

철근 콘크리트 구조설계 통합시스템을 위한 데이터베이스 모델 제시와 응용

A Research on the Proposal and Application of Data Model for an Integrated System of Reinforced Concrete Structures

정 윤 철* 천 진 호** 서 용 표*** 이 병 해****
Jeong, Yoon-Cheol Cheon, Jin-ho Seo, Yong-Pyo Lee, Byung-Hai

ABSTRACT

The purpose of this study is to develop a central database and a database management system to store and manage information systematically from each module of an integrated structural design system. In order to efficiently express structural design process related to the data which is very complex, we used an object-oriented modeling methodology to propose the possibility to apply a database schema for application programs in an integrated system for reinforced concrete structural design .

Based on this model, we developed an interface between each module and central database. After modeled by using object modeling technique, the database was mapped by the relational database table. Then the central database and the interface were programed by using Visual C++, a windows environmental development tool.

1. 서 론

최근 컴퓨터 하드웨어와 소프트웨어의 비약적인 발전으로 인하여 건축 분야에서도 기존의 인력에 의존한 작업들을 컴퓨터를 이용하여 자동화하려는 시도가 다방면으로 이루어지고 있다. 건축구조 분야도 예외는 아니어서 통합시스템 구축을 위한 연구 사례가 발표¹⁾²⁾되고 있으며 이러한 연구 결과로 INDECON(INTElligent structural DEsign system for reinforced CONcrete buildings)³⁾을 개발하였다.

건축 구조설계 통합시스템은 초기 구조설계, 구조해석, 부재 설계 등의 과정을 하나의 시스템으로 구성하는 것과 실제의 프로젝트들을 진행하면서 각각의 모듈별로 발생하는 다양한 정보들을 종합적으로 표현하고 체계적으로 저장하고 관리하는 것을 목적으로 한다. 그러나 기존의 파일 기반 시스템은 수용 능력이 한계에 다다르고 있어 중앙 데이터베이스(Central Database)와 데이터베이스 관리 시스템(Database Management System)은 필수적인 요소이다.

현재까지 구조분야에서 철골구조는 다양한 방법의 모델링 방안이 제시⁴⁾되어 있으나 철근 콘크리트 구조관련 데이터 모델링과 구현은 그 연구가 미비한 실정이다. 따라서, 본 연구의 목적은 실제의 구조설계 과정을 면밀히 검토하여 철근 콘크리트 구조설계 통합시스템(INDECON)에 적합한 데이터 모델과 데이터베이스 스키마 모델을 제시하고 제시된 모델을 바탕으로 관계형 데이터베이스 테이블을 구현한 후, 최종적으로 데이터베이스 시스템과 연동된 응용 프로그램 모듈과의 자동화된 Interface를 개발하여 그 유용성을 검증하는 데 있다.

* 학생회원, 한양대학교 건축공학과 석사과정
** 학생회원, 한양대학교 건축공학과 박사과정
*** 정회원, 한국전력연구원 기계공학연구소 선임연구원
**** 정회원, 한양대학교 건축공학과 교수

구조설계 관련 데이터들은 비정형적 요소들이 많고 요소들간의 상관관계가 매우 복잡하게 얽혀 있기 때문에 이를 효과적으로 표현하기 위해서는 객체지향 모델링 방법이 필요하며, 본 연구에서는 James Rumbaugh가 제안한 OMT(Object Modeling Technique)⁵⁾를 사용하여 통합 구조설계 시스템의 각 모듈에 필요한 데이터를 개념적인 세부분류보다는 어느 정도 단순화된 분류와 응용 프로그램의 프로세스와의 호환을 고려하여 모델링하고 각각의 객체간의 연결관계를 정의하였다.

데이터베이스 관리 시스템으로는 실제 실무에서 가장 널리 쓰이고 있고 응용 프로그램과 데이터베이스와의 연결이 비교적 용이한 관계형 데이터베이스인 Oracle 7.3.3을 선택하였고 Oracle 내에서 제공하는 SQL Plus 3.3^{6,7)}을 사용하여 데이터베이스 스키마를 구현하였다. 구현된 데이터베이스 스키마를 적용해서 윈도우 개발환경에 최적화된 객체지향 기반의 개발툴인 Visual C++ 5.0을 사용하여 철근 콘크리트 구조설계 통합시스템의 구조설계 전처리 과정(Pre-processing) 모듈, 구조해석 모듈, 부재 설계 모듈, 물량산출 모듈등 각 단위 설계 모듈과 데이터베이스와의 능동적인 데이터교환이 자동적으로 이루어질 수 있도록 하는 Interface를 개발하였다. 데이터베이스와 응용프로그램과의 연결은 이미 그 안정성이 입증된 ODBC(Open Database Connectivity)^{8,9)}를 사용하여 프로그램 코드 작성 과정이나 실행 중에 발생할 수 있는 오류의 가능성을 감소시켰으며 일반 윈도우 사용자라면 누구나 쉽게 사용 환경을 구축할 수 있도록 하였다.

