

통합 구조설계 시스템을 위한 설계 객체 모델의 개발과 구현

Development and Implementation of Design Object Model for Integrated Structural Design System

천진호* 이창호** 이병해***
Cheon, Jin-Ho Lee, Chang-Ho Lee, Byung-Hai

ABSTRACT

This paper describes an example of developing an integrated design system, Integrated Structural Design System for Reinforced Concrete Buildings(INDECON). INDECON incorporates a central database and three design modules: a preliminary design module(PDM), a structural analysis module(SAM), and a detailed design module(DDM). The development of INDECON begins with the development of design models including Design Object Model(DOM) which describes design data during the structural design process. The Design Object Model is transformed to Design Table Model(DTM) for the central database, and is specified to be in detail for the three design modules. Then the central database is implemented and managed by relational database management system(RDBMS), and the three design modules are implemented using C++ programming language. The central database in the server computer communicates with the design modules in the client computers using TCP/IP internet protocol. The developing procedure for INDECON in this paper can be applied for developing more comprehensive integrated structural design systems.

1. 서론

통합 구조설계 시스템은 초기 설계, 구조 해석, 상세 설계 등의 단위 설계 작업들을 하나의 시스템으로 구성하며, 구조설계 작업을 진행하면서 단위 설계 작업별로 발생하는 다양한 설계 정보들을 종합적으로 표현하고 체계적으로 저장 관리할 수 있는 능력을 갖추어야 한다. 그러나 구조설계 과정은 상당부분에 있어서 수치적 알고리즘 방법론으로는 시스템화하기에 적절치 못한 비정형의 설계 데이터와 작업을 다수 포함하고 있다. 따라서 구조설계 전 과정을 지원할 수 있는 통합 구조설계 시스템 개발을 위해서는 이러한 비정형적인 설계 정보를 일관되게 표현할 수 있는 방법론이 모색되어야 한다.⁽¹⁾⁻⁽⁴⁾ 즉, 효율적인 통합 구조설계 시스템의 개발을 위해서는 비정형의 특성을 가진 구조설계 정보를 정형적으로 표현한 설계 모델의 개발과 설계 모델을 통한 효율적인 시스템 개발 방법론이 요구된다. 본 연구는 철근 콘크리트 구조물을 대상으로 통합 구조설계 시스템 구현을 고려한 설계 모델의 개발과 개발된 설계 모델을 적용한 통합 구조설계 시스템의 구현 과정을 제시하고자 하는데 그 목적이 있다. 일련의 연구를 통해 개발된 설계 모델에는 설계 데이터를 표현하는 설계

* 정회원 · 국토연구원 책임연구원
** 정회원 · 한경대학교 건축공학과 조교수
*** 정회원 · 한양대학교 건축공학부 교수

객체 모델(Design Object Model, DOM)과 설계 작업을 표현하는 설계 행위 모델(Design Activity Model, DAM)이 있으며, 본 논문에서는 설계 객체 모델을 중심으로 그 내용을 기술한다.

본 연구에서 제안된 설계 객체 모델이 통합 구조설계 시스템의 구현에 이용되는 과정은 그림 1과 같이 3 단계로 표시할 수 있다.

① 1 단계: 설계 객체 모델의 개발

1 단계에서는 통합 구조설계 시스템에 대한 기능적 요구사항을 정의한 후, 정의된 기능적 요구사항을 고려하여, 구조설계 과정 분석을 통해 얻은 설계 데이터를 이용하여 설계 객체 모델을, 설계 작업을 이용하여 설계 행위 모델을 개발한다. 이러한 두 가지 모델 중에서 설계 객체 모델을 본 논문에서 다룬다.

② 2 단계: 설계 객체 모델의 변환 및 상세화

1 단계에서 개발된 설계 객체 모델은 물리적인 통합 구조설계 시스템의 구현에 적용하기 위해 시스템 개발 환경을 고려한 설계 객체 모델의 변환과 상세화 과정이 요구

된다. 즉, 중앙 데이터베이스 구현을 위하여 객체지향 형식의 설계 객체 모델을 관계형(Relational) 형식의 설계 테이블 모델로 변환하며, 설계 응용 모듈의 구현을 위해서 설계 객체 모델을 상세화한다.

