

개체형 통합설계모델의 객체지향 프로그래밍 Object-Oriented Programming of Entity-Based Integrated Design Model

이 창 호* 김 진 근**
Lee, Chang-Ho Kim, Jin-Guen

ABSTRACT

An entity-based integrated design product and process model uses product and process entities to describe design information and design activities, respectively. The concepts and notation for product and process entities in the entity-based integrated design model are similar to the concepts of object-oriented programming languages such as C++ and Smalltalk. This paper uses C++ to program an entity-based integrated design model for building frames structures. The design information and activities involved in the three dimensional building space, the locations of frames, and the grouping of frames represented as entities in the entity-based integrated design model are transformed to C++ codes. Each product or process entity can be basically transformed to a class. The attributes of an entity can be defined as variables and member functions of a class.

1. 서 론

빌딩과 같은 구조물을 설계하는데 있어서 초기 설계 단계에는 대개 여러 가지 설계 대안들을 고려하고, 설계가 진행됨에 따라 가장 적합한 한가지 설계안을 택하게 된다. 이에 따라서 구조설계도 초기 단계에는 여러 가지 구조시스템 등을 고려해야하고, 프레임시스템과 같은 하나의 구조시스템이 결정된 후에도 프레임의 배치와 같은 여러 사항들을 결정해야한다. 구조설계가 진행됨에 따라서 구조해석을 실시하며, 기둥과 보와 같은 기본구조요소들의 크기 등을 결정한다. 구조물이 고층화되고 대형화될수록 구조설계 과정은 복잡해지게 된다. 구조해석과 부재설계 등의 구조설계 작업들은 컴퓨터의 도움으로 처리할 수 있지만 초기 구조설계 단계에서는 많은 부분이 컴퓨터로 처리하기 어려운 상태로 있다. 이는 초기설계와 관련된 설계정보(design information)와 설계작업(design activity)이 정형화 되어있지 않기 때문인데 이것을 해결하기 위한 목적으로 사용되는 것이 설계모델(design model)이다.⁽¹⁾⁻⁽³⁾ 설계모델은 설계정보와 설계작업을 정해진 표기법에 따라서 질서정연하게 표현하여 컴퓨터시스템의 개발에 도움을 주기 위한 것이다.

개체형 통합설계모델(entity-based integrated design product and process model)은 설계모델의 하나로서 프로덕트 개체(product entity)와 프로세스 개체(process entity)를 포함하는데, 프로덕트 개체는 설계정보를 나타내고 프로세스 개체는 설계작업을 나타낸다.⁽⁴⁾⁻⁽⁵⁾ 개체형 통합설계모델의 개체 표기법은 C++ 또는 Smalltalk 등의 객체지향 프로그래밍 언어(object-oriented programming language)의 개념과 유사한 점이 많

* 정희원 · 한경대학교 건축학부 조교수

** 한경대학교 건축학부 석사과정

기 때문에 이러한 언어들을 이용하여 컴퓨터 시스템을 개발하는 것이 유리한데,^{(4)~(8)} 본 논문에서는 C++로 프로그래밍하기 위한 방법을 다루었다. 먼저 빌딩 프레임구조물의 공간정보와 교차하는 프레임간의 정보 전달을 위한 개체형 통합설계모델을 소개한다. 이를 C++로 프로그래밍 하였고, 실행 예를 통하여 프로그래밍 방법상의 여러 가지 사항에 대하여 분석하였다. 그리고 향후 보다 광범위하고 상세한 프로그래밍을 위하여 고려해야할 사항을 설명하였다.

2. 빌딩 프레임구조물을 위한 개체형 통합설계모델

간단한 형태의 빌딩 프레임구조물을 설명한 후에, 이러한 빌딩 프레임구조물의 설계와 관련된 설계정보와 설계작업을 개체형 통합설계모델을 이용하여 표시하였다.