2. 철근 콘크리트 구조설계 통합 시스템의 데이터 모델링

본 연구에서 구현된 철근 콘크리트 구조설계 통합 시스템을 위한 중앙 데이터베이스와 각 응용 모듈과의 전체적인 구성은 그림 1과 같다. 이와 같은 시스템에서는 중앙 데이터베이스와 각 단위 설계 모듈간의 능동적인 데이터 교환이 필요하며 각 단위 설계 모듈에서 다루는 데이터들은 매우 복잡하고 여러 단계에서 동일한 데이터들을 다루는 경우가 많기 때문에 데이터들의 중복을 최소화하고 효율적으로 모델링 하기 위해 체계적인 데이터의 분류가 필요하다.

본 연구에서는 실제 구조설계 과정을 분석하여 STEP (Standard for the Exchange of Product model data) part104에서 정의된 기본 요소들을 바탕으로 철근 콘크리트 구조물에 필요한 요소들을 확장, 보완하였으며 OMT를 사용하여 모델링하고 다음과 같은 규칙에 의하여 데이터를 분류하였다.

- 대상 구조물 자체의 정보를 가지는 데이터와 구조물에 영향을 미치는 환경정보에 관련된 데이터는 분리한다.
- 단위 설계 모듈별로 그 과정을 세분화한다.
- 각 단위 설계 모듈을 분석하여 입력, 출력에 사용되는 데이터를 구분한다.
- 사용자의 입력을 필요로 하는 데이터와 시스템의 판단을 필요로 하는 데이터를 구분한다.

데이터의 분류하여 객체군을 형성하기 위한 방법으로는 다음과 같은 방법⁵⁾을 사용하였다.

- 일반화/특수화(generalization/specialization)
하위 클래스는 상위 클래스에 대하여 'is_a' 관계이며 상위 클래스의 속성들을 상속받는다. 반대로 상위 클래스는 하위 클래스를 'include'하고 있다고 볼 수 있다. (예: Girder is_a element, elements include girder)
- 합성화/분해화(aggregation/decomposition)
상위 클래스는 하위 클래스들로 이루어져 있으며 하위 클래스들이 모두 사라지면 상위 클래스는 존재할 수 없다. 하위 클래스는 상위 클래스에 대하여 'a_part_of' 관계이며 상위 클래스는 하위 클래스에 대하여

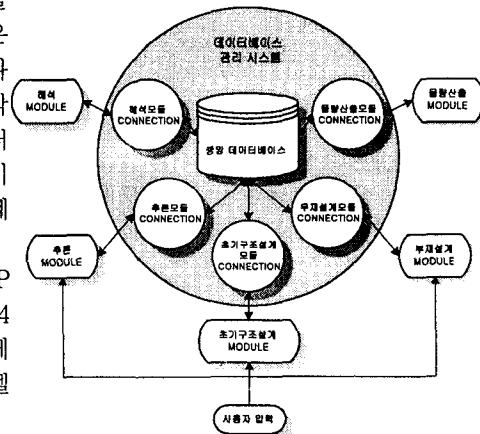


그림 1 구조설계 통합 시스템의 구성

'has_part'관계라고 할 수 있다. (예:Deadload is a_part_of loadcase, Loadcase has deadload)

- 클래스화/인스턴스화(classification/instantiation)

상위 클래스에서 어떠한 목적에 의하여 발생시킨 instance와의 관계이다. Instance는 상위 클래스의 모든 값과 속성을 상속받는다. (예:Column에서 만들어낸 외곽 column).

본 연구에서 사용된 표기방식을 그림 2에 표현하였다.

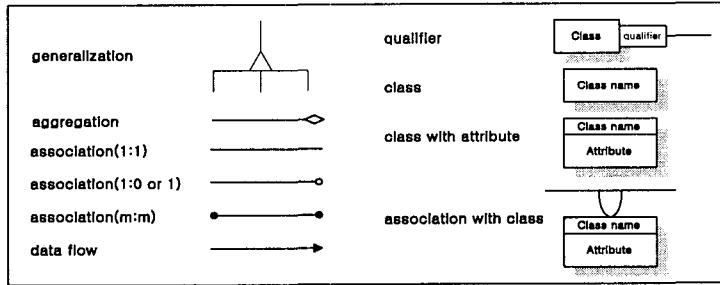


그림 2 본 연구에 사용된 표기양식

본 통합 시스템의 구현을 위한 모델링 및 표현 기법은 그 범위를 일반적인 모델링의 적용 범위보다 간소화하여 개념적인 정의보다는 실제 구현에 적합하도록 하였다.