③ 3 단계: 중앙 데이터베이스 및 설계 응용 모듈의 구현

3 단계에서는 2 단계에서 변환 및 상세화된 설계 객체 모델을 이용하여 물리적인 통합 구조설계 시스템을 구현하는 과정을 거친다. 즉, 통합 구조설계 시스템의 구성 모듈인 중앙 데이터베이스를 설계 테이블 모델을 이용하여 구현하고, 상세화된 설계 객체 모델을 이용하여 설계 응용 모듈을 구현한다.

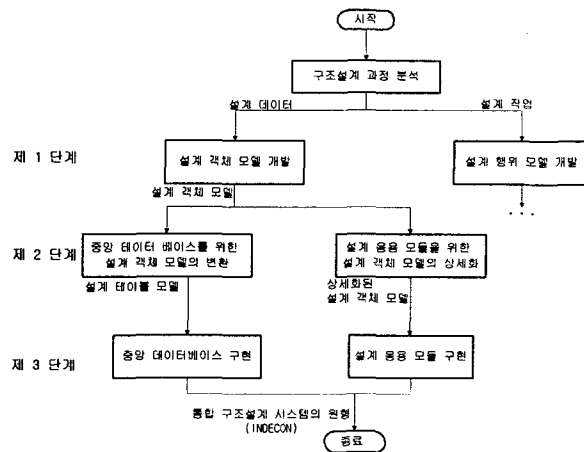


그림 1 통합 구조설계 시스템 구현 과정

2. 설계 객체 모델의 개발

먼저 구조설계 과정을 분석하여 설계 정보를 분류하고 분류된 설계 데이터와 설계 작업을 통하여 설계 객체 모델의 기본 구조를 결정한다. 이와 같이 작성된 설계 객체 모델의 기본 구조는 통합 구조설계 시스템의 기능적 특성을 정의한 후 이에 따른 설계 객체 모델의 요구사항을 적용하여 재구성된다. 본 연구에서 정의한 통합 구조설계 시스템의 기능적인 특성은 구조설계에 대한 전문지식을 요구하는 설계 작업(예: 단면 가정, 부재 그룹핑 등)을 수행할 수 있는 것으로 정의하였다. 이러한 기능적인 특성은 다음과 같은 설계 객체 모델 구조에 대한 요구사항으로 기술될 수 있다.

- ① 계획 설계 정보와 구조설계 정보를 유기적으로 표현하여, 계획 설계 정보에서부터 구조설계 정보로 유연하게 변환이 가능한 구조이어야 한다.
- ② 삼차원 공간상에서 구조부재의 위상정보(부재 배치상태)를 인식하고 표현할 수 있는 구조이어야 한다.

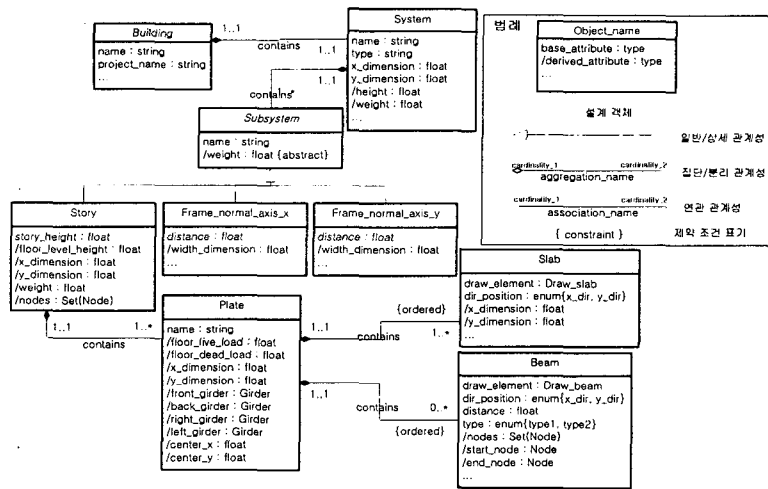


그림 2 설계 객체 모델

이와 같은 설계 모델에 대한 요구사항을 고려하여 UML⁽⁵⁾을 적용하여 설계 객체 모델을 개발하였다. 그림 2는 설계 객체 모델의 일부분을 보여준다.

