절한 데이터 모델의 제시로 데이터베이스의 효율을 높일 수 있어야 하겠다.

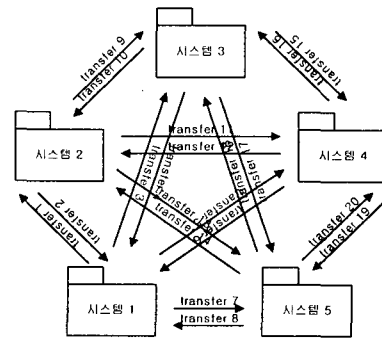
연구를 위해서 먼저 철골 구조물을 몇 단계의 구조적 위계로 나누어 물리적인 구조물 기본 정보를 나타내는 데이터 모델을 제시하였다. 또 철골 구조설계 작업과정을 따라 초기 구조설계 단계, 구조 해석 단계, 부재설계 단계, 접합부 설계 단계, 구조 도면화 단계로 나누어 각 단위과정에서 사용되는 데이터를 중심으로 데이터 모델을 하였다. 각 단위 작업에서 사용되는 응용 프로그램은 초기 구조설계 작업, 구조 해석 및 부재설계 작업은 GT STRUDL<sup>(9)</sup> 프로그램을, 접합부 설계 작업은 본 연구실에서 개발된 프로그램을, 도면화 작업은 X-Steel 프로그램<sup>(10)</sup>을 이용하여 통합화를 시도하였다.

구조 설계 데이터 모델링은 객체지향 모델링 방법론을 적용하였으며, 최근 각 모델링 언어 통합의 일환으로 제안된 UML(Unified Modeling Language)<sup>(6)</sup>을 이용하여 모델을 표현하였다. 데이터 모델링을 객체지향 기법을 이용한 것과는 달리 데이터베이스는 현재 가장 많이 사용되고 있고 데이터 관리 능력이 우수한 관계형 데이터베이스인 Oracle7.3<sup>(8)</sup>을 기반으로 구축하였고, 데이터베이스 스키마의 정의는 관계형 데이터베이스 표준어인 SQL(Standard Query Language)을 사용하였다.

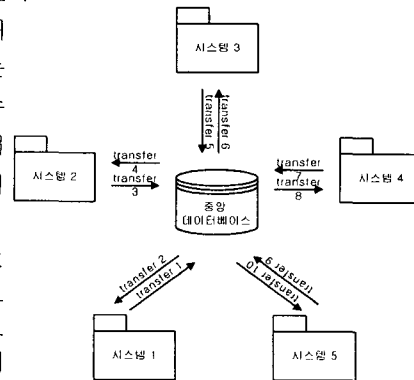
## 2. 통합시스템에서 데이터 교환 방법-데이터베이스의 역할

개별 시스템간의 통합에서 가장 중요시되는 것은 시스템간의 데이터 교환 문제이다. 개별 시스템을 통합하려는 경우 각 시스템 개발사가 시스템 내부를 완전히 개방하지 않기 때문에 그 구조를 정확히 알 수는 없다. 따라서 완전한 데이터의 공유와 교환을 이루는 것은 어렵다. 본 연구에서 이점을 전제로 하고 있다. 시스템 내부를 공개하지 않기 때문에 시스템의 운영에 관한 부분은 제외시킨다. 또 시스템이 운영되는 동안은 통합 시스템과는 개별적으로 행동하며 작업을 완전히 마친 후에만 데이터베이스와의 인터페이스를 통해 통합 시스템의 개념을 갖게 된다. 따라서 각 작업단위에서의 작업 프로세스와 내부 데이터 구조를 모두 알 필요는 없고, 단순히 입력되고 출력되는 데이터만을 분석하여도 통합시스템을 위한 데이터 모델링을 할 수 있다. 이러한 면에서 프로세스 모델링 작업은 생략될 수 있으며 기존의 수준 높은 작업기술을 그대로 사용할 수 있는 통합시스템의 장점을 이용할 수 있게 되는 것이다.