크게 나누어 사용자의 입력작업이 주가 되는 초기 구조설계 모듈(Pre-processing module), 구조해석 모듈로의 데이터 변환과 구조해석, 구조설계를 위한 해석 결과값 추출, 부재 설계와 구조 계산서 작성, 개략적인 물량산출 까지를 그 범위로 하고 대상 구조물은 철근 콘크리트 저층 구조물로 한정 지었다.

실제 구조설계 과정을 중심으로 한 단계별 분류는 다음과 같다.

1) Project 단계

본 모델에서 최상위 객체는 Building이며 하나의 프로젝트를 진행함에 있어서 기본적인 정보들과 구조물에 영향을 미치는 환경요소들인 건물이 속한 지역, 지반 특성, 용도, 구조 형식 등에 관련된 데이터들을 저장한다. 위와 같은 환경요소들은 지진하중, 풍하중을 산정하는데 사용된다.

2) Structure 단계

건축물의 구조 시스템에 대한 정보를 다룬다. 즉 건축물을 구성하고 있는 기둥, 보, 벽, 슬래브, 기초 등의 부재에 대한 단면의 정보, 재료에 대한 데이터, 부재의 구속 조건, 지지상태, 건축물에서의 부재의 배치상태 등에 대한 데이터를 저장한다.

3) Load 단계

지역적 특성에 따라 정해지는 풍하중과 지진하중에 대한 데이터와 사용자의 정의에 의한 하중정보, 하중의 유형에 따른 분류와 각 하중들의 조합에 대한 정보를 저장한다.

4) Analysis 단계

Structure단계에서 입력된 구조물의 정보와 Load단계에서 정의된 하중에 따른 건축물의 구조해석 방법 및 하중 조건에 따라 해석을 수행한 후 각 부재의 변위와 반력등에 대한 정보를 저장한다.

5) Design 단계

대상 구조물의 해석이 끝난 후 그 결과값을 바탕으로 부재설계 모듈의 수행 후 설계결과에 따른 부재들의 최적화된 단면 형상과 배근 정보를 저장한다.

6) 물량 산출 단계

모든 단위 설계 모듈에서 저장된 결과값을 바탕으로 부재의 형상 정보등 필요한 정보만을 추출하여 개략적 산출 후 그 결과값을 저장한다.

시스템 전체의 모델링 결과는 그림 3에 나타나 있다. 이해를 돕기 위하여 각 분류 단계별로 구분 지어 표현하였다.

