

BIM/IFC 파일로부터 건물요소의 형상모델 추출에 관한 연구

Extracting Building Element Geometry from BIM/IFC Physical Files

고 일 두* 최 중 현** 김 이 두† 이 재 민***
Goh, IL-Du Choi, Joong-Hyun Kim, E-Doo Lee, Jae-Min
(논문접수일 : 2009년 1월 9일 ; 심사종료일 : 2009년 4월 9일)

요 지

3차원 패러메트릭기반 솔리드모델링 방법으로 형상을 표현하고 건설산업분야에서 요구되는 다양한 정보를 처리할 수 있는 BIM기술은 2차원 도면의 자동생성, 건축요소들간의 간섭검토와 물량산출 이외에도 건물내·외 공간분석 및 관리, 구조 해석 및 설계, 에너지 분석, 즉시조달, 시설물 및 재난 관리, 법규검토 등을 위한 고급 엔지니어링업무까지 지원할 수 있게 한다. 본 논문은 이러한 IFC기반 BIM기술을 건설분야의 각 업무에 적용하기 위하여 근원이 되는 각 건물요소의 기하형상을 추출하기 위한 연구내용을 제시하고 구현된 프로그램에 의한 적용사례를 보여준다.

핵심용어 : 건물정보모델링, 국제표준 건물정보모델(IFC), IFC 뷰어, 건물의 형상모델링

Abstract

BIM technologies based on three-dimensional parametric solid modeling can provide building industries with a wide range of information, and then enable not only to automate architectural drawings, detect clashes between building components, and estimate building materials, but also to manage effectively architectural and engineering information about building spaces, structures, energy, just-in-time delivery, facility management, and code checking. This paper presents an implementation to extract geometric data from IFC files, and validates the system with simple and complex buildings.

Keywords : *building information modeling, industry foundation classes, IFC viewer, building geometric modeling*

1. 서 론

건물을 구성하는 부재들을 3차원 패러메트릭기반 솔리드모델링 방법으로 형상을 표현하고 건설산업의 각 분야에서 필요로 하는 다양한 종류의 정보를 처리할 수 있는 BIM 기술은 이제 건축계획 및 설계, 엔지니어링, 시공, 유지관리 등의 실무에 적극 활용되고 있다. 현재 BIM용 프로그램들은 모델링된 3차원 건물정보로부터 2차원 도면을 자동으로 생성할 뿐만 아니라, 건축요소들간의 간섭검토와 물량산출 등의 일차적인 데이터제공에서부터 건물내·외 공간분석 및 관리, 구조해석 및 설계, 에너지 분석, 즉시조달, 시설물 및 재난 관리, 법규검토 등

을 위한 고급 엔지니어링정보까지 제공할 수 있다(김이두 등, 2007). 이러한 BIM 기술의 개발과 적용에 앞서는 미국의 GSA(General Services Administration)는 BIM으로 건물 설계정보를 모델링하기 위한 지침을 제공하고 건축설계의 결과물을 디지털로 납품하도록 하고 있다. 뿐만 아니라 근래에는 법원건물을 설계함에 있어 건물의 임대면적, 건물내의 통행제한에 따른 동선분석, 에너지 효율, 그리고 설계된 건물의 물량산출 등에 BIM기법을 응용하려는 연구를 진행하고 있다(GSA, 2006; Eastman 2007).

이러한 BIM용 프로그램에 의해 모델링된 건물정보는 IAI에 의해 개발된 건설분야의 국제표준정보모델인 IFC 파일형

† 책임저자, 정희원·울산대학교 건축대학 교수
Tel: 010-7553-8932 ; Fax: 052-259-1690
E-mail: edookim@ulsan.ac.kr

* 정희원·서울산업대학교 건축공학과 교수

** 우송대학교 건축학부 교수

*** 조지아텍 건축대학 박사과정

• 이 논문에 대한 토론을 2009년 6월 30일까지 본 학회에 보내주시면 2009년 8월호에 그 결과를 게재하겠습니다.

태로 교환된다. 현재 IFC2x3 버전이 대부분의 BIM용 프로그램들에서 지원되며, IFC2x4가 출시 예정이다(IAI 홈페이지). ArchiCAD, Revit, Bentley Architecture, 그리고 Digital Project 등 건축분야에서 사용되는 BIM용 프로그램들은 관련 당사자들간에 모델링된 건물정보를 교환하기 위하여 IFC 파일형태로 입력과 출력을 할 수 있는 기능을 제공하고 있다. 이러한 IFC 파일에 저장된 데이터들은 국제표준에 따른 건물정보모델이기 때문에 호환성은 물론이고 업무분야의 활용측면에서 안정적이다. 국내의 여러 연구들도 이러한 IFC 모델을 물량산출의 자동화 또는 시설물 유지관리, 그리고 복합 엔지니어링 분야를 위한 정보기저로서 활용하기 위하여 수행된 바 있다(황영삼 2004; 이상현 등 2003; 김인한 등, 2007).