데이터 교환의 한 가지 방법으로 시스템간에 직접 데이터를 교환하는 것이 있다. 이 경우 하나의 시스템이 또 다른 시스템으로 데이터를 전송하기 위해서는 데이터를 수신할 시스템에 적절하도록 데이터 형식을 재구성하는 번역기가 필요하게 된다. 역으로 데이터를 수신한 시스템이 다시 이전 시스템으로 데이터를 전송하려면 역 번역기가 있어야만 한다. 이렇게 할 경우 한 쌍의 시스템이 데이터를 교환하기 위해서는 두 개의 번역기가 있어야 하며 시스템의 개수가  $n$ 개로 증가될수록 번역기는  $n(n-1)$ 개가 필요하게 된다. 예를 들자면 [그림 1-a]와 같이 통합하려는 시스템이 5개일 경우 요구되는 번역기의 총 개수는 20개가 된다. 이러한 방식은 번역기 수의 기하급수적인 증가로 그 효율성이 떨어지는 단점이



[그림 1-a] 직접번역기를 이용한 데이터 교환



[그림 1-b] 중앙데이터베이스를 이용한 데이터 교환

있다.

이 문제를 해결하기 위해서 본 연구에서는 중앙 데이터베이스를 이용하여 데이터 교환이 이루어지도록 는 또 다른 데이터 교환방식을 채택하였다. 각 시스템에서 생성된 데이터들은 우선 데이터베이스에 저장된다. 따라서 각 시스템에서 데이터베이스로 데이터를 전송하기 위한 번역기가 필요하다. 데이터베이스에 저장된 데이터들을 다른 시스템에서 요구할 경우에는 시스템의 입력과일 형식에 맞도록 변환하는 번역기를 통해 데이터를 전송하게 된다. 즉, 시스템간에 직접 데이터를 교환하는 것이 아니라 중앙 데이터베이스라는 매개를 통해 서로간의 데이터를 교환할 수 있도록 하는 것이다. 이렇게 할 경우 번역기는 데이터베이스와 시스템간의 교환만 가능하도록 하면 되므로 시스템의 개수가 n개가 될 경우 번역기는 2n개만 있으면 되는 것이다. 또 중앙 데이터베이스를 통하여 동일한 정보를 공유할 수 있게 될 수 있다. 이것을 그림으로 표현하면 [그림 1-b)와 같다. 데이터베이스는 이와 같이 개별 시스템의 모든 데이터를 다룰 수 있어야 하므로 통합적인 데이터 모델이 제시되어야 한다. 이어서 데이터 모델에 관해 논의하도록 하겠다.

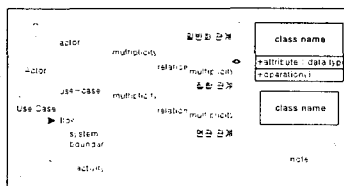
### 3. 데이터 모델

본 연구에서 사용되는 데이터 모델 언어인 UML<sup>(6)</sup>에는 총 9개의 다이어그램으로 시스템을 표현하는데 앞에서 논의된 통합 방법은 데이터가 다뤄지는 프로세스 중심적이기보다는 단순히 데이터가 데이터베이스에 저장되고 추출되어지는 기능을 하는 특성에 따라 유스케이스 다이어그램(use-case diagram)과 클래스 다이어그램(class diagram)만을 이용하도록 하겠다. 본 연구에서 그 대상은 건축 구조물이며 시스템은 건축 구조물 설계를 수행하기 위한 통합 시스템이 되겠다.

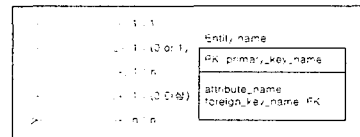
#### 3.1 유스케이스 다이어그램 (use-case diagram)<sup>(6)</sup>

시스템 개발에서 선행되어야 하는 것은 시스템의 요구사항을 분석하는 것이다. UML에서는 시스템 요구 사항 분석을 위한 톨로써 유스케이스 다이어그램을 제공한다. 이 단계의 모델은 사용자 관점에 관한 것으로 시스템과 시스템 사용자(actor) 그리고 요구사항 (use-case)으로 구성된다. activity 시스템은 관점에 따라 전체 통합 시스템이 될 수도 있고 중앙 데이터베이스가 될 수도 있다. 각각의 시스템 관점에 따른 유스케이스 다이어그램을 나타낸 것이 [그림 3]이다. 통합 시스템 관점([그림 3-a)에서 보면 사용자의 요구사항은 칠괄 구조설계를 일관적으로 수행할 수 있도록 하는 것이고 중앙 데이터베이스 관점([그림 3-b)에서는 단순히 데이터의 저장과 읽어들이기이다. 따라서 중앙 데이터베이스는 사용자(개별 응용 프로그램)가 원하는 데이터를 추출하고 제공하며 생성된 데이터를 저장하는 데이터 관리 기능을 갖추어야 한다.