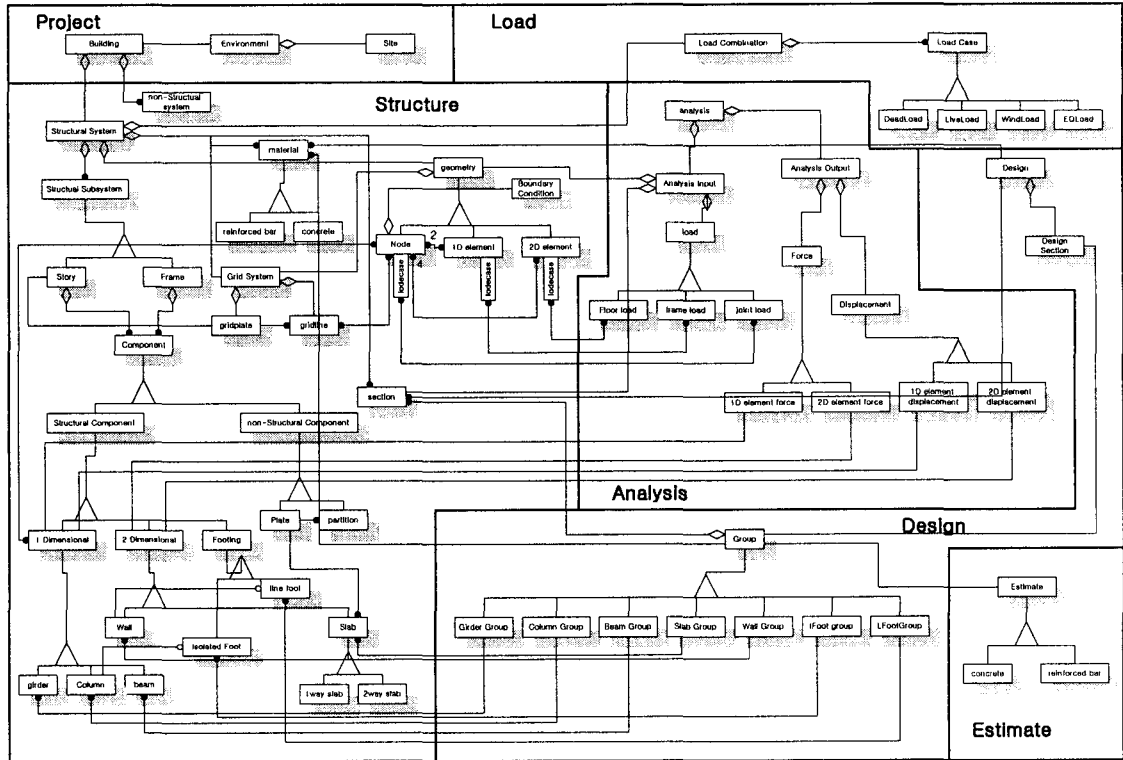


그림 3 데이터 모델의 전체 구조

3. 철근 콘크리트 구조설계 통합 시스템의 데이터베이스 스키마 모델링

데이터베이스 관리 시스템의 구현은 기본적으로 다음과 같은 세 가지 과정⁵⁾을 거친다.

- External schema : 데이터베이스와 연결된 단위 설계 모듈의 관점에서 디자인한 모델. 다른 데이터베이스의 변화와는 독립되어 있다. 즉 응용 프로그램별로 사용되는 데이터들의 구조라 할 수 있다. 그림 3에서 보여진 모델이 이에 해당한다.
- Conceptual schema : 필요한 데이터들을 데이터베이스에 매핑시키려 할 때 그 기획단계에서의 모델. 즉 구현되기 이전의 테이블 모델이며 각 단위 설계 모듈의 관계성을 정의하고 데이터베이스 관리 시스템의 특성들은 나타나지 않는다.
- Internal schema : Conceptual schema를 바탕으로 구축하는 해당 데이터베이스 관리 시스템의 한계와 형태에 대하여 다룬다.

그림 3에서 제안한 데이터 모델(External schema)을 바탕으로 Conceptual schema를 구현함에 있어서 응용 프로그램의 구현을 최종 목적으로 하기 때문에 단순 분류를 위한 객체는 데이터베이스 스키마로 변환하지 않았으며 수행 효율을 향상시키기 위하여 몇몇의 속성은 중복 사용하였다. (예:load는 deadload, liveload, windload, EQload를 통칭하는데 지나지 않으므로 구현하지 않음. 각 element들은 좌표를 가지고 있는 node를 포함하여 story의 검색이 가능하지만 각 부재들은 수행 효율의 향상을 위하여 story를 가진다.)

그림 3에서 제안한 분류를 사용하여 Oracle 7.3을 지원할 수 있는 데이터베이스 테이블 모델을 구현하였으

며 구현된 모델을 바탕으로 Internal schema를 SQL code로 작성하였다.

본 데이터베이스 모델을 구현함에 있어서 구조설계 과정 전반에 걸쳐 Entity-driven방식⁵⁾을 사용하였으며 이 방법은 응용 프로그램에 의미가 있으며 표현이 가능한 객체들을 찾아 관계를 정의하는 방법이다.

모델에 나타난 각 테이블간의 관계는 다음과 같은 방법^{5),7),8)}을 사용하여 표현하였다.

- 1) 속성을 지닌 각 객체들은 원칙적으로 하나의 테이블로 구성한다.
- 2) 일대일(1:1) 또는 일대다(1:m) 연관관계(Association) : 해당 테이블 자신은 Primary Key로서 ID를 갖고 Foreign Key로서 연결된 다른 테이블의 ID를 갖는다.
- 3) 다대다(m:m) 연관관계 : 서로 다른 두 테이블의 ID를 갖는 연관 클래스를 추가하여 연결관계를 정의한다. Association 클래스는 대부분 null이 아닌 Foreign Key로 구성되며 이 방법은 테이블 수의 감소와 수행 속도의 향상을 꾀할 수 있다.
- 4) 일반화(Generalization) : 하위 클래스에서 상위클래스의 ID를 갖는다. 모든 하위 클래스들의 속성을 상위 클래스로 올려 관리한다. 실제 속성들의 상속은 함수로써 구현한다.

데이터베이스 스키마를 테이블로 mapping한 그림들을 각 단계별로 표시함에 있어서 전체적인 구조를 설명하는데 크게 영향을 미치지 않는 객체들과 관계성의 표현은 생략하였다. 각 단위 설계 모듈별로 테이블의 사용 상황을 표 1에 나타내었다. 연관관계를 정의하거나 같은 종류의 테이블들은 표현에서 제외하였다.