IFC 파일은 IFC EXPRESS 스키마에 따른 엔티티 인스턴스들을 담고있는 텍스트파일로서 수천에서 많게는 수백만 줄에 달하는 광대한 양의 정보를 포함하고 있다. 이러한 IFC 파일의 내용을 다루기 위해서는 IFC 뷰어를 사용하게 되는데 그 종류는 크게 텍스트뷰어와 그래픽뷰어로 분류된다. 현재 IFC에 관한 모든 종류의 정보를 제공하기 위해 운영되는 IfcWiki 사이트는 IfcQuickBrowser(<http://www.team-solutions.de>), IfcObjectCounter(<http://www.fzk.de>), 그리고 IfcEngineBasic(<http://www.ifcbrowser.com>) 등의 텍스트뷰어들을 무료로 배포하고 있다. 이러한 텍스트뷰어들은 IFC 물리적 파일에 저장된 데이터간의 관계를 탐색할 수 있도록 하며 저장된 엔티티들의 속성값이나 인스턴스들의 통계값들을 보여준다. 그러나 이러한 텍스트뷰어는 파일 자체를 직접 살펴볼 수 있지만, 각 엔티티의 내용을 이해하려면 전문가 수준의 IFC와 EXPRESS에 관한 지식이 요구된다. 그러므로 IFC 파일을 다루기 위한 그래픽뷰어 프로그램들인 Solibri Model Viewer(<http://www.solibri.com>), Nemetschek IfcViewer(<http://www.nemetschek.com>), DDSViewer(<http://www.dds-cad.com>), IfcViewer(<http://www.fzk.de>), 그리고 IFC Engine Viewer(<http://www.ifcbrowser.com>) 등이 일반적으로 사용된다. 아래의 그림 1과 그림 2는 본 연구에 사용된 간단한 예제건물의 IFC파일을 Solibri Model Viewer와 Nemetschek IfcViewer에서 가시화한 화면을 보여준다.

이러한 외국의 IFC 뷰어들이 그 다양한 기능으로 국제표준 건물정보모델을 다룰 수 있도록 지원하고 있는 반면에, 국내에는 많은 BIM에 관한 연구들이 진행되었으나 아직 IFC 뷰어를 구현한 사례는 찾아보기 힘들다. 본 논문은 이러한 IFC기반 BIM 기술을 건축분야의 각 업무에 적용하기 위하여 그 기초가 되는 3차원 그래픽기반 IFC 뷰어를 구현

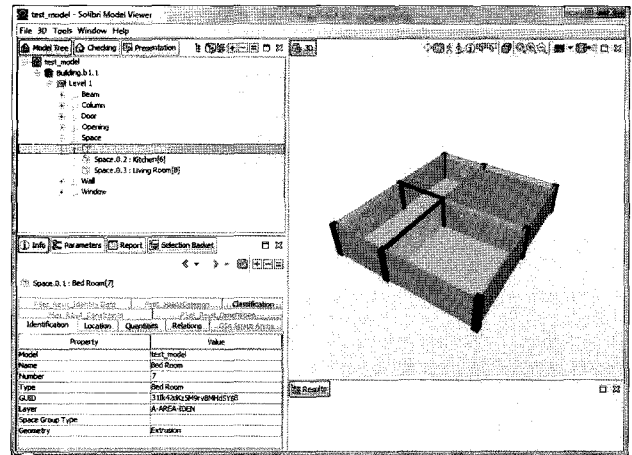


그림 1 Solibri Model Viewer 실행화면

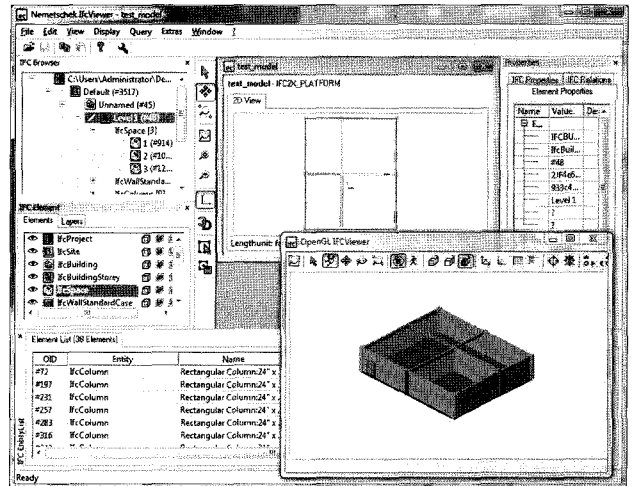


그림 2 Nemetschek IfcViewer 실행화면

하기 위해 수행된 연구내용을 제시하고, 실제 구현된 프로그램에 의한 적용사례를 보여준다.