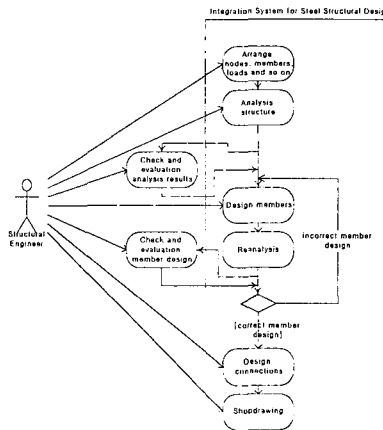
#### 3.2 데이터 모델



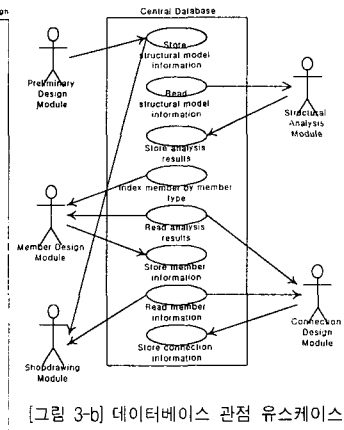
[그림 2-a) 객체 모델 표기 양식



[그림 2-b) 관계형 모델 표기 양식



[그림 3-a) 통합시스템관점 유스케이스 다이어그램



[그림 3-b) 데이터베이스 관점 유스케이스 다이어그램

데이터 모델은 클래스 다이어그램으로 표현되며 이것은 시스템에서 클래스의 정적인 구조를 보여주는 것으로 클래스와 클래스간의 관계로 구성되며 관계는 다음과 같이 분류한다.<sup>(6)</sup>

(1) 연관 관계(Association)

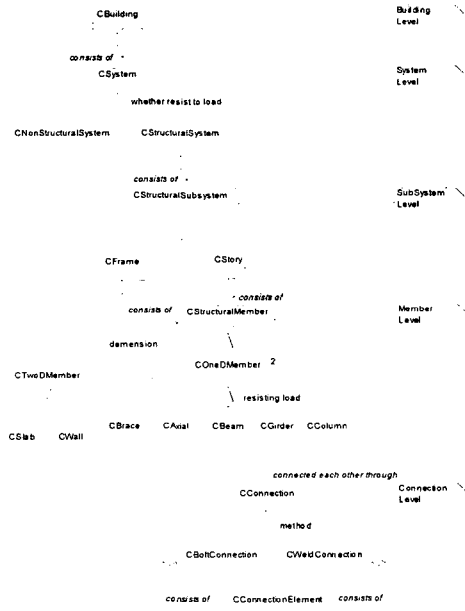
연관 관계는 객체 사이의 연결을 정의하는 것을 의미한다. 하나의 객체와 또 다른 객체가 연결되어있을 때 연결된 객체들은 서로의 본질적 속성에 영향을 주지는 못한다. 단지 단순 연결(link)만 되어있을 뿐이다.

(2) 집합 관계(Aggregation)

집합 관계는 연관 관계의 특수한 경우로서 일종의 전체와 부분(whole-part) 관계이다. 하나의 객체가 또 다른 객체의 성분이나 구성 요소가 될 때 집합 관계를 형성한다고 말할 수 있다.

(3) 일반화 관계(Generalization)

일반화 관계는 더 일반적인 요소와 더 특정화 된 요소사이의 분류학적인 관계, 혹은 분류 위계상 상위 클래스와 하위 클래스 사이의 관계이다. 일반화 관계는 is-a관계라고도 한다. 하위 클래스는 상위 클래스의 모든 속성과 메소드를 상속받으며 추가적으로 자신만의 고유한 속성이나 메소드를 갖는다.