표 1 각 모듈별 테이블 사용 여부

단위모듈 테이블	초기 구조 설계	구조 해석	부재 설계	물량 산출	단위모듈 테이블	초기 구조 설계	구조 해석	부재 설계	물량 산출
building	●	●	●		loadcombination	●	●		
site	●	●			loadcase	●	●		
soil	●	●			d1 concentrated load	●	●		
service	●	●			d1 distributed load	●	●		
story	●	●		●	usage	●			
gridplate	●				coating	●			
gridline	●				eqload		●		
node	●	●	●	●	windload		●		
d1element	●	●	●	●	d1element force		●	●	
d2element	●	●	●	●	d1element deformation		●	●	
column	●		●	●	d1element support		●	●	
girder	●		●	●	d2element force		●	●	
beam	●		●	●	release	●	●		
slab			●	●	column group			●	●
plate	●		●		girder group			●	●
isolated foot			●	●	beam group			●	●
line foot			●	●	slab group			●	●
material	●	●	●		wall group			●	●
section	●	●	●		iifoot group			●	●
concrete volume			●	●	lfoot group			●	●

3.1. 환경정보와 Structural Subsystem의 데이터베이스 스키마 모델링

본 모델에서의 최상위 객체는 Project이며 Project는 Foreign Key로서 Building ID를 갖는다. Building은 지역, 지반, 구조형식, 건물용도등 구조물의 전반적인 환경정보들을 가지며 Structural Subsystem으로 Story를 갖는다. Story는 시스템에서 구조물의 구성 요소들을 인식하기 위한 기반으로 Gridsystem과 연결되어 있다. 이 단계의 테이블의 구성은 그림 4에 나타나 있다.

3.2. 각 부재정보 데이터베이스 스키마 모델링

본 단계에서의 최 상위 객체는 Story이며 각 부재들은 일차원부재(선 부재), 이차원부재(면 부재), 기초로

구분된다. 실제 부재가 아닌 D1Element, D2Element, Plate는 각각 하위의 실제 부재들을 포함하며 구조물 형상 모델링 시에 필요한 정보들을 담고 있다. 실제 부재인 Column, Girder, Beam, Slab, Wall, IFoot, LFoot 등은 구조해석과 구조설계에 필요한 Material, Section을 가지며, ID외의 식별자로서 Name을 가지고 있다. 독립적이지 않고 다른 부재의 영향을 받는 부재들로는 Slab, Beam, Partition등이 있다. 이 단계에서의 테이블의 구성은 그림 5에 나타나 있다.

3.3. 하중정보 데이터베이스 스키마 모델링

최 상단의 Loadcombination은 최대 10개의 Loadcase를 가질 수 있으며 Loadcase는 Deadload, Liveload, Windload, EQload로 구성되어 있다. Deadload와 Liveload는 각 부재에 가해진 하중들의 합으로 이루어지며 지역과 지반의 영향을 받는 Windload, EQload는 Building에 연결되어 Site와 Soil테이블의 정보를 바탕으로 구성되어진다. 이 단계에서의 테이블의 구성은 그림 6에 나타나 있다.

3.4. 해석 모듈 데이터베이스 스키마 모델링

각 부재의 위치를 나타내는 정보들이 D1Element, D2Element, Node, Plate에 저장되며 각 부재별로 속성들을 가지고 있다. 본 모델에서는 실제 철근 콘크리트 해석 시에 잘 쓰이지 않는 판 요소의 구속조건은 모두 고정으로 간주하여 해석을 하고 난 후 반력은 각각 D1_ELE_FOR와 D2_ELE_FOR에 저장이 되고 Node에 연결된 Release의 조건에 따라 D1_ELE_SUP에 지지조건이 저장되며 해석후 계산된 처짐값은 D1_ELE_DEF에 저장된다. 이 단계에서의 테이블의 구성은 그림 7에 나타나 있다.

3.5. 부재설계 결과 데이터베이스 스키마 모델링

사용자의 선택에 의해 각 부재별, 또는 그룹별 설계가 가능하며 해석 결과값을 바탕으로 설계하여 그 결과값을 Group, Section, Bend테이블에 저장한다. 이 단계에서의 테이블의 구성은 그림 8에 나타나 있다.

3.6. 물량산출 데이터베이스 스키마 모델링

개략적산을 위한 콘크리트의 물량은 그룹별 단위 물량을 산출하여 그 값이 ConVolume에 저장되고 각 부재들과 다대다 연관관계를 갖는다. 최종적으로 단위 물량에 그 개수를 곱하여 최종 콘크리트 량을 계산한다. 이 단계에서의 테이블의 구성은 그림 9에 나타나 있다.