2.1 빌딩 프레임 구조물내의 프레임

그림 1은 빌딩 프레임구조물의 3차원 공간을 보여주고 있다. 이 3차원 공간은 x, y, z 세 방향의 좌표축 상에서 정의되며 각 좌표축상의 그리드 라인으로서 구분된다. 그림 2는 4층의 빌딩 프레임구조물의 2층 평면을 나타낸다.⁽⁸⁾ 프레임의 위치는 굵은 점선으로 표시되었다. 프레임에 속한 기둥은 사각형으로 표시되었고 다른 기둥은 원형으로 표시되었다. 평면에는 여러 개의 프레임이 있고, 일부의 프레임은 교차한다. 그림 3은 교차하는 프레임 1, 프레임 2, 프레임 3을 나타내고 있다. 프레임 1과 프레임 2는 위치 1에서 교차하고, 교차점에 있는 기둥은 두 프레임의 요소가 된다. 프레임 2와 프레임 3은 위치 2에서 교차하고, 교차점에 있는 기둥은 두 프레임의 요소가 된다. 각 프레임에 속한 기둥은 그 프레임의 일부로 설계되기 때문에 한 프레임에서의 기둥설계는 다른 프레임에서의 기둥설계와 상호관련이 있다.

2.2 공간정보와 프레임설계를 위한 모델

그림 1, 2, 3에서 나타난 빌딩 프레임구조물의 설계와 관련된 설계정보와 설계작업은 개체형 통합설계모델로 표현될 수 있다. 개체형 통합설계모델은 프로덕트 개체와 프로세스 개체를 포함하는데 프로덕트 개체는 설계정보를 나타내고 프로세스 개체는 설계작업을 나타낸다. 빌딩 프레임구조물의 설계와 관련된 프로덕트 개체군(product entity category)과 프로세스 개체군(process entity category)이 그림 4에 나타나 있다.⁽⁸⁾ 여기서 개체군(entity category)이란 여러 가지 개체(entity)가 생성될 수 있는 군(category)을 말한다. 그림에서 사각형으로 표시된 것들이 프로덕트 개체군 또는 프로세스 개체군이다. 각 개체군의 이름은 사각형 안에 있다. 예를 들어 그림4(a)의 building coordinates는 하나의 프로덕트 개체군이다. 각 개체군의 속성은 사각형 아래에 수평선과 함께 나열되어 있다.^{(4)~(5)} 상하 한 쌍의 점은 속성이 생략되었음을 의미한다. 만일 속