[그림 4] 구조물 모델

집합 등으로 나뉘고 각 집합 방식에 따른 집합 요소들을 갖게 된다. 구조물의 물리적 위계 기준을 토대로 하여 구조물의 개략적인 데이터 모델을 하면 [그림 4]와 같다.

3.2.1 구조물 모델

구조물을 구조적 구성 요소에 따라 다음과 같이 5단계의 위계로 나누었다.

- 건물 단계: 시각적으로 가장 먼저 인지되는 대상이며 모델에 있어서 최상위 단계가 된다. 사용자는 실세계에서 건물이라는 하나의 커다란 대상을 인식하는 것으로 시작한다.
- 구조시스템 단계: 건물은 시스템으로 구성된 것으로 볼 수 있다. 시스템에는 공간시스템, 설비시스템, 구조시스템 등으로 분류되며 이 중 구조 엔지니어가 관심을 갖게 되는 부분은 구조시스템이 되며, 건물을 구조물로서 모델링하게 된다
- 하부구조시스템 단계: 구조 시스템은 하부구조시스템들로 구성된다. 하부구조시스템은 기능과 위치에 따라 구분되며 이들 중 가장 관심이 있는 것은 도면 정보의 단위가 되는 층과 해석의 단위가 되는 골조이다.
- 요소 단계: 각 하부 시스템들은 더 세분화 된 요소들로 구성된다. 구조 요소는 하중을 저항하는 모든 개체로서 보, 기둥, 슬래브 등을 말한다.
- 집합부 단계: 각각의 요소들은 집합부를 통해 연결되고 일체화된다. 집합부는 집합 방식에 따라 용접집합, 볼트

3.2.2 구조설계 데이터 모델

구조설계 데이터 모델을 하기 위해 각 단위 작업으로 분해하여 입력에 필요한 데이터와 출력이 되는 데이터를 분석하였다. 입력된 데이터가 단위 작업을 거쳐 출력 데이터를 생성하기까지의 프로세스는 각 응용 프로그램 내부에서 수행되는 일이므로 내부 프로세스는 고려하지 않았다. 각 작업 단위에서 사용되는 응용 프로그램의 데이터들을 분석하여 동일 객체가 다른 작업 단위에서 사용될 경우 이를 모두 표현하도록 하였다. 개별적인 응용 프로그램을 사용하지는 않으나 개념적인 과정으로서 프로젝트라는 단계를 추가하였다.

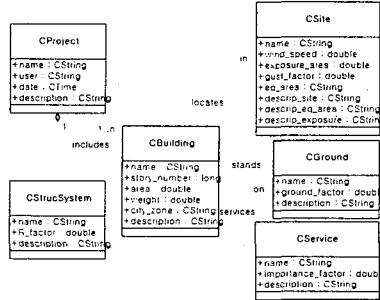
데이터 모델은 논리적 객체 데이터 모델과 물리적 관계형 데이터 모델 두 가지로 제시되었다. 논리적 데이터 모델은 구조설계와 관련된 데이터들의 분석적 모델인 반면 물리적 관계형 모델은 데이터베이스의 처리

를 고려한 모델이다. 데이터베이스 구현을 위한 물리적 모델에는 논리적 모델에서 보여진 모든 객체들을 다루지는 않는다. 데이터베이스의 효율을 고려하여 삭제된 객체도 있고 관계를 나타내기 위해 추가된 부분도 있다. 객체 모델을 관계형 모델로 전환하기 위해서 다음과 같은 규칙을 적용하였다.

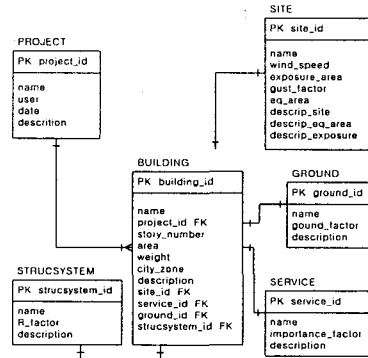
1. 객체지향 모델에서 추출된 각 객체들은 하나의 테이블로서 각각 표현된다.
2. 모든 테이블은 고유 식별자(id)를 갖으며 이것은 테이블의 기본키(primary key)가 된다.
3. 1:1 또는 1:n 연관관계: "1"쪽 객체에 해당하는 테이블의 기본키를 "n"쪽 테이블에서 외래키로 가진다.