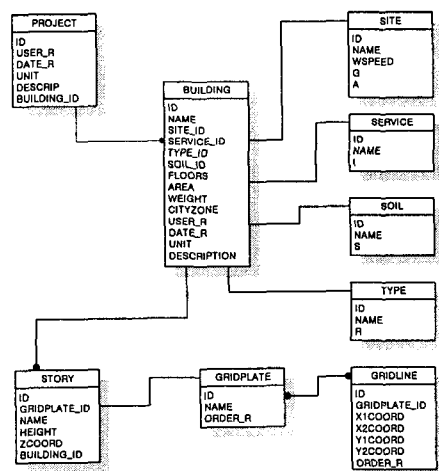


그림 4 건물, 층, 그리드 정보 테이블

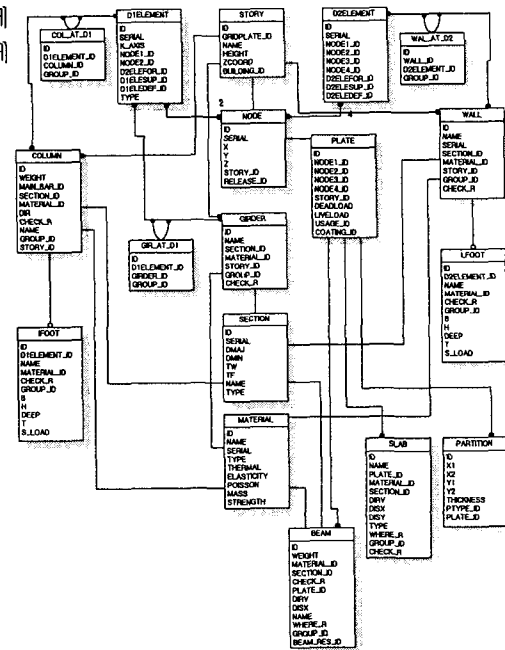


그림 5 부재 정보 테이블

4. 응용 프로그램과 데이터베이스간의 Interface 작성

본 연구에서는 객체지향 모델링 방법을 사용하여 모델링한 후 데이터베이스 모델로 이식시키는 작업을 수행하였다. 모델의 특성을 그대로 데이터베이스 모델에 반영하는 점에 있어서는 객체지향 데이터베이스를 사용하는 것이 바람직하지만 객체지향 데이터베이스는 매우 고가이고 표준이 정해지지 않아서 시스템간의 호환성과 다른 데이터베이스로의 이식성이 관계형 데이터베이스에 비하여 떨어지며 데이터베이스 관련 응용 프로그램을 위한 제작 환경이 불편하다. 따라서 본 연구에서는 이미 표준화 되어있고 현재 널리 쓰이고 있는 관계형 데이터베이스 관리 시스템인 Oracle 7.3을 사용하였다. 객체지향 개념의 모델은 그 특성상 관계형 데이터베이스로도 이식이 가능하다. 데이터베이스와 응용 프로그램과의 연결은 다른 시스템과의 호환성이나 이식성을 고려하여 Microsoft사에서 제안한 ODBC(Open DataBase Connectivity)를 사용하였고 관계형 데이터베이스 테이블의 객체지향 클래스화^{11),12)}작업에 있어서는 객체지향 언어를 이용하여 관계형 데이터베이스의 단점을 보완하는 코드를 작성하는 방식을 취했다. 단위 설계 모듈을 제작하기 위한 프로그래밍 언어로는 객체지향 개념을 구현할 수 있고 윈도우 환경에 최적화 되어 있는 Visual C++ 5.0을 사용하였다.

4.1. 관계형 데이터베이스와 객체지향 프로그래밍 언어

관계형 데이터베이스와 객체지향 언어는 각각 다음과 같은 특징을 가지며 각각의 단점을 서로 보완하여 이상적인 조합을 만들어 낼 수 있다. 그 특징들은 표 3와 같으며 본 연구에서는 관계형 데이터베이스에서는 지원하지 않는 객체지향 개념들은 각 단위 설계 모듈에 함수로써 구현하였다.

표 3 관계형 데이터베이스와 객체지향 언어의 비교

	관계형 데이터베이스	객체지향 프로그래밍 언어
대용량 데이터 관리	우수	미흡
데이터 지속성	우수	없음
데이터 검색	우수	미흡
데이터 조작	미흡	우수
복합 객체의 관계 표현	없음	우수

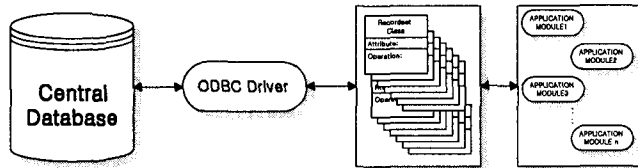


그림 10 시스템과 ODBC와의 관계

4.2. ODBC

데이터베이스 내부에는 매우 복잡한 알고리즘들로 구성되어 있어 그 사용법을 완벽히 이해하지 않는 이상 데이터의 유실과 전체적인 흐름의 충돌 등을 피할 수 없다. 이러한 난점을 극복하기 위해 Microsoft사에서는 ODBC라는 데이터베이스 연결 드라이버를 발표하였다. 일반 응용 프로그램에서는 이 ODBC함수를 호출하고 ODBC에서는 내부의 데이터베이스 지원 함수의 세트로부터 적절한 함수를 호출하여 데이터베이스에 명령을 내리게 되는 것이다. 본 연구에서는 ODBC함수 호출을 위하여 레코드셋 클래스^{9),11)}를 작성하고 개별 모듈의 실행만으로 데이터베이스를 관리할 수 있도록 프로그래밍 하였다.