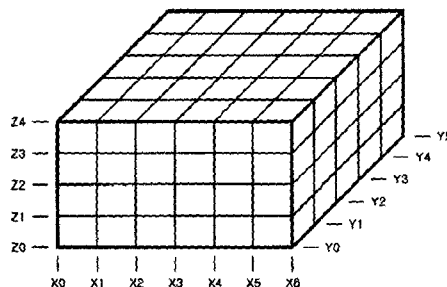


그림 1. 빌딩 프레임 구조물의 3차원 공간

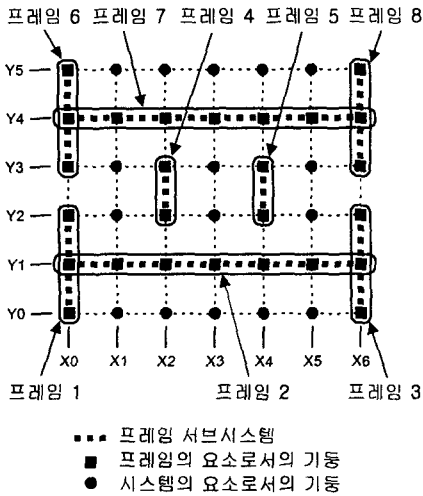


그림 2. 평면상의 프레임 배치

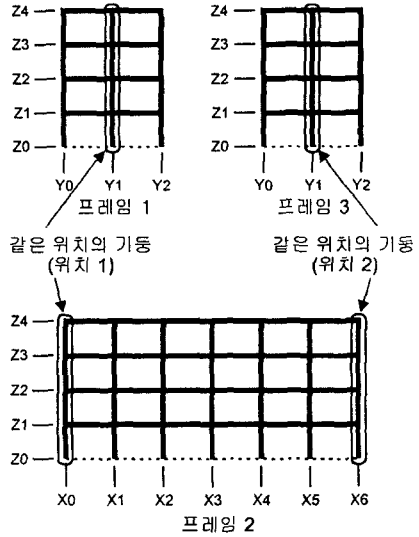


그림 3. 세 개의 교차하는 프레임

성이 단일 값을 갖는(single-valued) 속성이면 수평선의 끝이 흰색 원이 되고, 속성이 복수 값을 갖는(multiple-valued) 속성이면 수평선의 끝이 검은색 원이 된다. 속성의 종류표시는 괄호 속에 있고, 속성의 값의 형태는 모난 괄호([])에 표시되어 있다. 속성의 종류표시는 여러 가지가 있는데 그중 일부가 그림 4에 사용되었다. “B-DVA(base data-valued attribute)”는 속성의 값이 숫자 또는 문자가 되고, “B-OEVA (base object entity-valued attribute)”는 속성의 값이 프로덕트 개체가 되며, “B-AEVA(base activity entity-valued attribute)”는 속성의 값이 프로세스 개체가 되는 것을 의미한다.⁽⁴⁾⁻⁽⁵⁾

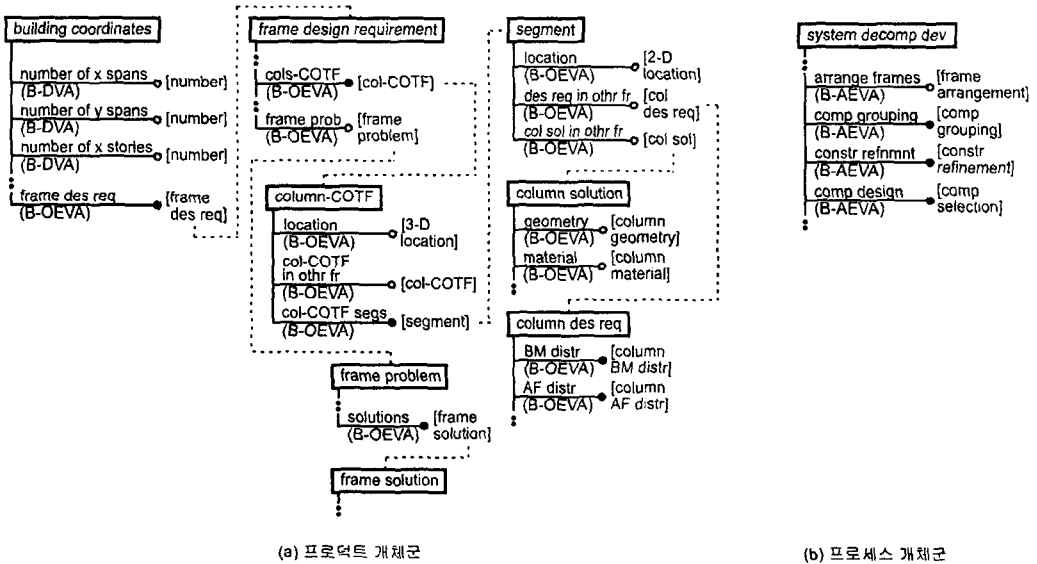


그림 4. 프로덕트 개체군과 프로세스 개체군

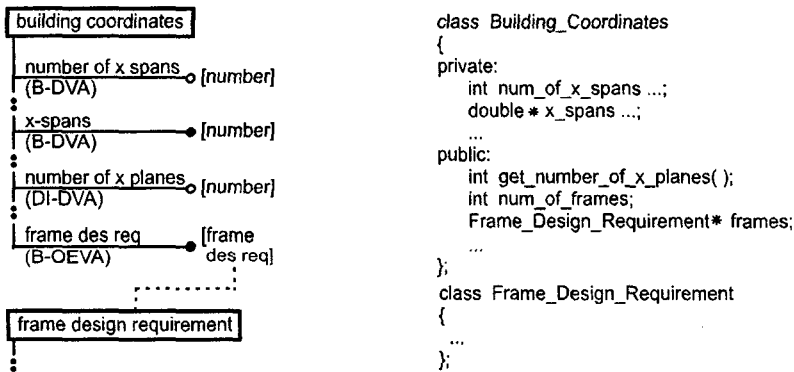
그림 4에서의 building coordinate 개체군은 빌딩 프레임의 3차원 공간을 표현한다. frame design requirement 개체군은 프레임 설계를 위한 개체군을 나타내며 column-COTF(column as components of two frames) 개체군은 두 개의 교차하는 프레임간의 상호작용을 나타내기 위한 것이며 이와 관련된 다른 프로덕트 개체군이 그림 4(a)에 표시되어 있다. 그림 4(b)의 system decomposition development 개체군은 프레임의 배치를 정하고 각 프레임을 보와 기둥 등의 요소로 분해하는 설계작업을 나타낸다.⁽⁸⁾