4. n:n 연관관계 : 연관 관계를 표현하는 테이블을 추가하여 연관된 객체에 해당하는 테이블의 기본키를 외래키로 한다. 추가적인 속성은 없고 외래키들로만 구성된다.

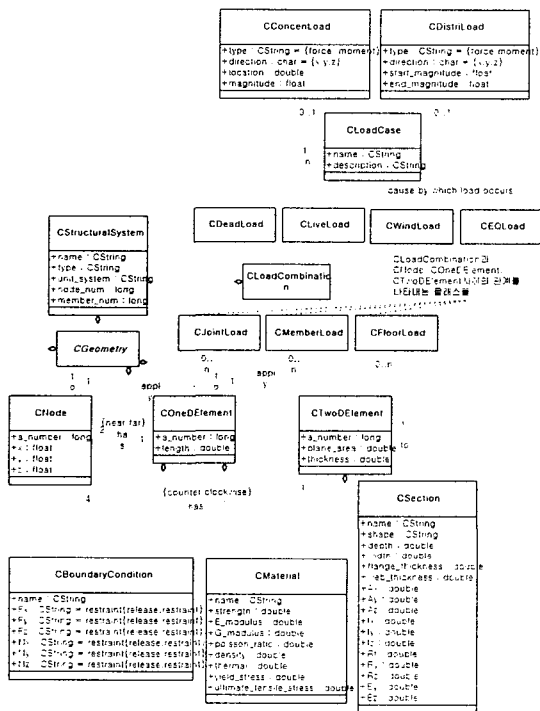
5. 집합관계 : 집합관계는 전체에 해당하는 테이블이 부분에 해당하는 테이블의 기본키를 외래키로 갖고 있도록 한다.



[그림 5-a] 프로젝트 단계 객체 모델



[그림 5-b] 프로젝트 단계 관계형 모델



[그림 6-a] 초기구조설계 단계 객체 모델

된다.

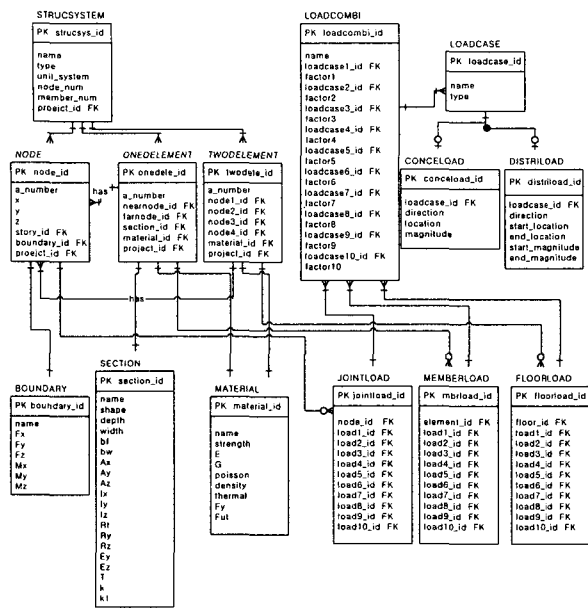
### 3.2.2.3 구조해석 단계

### 3.2.2.1 프로젝트 단계

프로젝트 단계는 독립적인 시스템을 사용하지는 않으나 구조설계 업무 수행에서 개념적으로 이루어지는 작업 단계이다. 이 단계에서는 하나의 구조설계 업무를 프로젝트라는 개념 아래 두고 건물에 관한 일반적인 정보를 다루게 된다. 최상위 객체는 프로젝트가 되고 이 객체는 프로젝트에 관한 일반 정보를 담는다. 객체 모델과 이를 관계형 모델로 전환한 그림은 [그림 5]에서 표현된다.