3. 설계 객체 모델의 변환 및 상세화

본 연구에서 제안된 통합 구조설계 시스템의 구조는 크게 중앙 데이터베이스와 단위 설계 작업을 수행하는 설계 응용 모듈로 구성된다. 통합 구조설계 시스템 구현 과정의 1 단계(그림 1)에서 개발된 설계 객체 모델은 시스템 개발 환경에 독립적인 구조이다. 이러한 설계 객체 모델은 2 단계에서 중앙 데이터베이스를 위한 설계 테이블 모델로 변환되고 동시에 설계 응용 모듈을 위해 상세화 된다. 또한 본 단계에서 추가의 설계 객체 모델로서 설계 정보의 그래픽 처리를 담당하는 그래픽 객체 모델(Graphic Object Model, GOM)과 설계 응용 모듈의 제어관련 정보를 처리하는 사용자 인터페이스 객체 모델 (User Interface Object Model, UIOM)이 개발된다.

3.1 시스템 개발 환경과 고려사항

설계 객체 모델의 변환 및 상세화 작업은 시스템 개발 환경을 고려하여 수행된다. 즉, 시스템 개발 환경을 통하여 설계 객체 모델에 대한 기능적 특성과 구현을 고려한 상세 정보를 부여한다. 시스템 개발 환경에 대한 항목은 다음과 같다.

- ① 어떠한 소프트웨어 개발 도구를 사용하는가
- ② 어떠한 시스템 구조로 통합 구조설계 시스템을 개발하는가

이에 본 연구에서는 통합 구조설계 시스템 구현을 위한 시스템 개발 환경으로 다음과 같은 방법을 선정 적용하였다.

- ① 본 연구에서 개발된 설계 객체 모델은 객체지향 개념을 기반으로 개발되었으며, 본 모델에 대한 자연

스러운 구현 도구인 객체지향 프로그램 언어를 사용한다.

- ② 중앙 데이터베이스 구현을 위해 관계형 데이터베이스 관리 시스템을 선정한다.
- ③ 본 연구에서 구성하고자하는 통합 구조설계 시스템의 구조는 중앙 데이터베이스를 중심으로 3 가지 설계 응용 모듈인 초기 설계 모듈(Preliminary Design Module, PDM), 구조 해석 모듈(Structural Design Module, SAM), 상세 설계 모듈(Detailed Design Module, DDM)로 구성한다.

3.2 중앙 데이터베이스를 위한 설계 객체 모델의 변환

시스템 개발 환경에서 선정된 중앙 데이터베이스 관리 시스템은 관계형 데이터베이스 관리 시스템으로, 데이터베이스에 저장되는 데이터 형식은 테이블(Table) 구조의 관계형 데이터 형식이다. 한편, 중앙 데이터베이스 구현을 위해 적용되는 설계 모델인 설계 객체 모델은 객체지향 데이터 형식으로 이루어져있으며, 중앙 데이터베이스는 테이블 구조로 데이터를 표현함으로써 두 데이터 형식의 차이점에 대한 적절한 변환 작업을 수행되어야 한다. 중앙 데이터베이스 구현을 위한 설계 객체 모델의 변환 작업은 다음과 같이 두 과정으로 이루어지며 이러한 변환 작업의 결과로 생성된 정보 모델을 설계 테이블 모델(Design Table Model, DTM)이라고 지칭한다.

1) 설계 객체의 선정

설계 객체 모델로부터 중앙 데이터베이스 구현에 적용할 설계 객체의 선정 기준은 다음과 같다. 즉, 설계 응용 모듈들 사이에 공유되거나 다음 설계 단계의 설계 응용 모듈에 이용되는 설계 객체들과 하나의 설계 응용 모듈에서만 사용되는 설계 객체 중 구조설계 완료 시기까지 연속성을 요구하는 설계 객체들이다.