3. 개체형 통합설계모델의 프로그래밍

개체형 통합설계모델의 구성이 객체지향 프로그래밍 언어인 C++와 유사하기 때문에 기본적으로 하나의 개체는 하나의 클래스(class)로 프로그래밍 하였다. 프로덕트 개체의 프로그래밍, 프로세스 개체의 프로그래밍, 그리고 프로덕트 개체와 프로세스 개체의 연결에 대한 프로그래밍 방법에 대한 예가 각각 그림 5, 6, 7에 나타나 있다.

3.1 프로덕트 개체의 프로그래밍

그림 5는 프로덕트 개체를 C++ 클래스로 프로그래밍 하는 예를 보여주고 있다. 그림 5(a)에서 building coordinates 개체는 그림 5(b)에서 Building_Coordinates 클래스로 바뀌었다. 속성 값이 하나의 숫자가 되는 number of spans 속성은 integer 형의 변수로 바뀌었으며, 속성 값이 복수인 x-spans 속성은 double의 포인터형 변수로 바뀌었다. number of x planes 속성은 그 값이 number of x spans 속성 값에 1을 더한 수로서 결정이 되기 때문에 내부유도(internally derived, DI) 속성으로 표시되었는데, 이 속성은 get_num_of_x_planes() 라는 멤버 함수로서 프로그래밍 되었다. 그리고 building coordinates 개체는 frame design requirement 개체를 포함하고 있다. frame design requirement 개체는 Frame_Design_Requirement 라는 클래스로 선언된 후에 이 클래스형의 변수가 frames로 Building_Coordinates 클래스에 정의되었다. 이때 num_of_frames 라는 변수도 선언하여 프레임의 수를 저장하도록 처리하였다.



(a) 프로덕트 개체군

(b) C++ 클래스

그림 5. 프로덕트 개체와 C++ 클래스

3.2 프로세스 개체의 프로그래밍

그림 6은 프로세스 개체를 C++ 클래스로 프로그래밍 하는 예를 보여주고 있다. 그림 6(a)에서 system

decomposition development 개체는 프레임의 배치를 결정하는 arrange frames 속성을 포함하고 있으며, 비슷한 설계조건을 가진 프레임들을 그룹화하는 component grouping 속성도 포함하고 있다. 이러한 속성들은 그림 6(b)의 System-Decomp-Development 클래스 내에서 arrange_frames()와 determine_frame_groups()의 멤버함수로 바뀌어졌다. 그림 6(a)에서 arrange frames 개체는 put number of frames 속성과 locate each frame 속성들은 그림 6(b)에서 각각 put_num_of_frames() 함수와 for loop로 프로그래밍 되었다.