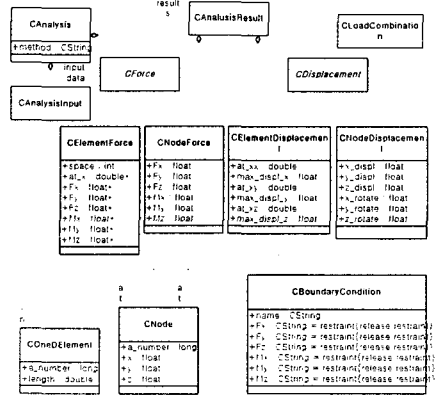
### 3.2.2.2 초기 구조설계 단계

초기 구조설계 단계에서는 계획된 건물을 구조물로서 모델링하고 하중을 배치하는 작업이 이루어진다. 초기 구조설계 단계에서 다루어지는 주요 정보는 절점(node)과 요소(element)이다. 요소는 부재와는 다른 개념으로 실제로 우리에게 인식되는 부재와는 달리, 구조해석을 위해서 분류된 것으로 절점에 의해서 구분될 수 있다. 모델링을 수행하는 주체의 판단에 따라 단일 요소가 부재가 되거나 몇 개의 요소가 모여 부재를 구성하게 될 것이다. 초기 구조설계 작업이 완료되면 구조물의 위상 정보, 하중 정보, 부재 속성 정보 등이 생성된다. 객체 모델과 이를 관계형 모델로 전환한 그림은 [그림 6]에서 표현



[그림 6-1] 초기구조설계 단계 관계형 모델

부재 그룹화와 구체적인 기준 검토 작업은 시스템 내에서 이루어지는 작업이므로 데이터 모델에서 제외한다. 객체 모델과 이를 관계형 모델로 전환한 그림은 [그림 8]에서 표현된다.



[그림 7-a] 구조해석 단계 객체 모델

### 3.2.2.5 접합부설계 단계

접합부 설계는 부재 정보, 연결 부재 정보, 부재력 정보를 갖고 접합부를 제안하고 설계 기준을 만족하는 지 여부를 검토하게 된다. 이 단계에서 다루는 부재는 이전 단계에서 다루었던 요소, 부재와는 다른 개념이다. 실제로 우리에게 인지되고 제작되는 것이다. 접합부 설계가 완료되면 접합부 정보가 데이터베이스에 저장된다. 객체 모델과 이를 관계형 모델로 전환한 그림은 [그림 9]에서 표현된다.

### 3.2.3.6 도면화 단계

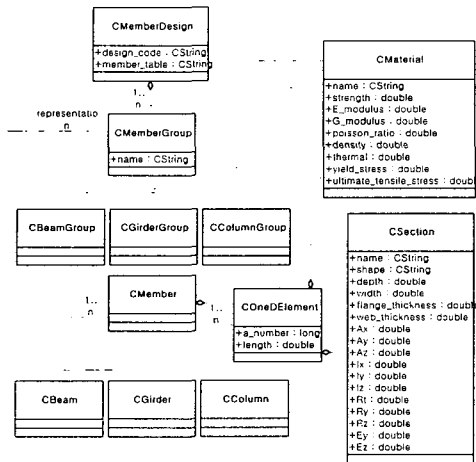
모든 구조설계 작업이 완료되면 철골 부재의 제작을 위한 상세 도면이 제작된다. 도면 작업을 위해서는 구조물의 위상정보와 부재정보가 입력과일로 필요하다. 위상정보와 부재정보는 이전 단계에서 생성된 정보를

구조해석 단계에서는 초기 구조설계 단계에서 생성된 정보들을 받아들여 구조물 모델 정보와 하중정보를 기반으로 하여 구조해석을 수행한다. 구조해석 방법을 결정하고 지정된 구조해석 법에 따라 해석을 수행한 후 그 결과물로서 각 요소의 내력, 반력, 절점 변위, 처짐 등을 산출한다. 내력의 산출은 선형 요소에 대한 것으로 제한하였다. 객체 모델과 이를 관계형 모델로 전환한 그림은 [그림 7]에서 표현된다.