2) 설계 객체의 변환

선정된 설계 객체들은 일대일(1:1)의 설계 테이블로 변환됨을 기본 원칙으로 한다. 그림 3은 하나의 설계 객체를 설계 테이블로 변환하는 예를 보여준다. 그림 3(a)는 설계 객체 모델(그림 2)에서의 System 설계 객체이며 그림 3(b)는 중앙 데이터베이스의 구현을 위해 변환된 SYSTEM 설계 테이블이다. 설계 객체 모델의 각 설계 객체는 중앙 데이터베이스를 위한 설계 테이블 모델의 설계 테이블에 해당하며, 설계 객체의 실체인 인스턴스(Instance)는 설계 테이블의 레코드(Record)에 해당한다. SYSTEM 설계 테이블의 속성에는 System 설계 객체에는 없는 속성 ID가 있다. 이는 테이블에 포함되는 레코드들의 유일성을 표현한다. 설계 객체는 대한 설계 정보의 특성에 따라 그림 3의 System 설계 객체의 변환처럼 두 개의 설계 테이블 (SYSTEM 설계 테이블과 TYPE 설계 테이블)로 분할될 수 있다.

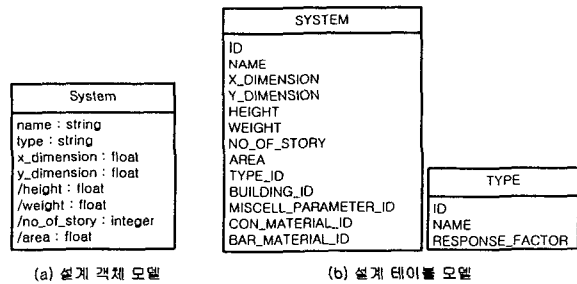


그림 3 설계 객체와 변환된 설계 테이블

3.3 설계 응용 모듈을 위한 설계 모델의 상세화

설계 객체 모델은 응용 모듈을 구현하기 위한 일부로서도 이용이 되는데, 설계 객체 모델이 설계 응용 모듈의 구현 작업에 적용되기 위해서는 시스템 개발 환경을 고려한 상세화가 요구된다. 시스템 개발 환경을 고

려한 설계 모델의 상세화 작업은 다음과 같은 내용으로 이루어진다.

- ① 설계 객체 모델의 각 설계 객체에 대한 유도속성(Derived Attribute)과 제약조건(Constraint)에 대한 객체 기술서를 작성한다.
- ② 설계 응용 모듈별로 설계 객체 모델을 분류한다.
- ③ 구현을 고려한 추가의 객체 모델을 작성한다.

1) 객체 기술서 작성

설계 응용 모듈의 구현을 위한 설계 객체 모델의 상세화 작업에서, 설계 객체 모델의 각 설계 객체에 대한 유도속성과 제약조건에 대한 객체 기술서를 작성한다. 본 연구에서 객체 기술서는 OCL⁽⁶⁾로 표기된다. 그림 4는 System 설계 객체(그림 2)에 대한 객체 기술서로서 세 부분으로 되어 있는데 각각 객체 정의문, 유도속성 정의문, 그리고 제약 조건 정의문을 나타낸다.

```

/* The System design object is top-level design object for structural system information on a structural design. */
System
name : string                -- identity name
type : string                -- type of structural system
x_dimension : float          -- dimension of x axial direction
y_dimension : float          -- dimension of y axial direction
height : float               -- height of building
...
/* The value of the height attribute is derived from accumulation of height of stories, which constitute a system. */
System :: height() : float
result := story->iterate(
    s      : Story;
    res : float :=0 |
    res := res + s.story_height)
...
/* The maximum value of no_of_story is 10. */
System
no_of_story() <= 10
...

```

그림 4 설계 객체의 객체 기술서

2) 설계 응용 모듈별 설계 모델의 분류

이와 같이 상세화된 설계 객체 모델은 세 가지로 구성된 설계 응용 모듈(초기 설계 모듈, 구조 해석 모듈, 상세 설계 모듈)별로 분류된다. 즉, 설계 객체 모델은 구조설계 과정 전체를 대상으로 표현된 것이며, 단위 설계 작업별로 구현된 각 설계 응용 모듈들은 전체 설계 모델 중 한 부분만을 그 대상으로 한다. 설계 객체 모델은 각 단위 설계 작업별로 설계 객체를 중복 사용함으로써, 각 설계 응용 모듈별로 설계 객체 모델을 구분하기보다는 필요한 설계 객체를 선택 구성하여 설계 응용 모듈별 설계 객체 모델의 분류 작업을 수행한다.