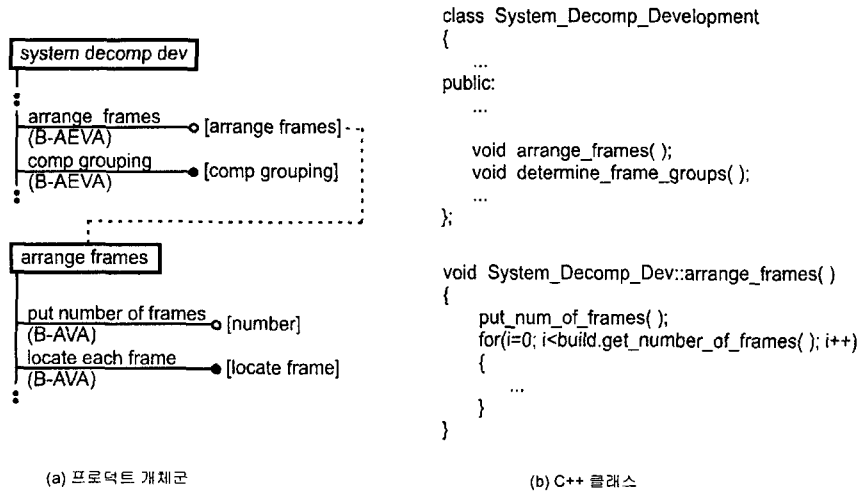


그림 6. 프로세스 개체와 C++ 클래스

3.3 프로덕트 개체와 프로세스 개체의 연결의 프로그래밍

개체형 통합설계모델에서 모든 설계과정은 프로세스 개체들을 통하여 진행이 된다. 프로세스 개체는 프로덕트 개체를 생성시키며 생성된 프로덕트 개체는 또 다른 프로세스 개체에 의하여 이용된다. 따라서 프로세스 개체에서 프로덕트 개체를 컨트롤하기 위한 연결이 있어야 한다. 이를 위하여 개체형 통합설계모델에서 프로덕트 개체를 프로세스 개체의 한 속성으로 정의를 해 둔다.^{(4)~(5)} 그림 7(a)에서 프로덕트 개체인 building coordinate 개체가 프로세스 개체인 system decomp development 개체의 한 속성으로 표시되었다. 이러한 프로덕트 개체와 프로세스 개체의 연결을 그림 7(b)에서 Building_Coordinates 클래스형의 변수를 System-Decomp-Development 클래스에 포함시키는 것으로 처리하였다. 그리고 모든 설계과정이 프로세스 개체를 통하여 이루어지기 때문에 그림 7(b)의 main 함수 내에서는 System-Decomp-Development 클래스의 객체로 정의된 sys의 함수들인 sys.arrange_frame()과 sys.determine_frame_groups() 등을 실행시키는 것으로 프로그래밍 하였다.

4. 시스템의 실행 예

앞에서 기술된 개체형 통합설계모델의 프로그래밍 방법에 따라서 프로그래밍을 하였고 이를 간단한 빌딩 프레임구조물에 대하여 적용하였다. 그리고 시스템을 프로그래밍하고 실행하면서 분석된 주요 내용을 정리 하였다