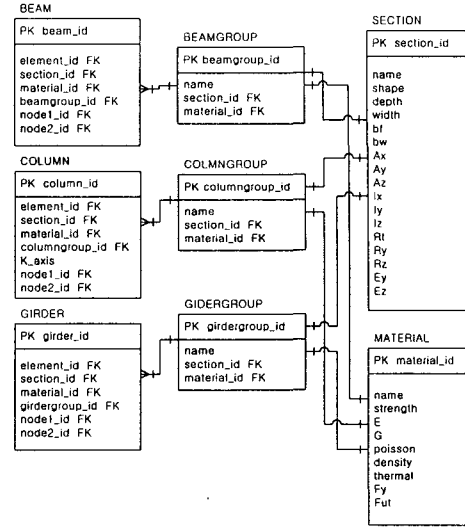
### 3.2.2.4 부재설계 단계

이 단계에서 다루는 부재라는 것은 이전 단계에서 다루었던 요소와 같은 개념으로 볼 수 있다. 일반적으로 내력이나 위상정보 등으로 부재들을 그룹화하여 대표 부재들만 설계하게 된다. 부재설계 단계에서는 초기에 가정된 단면을 이용하여 초기 구조설계와 해석 단계를 거친 결과 값을 이용하여 설계 기준에서 제시하는 사항들을 만족하는지 여부를 검토한다. 부재설계가 완료되면 부재 설계 정보가 데이터베이스에 저장된다.

[그림 7-b] 구조해석 단계 관계형 모델

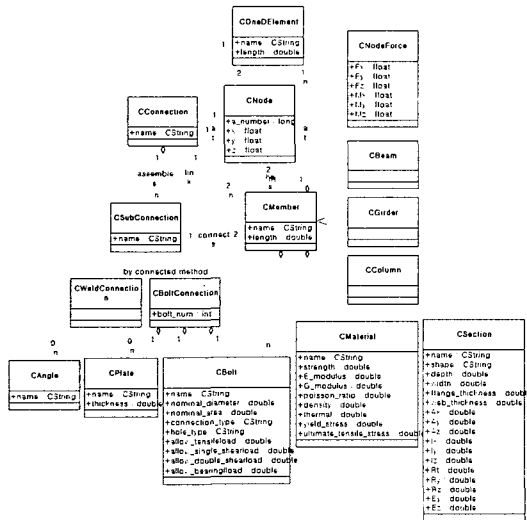


[그림 8-a] 부재설계 단계 객체 모델

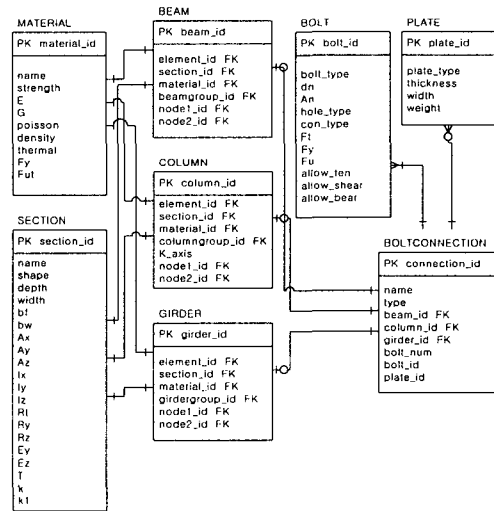


[그림 8-b] 부재설계 단계 관계형 모델

받아들이고 접합부와 관련된 상세설계는 도면 작업을 위한 응용 프로그램 내에서 수행하게 된다. 여기서 생성된 정보는 데이터베이스에 저장되지 않는다. 객체 모델과 이를 관계형 모델로 전환한 그림은 [그림 10]에서 표현된다.



[그림 9-a] 접합부설계 단계 객체 모델

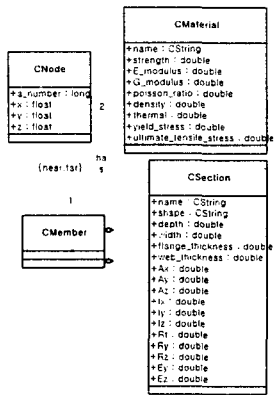


[그림 9-b] 접합부설계 단계 관계형 모델

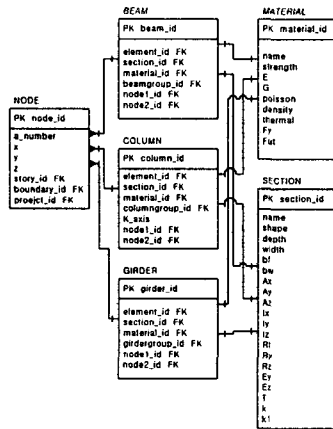
#### 4. 데이터베이스 내부 스키마

데이터베이스의 내부 스키마는 데이터들이 데이터베이스에 저장되는 구조를 표현한 것으로 SQL로 작성되었다. 스키마는 데이터의 추가, 삭제, 수정을 포함하여 필요한 데이터들만을 필터링(filtering)하는 작업을 한다. 많은 양의 코드로 되어있으므로 지면상 생략하도록 하겠다.