4. 중앙 데이터베이스 및 설계 응용 모듈의 구현

통합 구조설계 시스템 구현과정 3 단계(그림 1)에서는 2 단계에서 시스템 개발환경을 고려하여 변환 및 상세화된 설계 객체 모델을 구현도구를 이용하여 물리적인 통합 구조설계 시스템으로 구현한다. 먼저, 중앙 데이터베이스의 구현을 위해 변환된 설계 객체 모델(설계 테이블 모델)을 관계형 데이터베이스 구축언어인 SQL(Structured Query Language)을 이용하여 중앙 데이터베이스로 구현하며, 상세화된 설계 객체 모델을 C++ 객체지향 언어를 이용하여 설계 응용 모듈로 구현한다.

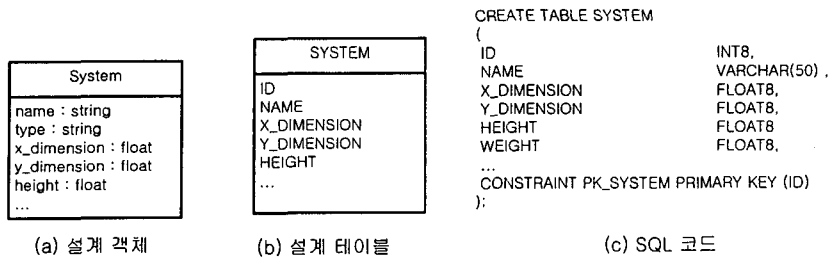


그림 5 설계 객체의 SQL코드 구현 과정

4.1 중앙 데이터베이스의 구현

중앙 데이터베이스 구현을 위해 변환된 설계 테이블 모델(DTM)은 관계형 데이터베이스 구축 언어인 SQL(Structured Query Language)을 이용하여 DBMS 내에 구현된다. 그림 5는 설계 객체에 대한 SQL 코드로의 구현 과정을 보여주고 있다. 설계 객체(그림 5(a))는 설계 테이블(그림 5(b))로 변환되고 다시 SQL 코드(그림 5(c))로 구현된다.

4.2 설계 응용 모듈의 구현

설계 응용 모듈의 구현을 위해 상세화된 설계 객체 모델에는 상세화된 설계 객체 모델(DOM), 그래픽 객체 모델(GOM), 그리고 사용자 인터페이스 객체 모델(UIOM)로 구성되어 있다(그림 6(a)). 한편 이와 같은 설계 모델로부터 구현되는 설계 응용 모듈의 C++ 클래스 군은 그림 6(b)과 같이 4 가지 부분으로 구성되어 있다. 즉, 구조 부재의 그래픽 출력을 표현하기 위한 그래픽 출력 정보 관련 C++ 클래스 코드 부분, 설계 정보 자체를 표현하는 구조설계 정보 관련 C++ 클래스 코드 부분, 설계 응용 모듈을 운영하고 사용자에게 설계 작업 환경을 제공하여 주는 설계 응용 모듈 수행 정보 관련 C++ 클래스 코드 부분, 그리고 중앙 데이터베이스의 설계 정보와 설계 응용 모듈내의 설계 정보에 대한 연결 기능을 수행하는 중앙 데이터베이스 인터페이스 정보 관련 C++ 클래스 코드 부분으로 구성되어 있다.

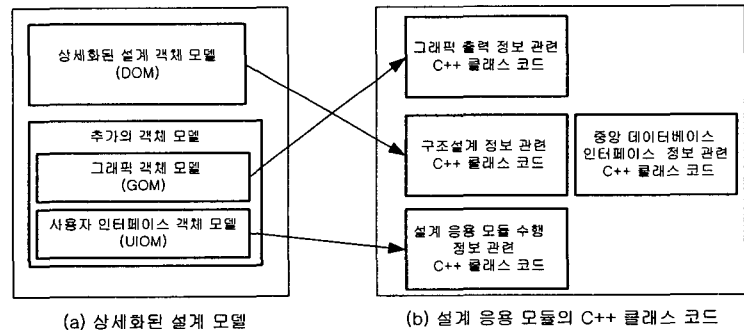


그림 6 상세화된 설계 모델과 C++ 클래스 코드 사이의 관계

4.3 통합 구조설계 시스템 구조

중앙 데이터베이스와 세 가지 설계 응용 모듈(초기 설계 모듈, 구조 해석 모듈, 상세 설계 모듈)을 포함하는 통합 구조설계 시스템(INDECON)의 원형(Prototype)시스템 구조는 그림 7과 같다. 중앙 데이터베이스는 서버(Server) 컴퓨터에 설치되며 설계 응용 모듈은 각각 다른 클라이언트(Client) 컴퓨터에 설치된다. 공간적으로 분리된 서버 컴퓨터와 클라이언트 컴퓨터 사이의 정보 교환은 인터넷(Internet)을 통하여 tcp/ip 통신 규약에 의해서 수행된다. 인터넷을 통한 중앙 데이터베이스와 설계 응용 모듈 사이의 연결은 타 시스템