시스템과 ODBC와의 관계를 개략적으로 살펴보면 그림 10과 같다.

중앙 데이터베이스는 ODBC 드라이버를 통하여 데이터베이스 테이블을 클래스화 시킨 레코드셋 클래스와 연결되어 있고 응용 프로그램과는 직접 연결되어 있지는 않다. 응용 프로그램 모듈 또한 단지 레코드셋 클래스와 연결되어 있다. 레코드셋 클래스는 ODBC와 연결되어 데이터베이스를 관리하는 함수들만을 포함하며 일반 응용프로그램 모듈에서 레코드셋 클래스에 접근하여 필요로 하는 데이터의 생성, 검색, 수정, 삭제 등의 명령을 내릴 경우에 그에 해당하는 명령을 데이터베이스 관리시스템의 명령으로 변환하여 데이터베이스에 전달한다. 따라서 응용 프로그램 모듈과 레코드셋 클래스간에는 중복되는 내용이 없고 어떠한 시점이나 상황에서 데이터베이스에 접근할 수 있기 때문에 각 단위 설계 모듈과는 별도로 개발이 가능한 장점이 있다.

4.3. 통합 구조설계 시스템에의 적용

레코드셋 클래스는 ODBC 드라이버를 지원하도록 MFC(Microsoft Foundation Class library)⁹⁾에서 구현해 놓은 CRecordset클래스를 상속받는다. 각각의 테이블의 attribute는 생성한 레코드셋 클래스에서 동일한 자료형의 변수로 지정하고 각각의 위치에서 데이터의 변화가 있을 때에는 즉시 상대방에 반영하도록 하는 함수를 구현하여 능동적인 데이터교환이 이루어질 수 있도록 하였다.

각 단위 설계모듈에서는 레코드셋 클래스의 객체를 생성하며 객체접근을 통하여 필요한 값을 가져오거나 함수를 호출하게 된다. 단위 설계 모듈에서 데이터베이스에 있는 데이터를 사용할 필요가 있으면 해당 데이터가 존재하는 테이블을 Open하고 함수를 실행하며 원하는 결과를 얻었다면 테이블을 Close하고 다음 작업을 수행한다.

레코드셋 클래스에는 기본적으로 다음 네 가지의 기능을 함수로서 구현하였다.

- 데이터의 추가 : 사용자의 요구에 의한 생성, 시스템의 판단에 의한 생성(예: 사용자에 의해 부채를 생성 하라는 명령이 있으면 노드는 자동으로 생성한다.)이 있으며 데이터베이스의 유일성 제약조건에 위배되지 않도록 하였다.
- 데이터의 수정 : 사용자의 요구에 의한 수정.
- 데이터의 검색 : ID또는 속성값으로 검색하며 ODBC에서 제공하는 Filtering(메모리 상에 조건에 맞는 데이터만을 남긴다.)^(7,11)을 사용한다. 조건을 함수의 인자로 지정하여 쉽게 사용할 수 있도록 하였다.
- 데이터의 삭제 : 사용자의 요구에 의한 삭제. 어떤 데이터를 삭제할 때 해당 데이터를 Foreign Key로써 가지고 있는 데이터가 존재한다면 검색하여 삭제한다.

본 연구에서는 통합 시스템에서 사용되는 테이블 전체를 하나로 묶어서 관리하며 각 단위 설계 모듈별로 필요한 테이블들을 불러 사용하도록 하였다. 구현된 철근 콘크리트 구조설계 통합 시스템의 개략적인 모습을 그림 11에 나타내었다. 각 모듈별로 데이터의 이동 경로를 표시하였으며 클래스의 세부적인 설명은 포함하지 않았다.