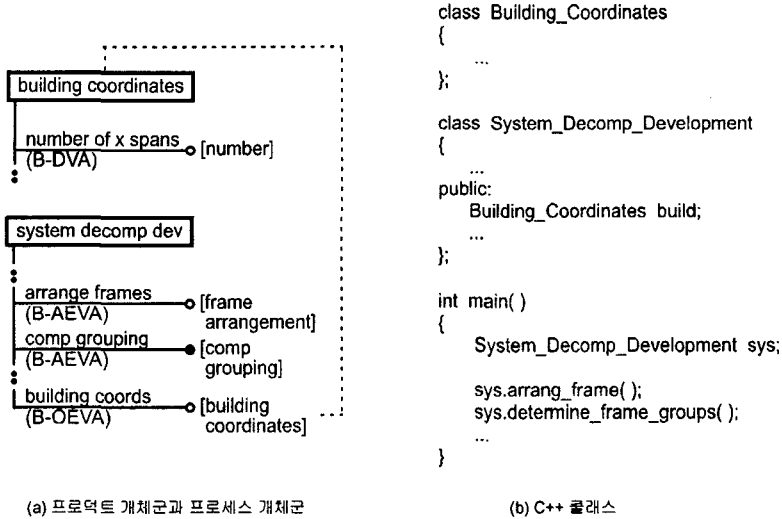


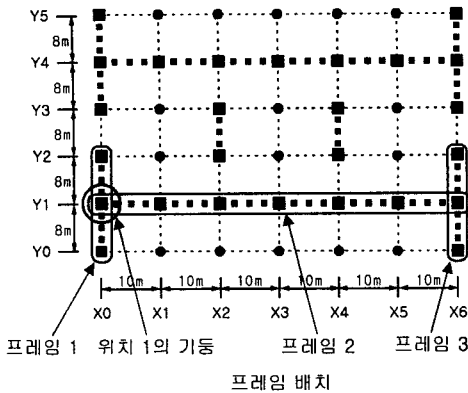
그림 7. 프로젝트 개체와 프로세스 개체를 위한 C++ 클래스

4.1 입력 및 출력

시스템의 실행 예가 그림 8에 나타나있다. 그림 8(a)는 실제의 구조물의 설계과정에서 세 과정을 표현한 것이다. 그림 8(b)는 시스템의 입출력을 나타낸 것인데 입력내용은 굵은 글씨로 표현되었다. 그림 2에서는 8개의 프레임이 나타나 있지만 그림 8의 실행 예에서는 3개의 프레임만을 대상으로 하였다. 그림 8(a)에서는 프레임 1, 프레임 2, 프레임 3의 배치가 나타나 있다. 3개의 프레임이 교차하는데 이중에서 프레임 1과 프레임 2의 교차가 나타나 있고, 프레임 1과 프레임 3은 설계과정에서 그룹이 되어 있다는 내용이 있다. 그림 8(b)에서는 먼저 프레임 1의 위치에 대한 내용을 입력하도록 되어있다. 프레임 2와 프레임 3도 같은 방법으로 입력이 되며 이것이 끝난 후에는 프레임의 교차에 대한 내용과 프레임의 그룹에 대한 내용을 입력하도록 되어있다. 이러한 입력과정이 끝나면 각 프레임의 좌표가 계산되며 교차위치에 대한 정보도 각 프레임 별로 계산이 되어 표시가 된다. 그리고 나서 설계를 진행하는데 하나의 그룹으로 정해진 프레임 1과 프레임 3은 가상의 2차원 좌표 상에서 설계를 진행하게 되므로 이에 대한 좌표가 계산되어서 표시되었다.

4.2 결과의 분석

본 연구에서 구현된 프로그램은 이미 개발된 개체형 통합설계 모델을 기본으로 하여 클래스를 정의하고 각 클래스 내의 변수와 함수들을 정의하는 방법을 사용하였다. 개체형 통합설계모델에서 언급되지 않은 알고리즘 등은 따로 개발하여 적용하였다. 그리고 공간 정보를 처리하기 위하여 다중 포인터를 많이 사용하였는데 이는 각 요소의 절점들을 관리하는데 효과적이었다. 그러나 최상위 클래스로부터 여러 단계의 클래스를 거쳐서 해당 객체의 정보를 찾기 위해서는 변수의 이름이 지나치게 길어지는 문제점도 발생하였다. 그리고 개발된 프로그래밍의 실행은 숫자와 문자로만 이루어지기 때문에 이를 윈도우 환경에서 그래픽을 사용하여 진행되도록 하여야 할 것으로 판단된다.



Location information of frame 1

Locating axis of frame:

1. y-z axis
2. z-x axis

Select axis: **1**

-Plane number for frame: **0**

-Number of left plane: **0**

-Number of right plane: **2**

.....

Intersection information of frame 1

-Number of intersections: **1**

-Intersecting frame number: **2**

-Plane numbers of intersection location(x, y): **0 1**

.....

Information of frame grouping

-Frame group number: **1**

-Number of frames in a group: **2**

-Frame numbers in a group: **1 3**

Coordinates of frame 1

0, 0, 13	0, 8, 13	0, 16, 13
0, 0, 10	0, 8, 10	0, 16, 10
0, 0, 7	0, 8, 7	0, 16, 7
0, 0, 4	0, 8, 4	0, 16, 4
0, 0, 0	0, 8, 0	0, 16, 0

Number of intersections: 1

Intersecting frame number: 2

Intersecting location: location 1

0, 8, 13
0, 8, 10
0, 8, 7
0, 8, 4
0, 8, 0

.....

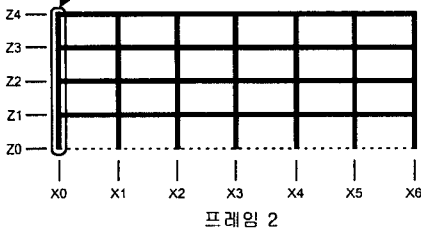
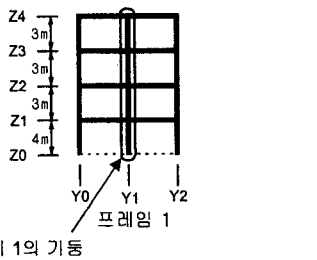
Which one do you want to design?