과의 호환성이나 이식성에서 표준으로 자리잡고 있는 Microsoft사의 ODBC(Open Database Connectivity)를 사용하였다. 중앙 데이터베이스 인터페이스 정보 관련 C++ 클래스 코드들이 ODBC Driver의 API 함수들을 구동하여 PostgreSQL DBMS를 제어한다. 이와 같은 통합 구조설계 시스템의 구조는 데이터 베이스 관리 시스템에 독립적인 구조이며 새로운 설계 응용 모듈의 추가 시에도 확장성에 유연한 구조이다. 이와 같은 통합 구조설계 시스템 개발 과정을 적용하여 개발된 통합 구조설계 시스템의 실행화면은 그림 8과 같다. 그림 8의 설계 단계는 공간용도와 바닥 마감을 설정하기 위한 층 평면 내의 단위 공간인 플레이트를 배치하는 작업 과정이다.

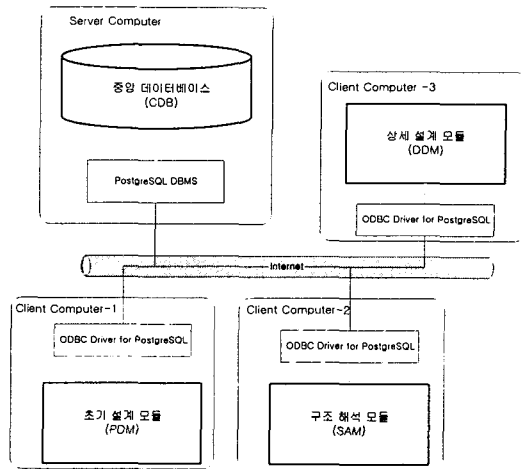


그림 7 통합 구조설계 시스템(INDECON)의 구조

5. 결론

본 연구에서는 통합 구조설계 시스템의 구현을 위한 한가지 개발 방법론을 기술하였다. 즉, 설계 객체 모델을 개발한 후, 중앙 데이터베이스와 설계 응용 모듈들로 이루어진 통합 구조설계 시스템의 구현을 위하여 설계 객체 모델을 변환하고 상세화 시켰으며, 이것을 바탕으로 하여 중앙 데이터베이스와 설계 응용 모듈을 구현하는 일련의 시스템 구현 방법론을 제시하였다. 구현 단계별 결론은 다음과 같다.

1) 설계 객체 모델의 개발에 관한 결론

본 단계에서 제안된 설계 객체 모델은 설계 정보를 정형적으로 기술하며, 시스템 개발 환경에 독립적인 구조이다. 통합 구조설계 시스템에 대한 새로운 기능적 요구 및 수정사항 발생시, 설계 모델을 통하여 현재의 통합 구조설계 시스템의 상태를 충분히 파악할 수 있으며 더욱 효율적으로 통합 구조설계 시스템에 대한 새로운 기능의 분석과 적용이 가능하였다.

2) 설계 객체 모델의 변환 및 상세화에 관한 결론

설계 객체 모델은 구조설계 과정 분석을 통해서 설계 정보를 정형적으로 표현한 구조이며, 본 설계 모델을 통합 구조설계 시스템의 구현에 적용하기 위해서는 시스템 개발 환경을 고려한 설계 모델의 변환 및 상세화 작업이 요구된다. 이에, 설계 객체 모델은 중앙 데이터베이스와 설계 응용 모듈의 구현을 위한 변환 및 상세화 작업에 공동으로 적용된다. 중앙 데이터베이스 구현을 위한 설계 객체 모델의 변환시, 변환의 대상이