#### 5. 결 론



[그림 10-a] 도면화 단계 객체 모델



[그림 10-b] 도면화 단계 관계형 모델

본 연구는 철골 구조설계 통합 시스템을 위해 개별 응용 프로그램과 중앙 데이터베이스를 이용한 통합 방법을 제시하였다. 개별 응용 프로그램에서 사용되는 데이터들을 모두 포괄할 수 있는 데이터 모델을 제안하였으며 이를 토대로 하여 중앙 데이터베이스를 구축하였다. 이러한 연구를 통하여 다음과 같은 결론을 얻을 수 있었다.

1) 개별 응용 프로그램의 데이터들을 분석하여 이를 모두 표현할 수 있는 데이터 모델을 제안함으로써 개별 응용 프로그램간 통합을 위한 토대를 마련하였다.

2) 몇 가지 규칙을 적용함으로써 실제 구조물을 가장 효과적으로 표현할 수 있는 객체 모델을 관계형 데이터베이스 구현에 적합한 관

계형 모델로 전환하는 것이 가능함을 볼 수 있었다. 개발자의 입장에서 두 번의 모델 전환을 위한 작업이 추가되므로 개발 효율은 다소 떨어진다고 볼 수도 있겠지만, 객체 모델의 실세계 표현 능력과 관계형 데이터베이스의 데이터 관리 능력을 고려할 때 두 가지 모델을 접목시키는 것이 운영상에 더 이점이 있을 것으로 생각된다.

3) 현재 중앙 데이터베이스와 응용 프로그램간의 데이터 교환을 위한 인터페이스를 개발 중에 있으며 이것이 완료되면 데이터 모델의 적합성을 검증할 수 있을 것이다.

4) 본 연구에서는 중앙 데이터베이스의 데이터 관리에만 초점이 맞추어져 모델링에 국한된 주제를 다루었으나 진정한 통합 시스템의 의미를 찾기 위해서는 사용자가 분산된 현실에 적당한 분산형 시스템 개발이 진행되어야 하겠다. 본 연구에서 구축된 데이터베이스를 중앙 데이터베이스로 활용하고 지역 데이터베이스를 이용한 분산 환경을 구축하는 것이 가능할 것이다.

## 6. 참고문헌

1. 정운철 "철근콘크리트 구조설계 통합 시스템을 위한 데이터베이스 모델 제시 및 응용", 한양대학교, 석사학위 논문, 1998.
2. 박용필, 정재훈, 김진무, 조철호, "통합설계 데이터베이스 개발을 위한 구조분야 데이터모델에 관한 연구", 대한건축학회학술발표논문집, 제17권 제2호, p.1013-p.1020, 1997.10.
3. 황영삼, "설계통합화를 위한 통합건물모델의 구조에 관한 연구", 대한건축학회논문집, 제12권 제2호, p.71-p.80, 1996.2.
4. Ajay Lavakare, H.Craig Howard, "Structural Steel Framing Data Model", CIFE Technical Report No.012, Stanford University, 1989.6.
5. A.J.Crowley, CIMSteel Integration Standards Release 1.1 Overview, University of Leeds, 1997.
6. Hans-Erik Eriksson & Magnus Penker, UML Toolkit, John Wiley&Sons, inc, 1998.
7. 김상하, 이영신, 관계형 데이터베이스 모델링 및 설계구축 실무, 광문각, 1997.
8. David Lockman, 노정운 역, Oracle 8 데이터베이스 개발 21일 완성, 인포북, 1998.
9. CASEC, GT STRUDL User Guide(Menu, Analysis, Design), Georgia Institute of Technology, 1997.
10. CSC, X-Steel user manual vol. 1, 1997.
11. <http://erg.ucd.ie/combine.html>
12. <http://www.leeds.ac.uk/civil/research/cae/cis/cis.htm>