1. Grouped frame
2. Non-grouped frame

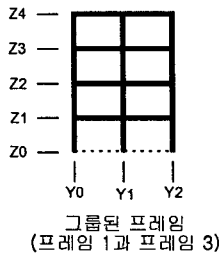
Select number: **1**

Frame group number: **1**

0, 13	8, 13	16, 13
0, 10	8, 10	16, 10
0, 7	8, 7	16, 7
0, 4	8, 4	16, 4
0, 0	8, 0	16, 0



교차하는 프레임



(a) 구조물

(b) 시스템의 입출력

그림 8. 시스템의 실행 예

5. 결론

본 논문은 구조설계를 컴퓨터로 처리하기 위한 목적으로 사용되는 설계모델의 하나인 개체형 통합설계모델을 객체지향 프로그래밍 언어인 C++로 프로그래밍 하기 위한 방법을 기술하였다. 빌딩 프레임구조물의

설계과정 중에서 3차원 공간처리와 프레임의 교차정보와 프레임의 그룹 등에 대하여 프로그래밍 방법을 살펴 보았다. 프로그래밍 과정과 실행 예를 통하여 얻어진 결론은 다음과 같다.

1) 개체형 통합설계 모델의 프로덕트 개체는 기본적으로 하나의 C++ 클래스로 바뀔 수 있다. 숫자나 문자를 속성의 값으로 갖는 기본 속성은 해당 데이터형의 변수로 정의하면 되고, 다른 개체를 속성의 값으로 갖는 기본 속성은 해당 클래스형 변수로 정의하면 된다. 다른 개체로부터 그 속성의 값이 계산되는 유도 속성은 그 유도과정을 포함하는 멤버 함수로 정의할 수 있다.

2) 개체형 통합설계 모델의 프로세스 개체도 하나의 C++ 클래스로 바뀔 수 있다. 프로세스 개체가 포함하는 여러 가지 설계작업들은 클래스 내의 멤버함수로 정의할 수 있다. 그리고 프로덕트 개체와의 연결을 위하여 필요한 프로덕트 개체에 해당하는 클래스형의 변수를 프로세스 개체에 포함시켜야 한다.

3) 개체형 통합설계 모델은 개체가 여러 계층에 걸쳐있는 복잡한 구성을 갖고 있는데 이를 클래스 내의 변수를 통하여 연결하기에는 복잡한 문제가 따르게 된다. 이를 클래스의 상속관계 등의 특징을 사용하여 처리하도록 변경하는 것이 바람직한 것으로 판단된다.

4) 프로그램 사용자의 편의를 위하여 윈도우 환경에서의 그래픽 처리가 필요하다.

5) 본 연구에서 취급하지 않은 개체형 통합설계모델의 개체간의 일반화(generalization) 등의 관계도 효율적으로 프로그래밍 해야 한다.

6) 시스템의 확장을 위해서는 구조해석 프로그램과 같은 다른 소프트웨어와의 연결이 필요하며, 시스템 자체 데이터베이스의 구축에 대한 고려도 필요하다.

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. Park, H., and Cutkosky, M. R. "Framework for Modeling Dependencies in Collaborative Engineering Processes. Research in Engineering Design", Vol. 11, 1999, pp. 84-102
3. 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
4. Hong, N. K. and Hong, S. H., "Entity-Based Models for Computer-Aided Design Systems," *Journal of Computing in Civil Engineering*, ASCE, 12(1), pp.1-12, 1998
5. 김남희, "통합 구조설계 시스템 개발 방법에 대한 연구," 대한건축학회논문집 제 13권 제 12호, 1997
6. Sause, R., Martini, K., and Powell, G. H., "Object-Oriented Approaches for Integrated Engineering Design Systems," *Journal of Computing in Civil Engineering*, ASCE, 6(3), pp.248-265, 1992
7. Lee, C.-H., Sause, R., and Hong, N. K., "Overview of Entity-Based Integrated Design Product and Process Models," *Advances in Engineering Software*, Vol. 29, No. 10, Elsevier Science, United Kingdom, pp.809-823, 1998
8. 이창호, "상호 의존적인 구조설계정보의 표현을 위한 모델", 대한건축학회 논문집, 제15권 제4호, 1999, pp. 61~72