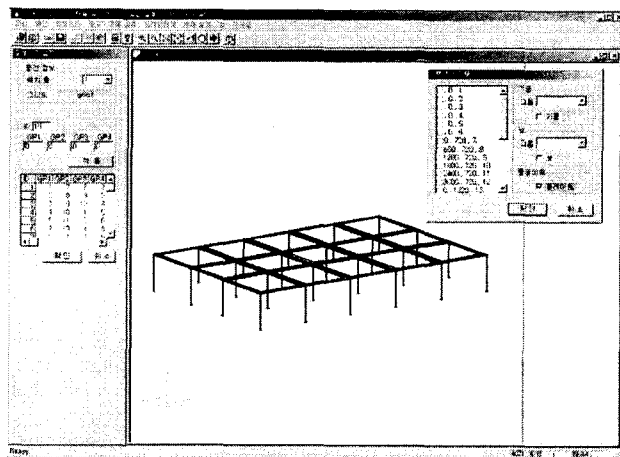


그림 8 플레이트 설정 실행 화면

되는 설계 객체는 각 설계 응용 모듈간에 공통으로 이용되는 설계 객체와 시스템 수행 종료까지 그 정보의 보존이 필요한 설계 객체를 그 대상으로 중앙 데이터베이스를 위한 설계 테이블로 변환함이 합리적이라 판단되며, 중앙 데이터베이스 구현을 위해 선택된 설계 객체 중 하나의 설계 객체는 하나의 설계 테이블로 변환됨을 원칙으로 하였다. 상세화된 설계 객체 모델은 설계 응용 모듈에 구분 없이 전체 구조설계 과정을 대상으로 작성되며, 각 설계 응용 모듈별 구현을 위하여 분류된다. 이와 같이 시스템 구현 전까지 설계 응용 모듈 전체에 대한 설계 모델을 작성함으로써 각 설계 응용 모듈 사이의 설계 정보 표현과 시스템 구조에 대한 일관성을 유지할 수 있었다. 설계 응용 모듈의 구현을 위한 설계 모델의 상세화 작업은 시스템 개발 환경인 소프트웨어 개발 도구와 통합 구조설계 시스템의 구조를 고려하여 수행되어야 한다.

3) 중앙 데이터베이스와 설계 응용 모듈 구현에 관한 결론

제안된 설계 객체 모델은 변환과 상세화가 이루어진 후, 통합 구조설계 시스템을 구성하는 중앙 데이터베이스와 각 설계 응용 모듈로 구현된다. 설계 테이블 모델은 SQL 코드로 구현된다. 중앙 데이터베이스의 이용은 각 설계 응용 모듈들을 결합함으로써 설계 정보의 독립성을 보장할 수 있고 설계 정보의 중복성을 최대한 줄여나갈 수 있었다. 상세화된 설계 객체 모델과 추가의 객체 모델들은 각각 설계 응용 모듈을 구성하는 C++ 클래스 군으로 구현된다.

본 연구에서는 설계 객체 모델 개발로부터 통합 구조설계 시스템 구현까지의 개발 과정을 구체적으로 제시함으로써, 제안된 시스템 개발 방법론은 유사 통합 구조설계 시스템 개발 과정에 대한 효율적인 방법론으로 이용될 수 있다고 판단된다.

감사의 글

본 연구는 한양대학교 초대형구조시스템 연구센터의 지원으로 수행된 과제로 이에 감사의 뜻을 표합니다.

참고 문헌

1. Eastman, C. M., "Building Product Models: Computer Environments supporting Design and Construction", CRC Press, 1999
2. Rivard, H., and Fenves, S. J., "A Representation for Conceptual Design of Buildings", Journal of Computing in Civil Engineering, ASCE, Vol. 14, No. 3, 2000, pp. 151-159
3. 이창호, "구조설계 CAD 시스템 개발을 위한 설계정보간의 상호관계의 표현", 대한건축학회 논문집, 제15권 제2호, 1999, pp. 97~108
4. 천진호, "통합 구조설계 시스템을 위한 설계 모델 개발과 구현", 한양대학교 박사학위논문, 2000
5. Eriksson, H. E., and Penker, M., "UML Toolkit", Wiley, 1998
6. Jos, W., and Anneke, K., "The Object Constraint Language Precise Modeling with UML", Addison-Wesley, 1999