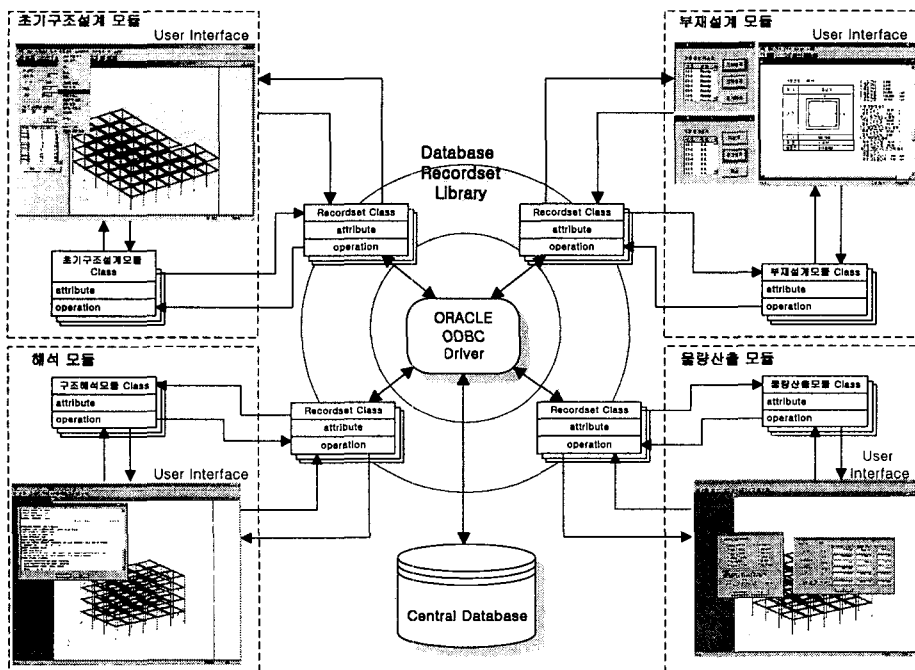


그림 11 통합 구조설계 시스템에 적용된 데이터베이스

5. 결 론

본 연구는 실제 구현 가능한 철근 콘크리트 구조설계 통합 시스템에 적합한 데이터베이스 모델의 제시하고 사용자의 작업을 최소화한 데이터베이스 관리 시스템을 구현하는데 초점을 맞추었다. 본 연구를 통하여 다음과 같은 결론을 얻었다.

- 1) 통합 구조설계 과정을 분석한 결과 객체지향 모델로 적절히 표현 가능함을 알게되었으며 객체지향모델의 관계형 데이터베이스로의 이식도 충분히 가능한 것으로 판명되었다.
- 2) 철근 콘크리트 구조설계 통합 시스템(INDECON)에 데이터베이스를 도입함으로써 전 구조설계과정의 데이터의 일원화를 통하여 사용자의 작업 감소와 데이터의 재사용시 효율성을 증대시킬 수 있다.
- 3) 객체지향 개념을 사용하여 후에 새로운 단위 설계 모듈의 추가가 용이하다.
- 4) ODBC를 사용하였기 때문에 객체지향 개념을 구현하기 위해서는 프로그램 소스코드의 양이 많아지는 단점이 있어 객체지향 데이터베이스의 호환성과 이식성이 향상된다면 객체지향 데이터베이스를 사용하는 것이 바람직하다.
- 5) 각 단위 설계 모듈에서 데이터베이스를 사용할 때 물리적인 저장공간을 액세스하기 때문에 수행속도를 향상시킬 필요가 있어 이에 대한 개선이 필요하다.

감사의글

본 논문은 한양대학교 초대형 구조 시스템 연구센터와 한국전력 연구원의 세부과제에 대한 연구비 지원으로 수행되었으며 이에 감사드립니다.

6. 참 고 문 헌

1. 안은경, 천진호, 신동철, 이병해, "철근 콘크리트 구조의 초기 구조설계 시스템 개발," 한국건축학회 논문집 14권 4호, 1998
2. 이진우, 김남희, 이병해, "통합 시스템 구축에 있어서 RC부재설계 모듈 개발," 한국전산구조공학회 학술발표회 논문집 제11권 제1집, 1998
3. 이병해 외, 실무용 구조설계 전문가 시스템 개발, 한국전력 연구원 3차년도 1/4분기 진도보고서, 1998
4. Ajay Lavakare, "Structural Steel Framing Data Model," *CIFE TECHNICAL REPORT*, No. 012, Stanford University, 1989
5. James Rumbaugh 외, *Object-Oriented Modeling and Design*, Prentice-Hall, 1991
6. 한국 오라클, *SQL 기초*, 한국 오라클 주식회사, 1995
7. Oracle, *SQL Plus User's Guide and Reference ver 3.0*, Oracle, 1991
8. Robert Gryphone, *이것이 ODBC 2 다*, 도서출판 삼각형, 1996
9. Microsoft Press, *MFC Library Reference, part 1,2,3*, 도서출판 삼각형, 1997
10. David Lockman, *Oracle 8 Database Development*, 인포북, 1998
11. Al Stevens, *C++ Database Development*, 2nd Edition, MIS Press, 1994
12. 강맹규, *Object-Oriented Programming*, 한양대학교 대학원 산업공학과, 1997
13. 이상엽, *Visual C++ Programming Bible ver 5.x*, 영진 출판사, 1997