2. IFC의 건물 구성요소

2.1 물리적 구성요소

IAI의 IFC는 건축물의 구성요소들이나 건설업무 단계들을 표현하는 데이터 요소들을 정의하는 건물정보모델이다. 이 정보모델은 건축물의 프로젝트 당사자들간에 정보를 공유하기 위한 응용프로그램을 구현하는 데 사용된다. IFC 정보모델은 여러 개의 작은 스키마들로 구성되는데, 이들 중에서 IfcKernel 스키마내의 IfcProduct 엔티티를 확장한 IfcShared Bldg Elements 스키마내의 IfcElement의 하위 엔티티인 IfcBuildingElement 엔티티가 건물구조를 이루거나 공간을 분리하는 구성요소들을 묘사한다. 여기서, IfcProduct는 건물내에서의 위치와 자체 형상정보를 가지는 상위 엔티티로서의 역

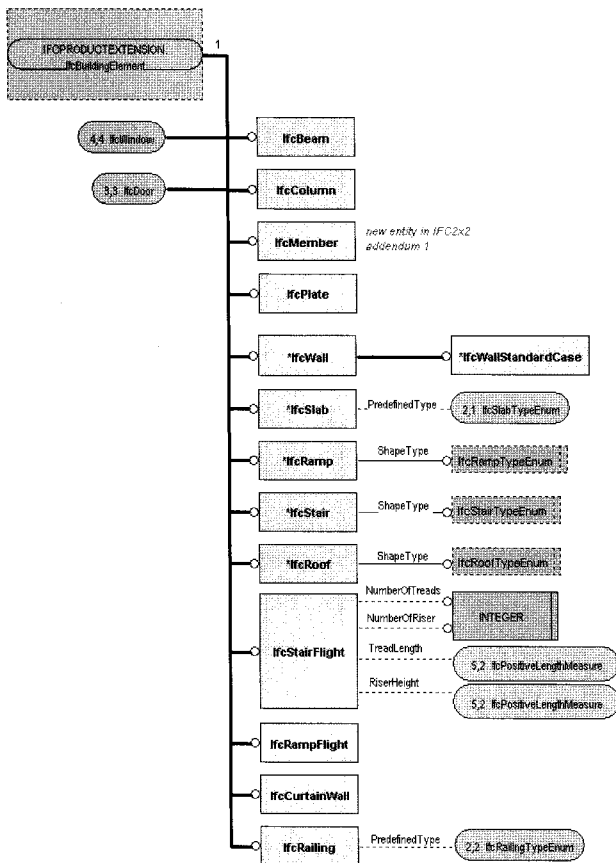


그림 3 IFC의 물리적 건물 구성요소

할을 하며, IfcBuildingElement 엔티티의 하위 엔티티들은 건물을 실제로 구성하는 지붕, 슬라브, 벽체, 보, 기둥, 문과

창문 등을 표현하며 각 요소의 특성을 나타내기 위한 속성들을 갖는다(그림 3 참조). 물리적으로 건물을 구성하는 이러한 IFC의 각 엔티티에 대한 개요는 표 1과 같다.

본 연구에서는 위의 표 1에서 보인 다양한 IfcBuilding Element의 하위 엔티티로 정의된 다양한 건물요소들 중에서 건물의 주요 구조체가 되는 IfcBeam, IfcColumn, IfcWall, 그리고 IfcSlab 등을 대상으로 형상데이터를 추출한다. 그리고 건물내에 경사진 기둥이나 굴곡진 보 등을 나타내기 위해 사용되는 IfcBuildingElementProxy도 대상에 포함한다.

2.2 공간적 구성요소

IfcKernel 스키마내의 IfcProduct를 확장한 IfcShared BldgElements 스키마내의 IfcSpatialStructuralElement는 IfcElement에 속하는 엔티티들을 층단위의 공간적인 위계나 그룹단위의 복합요소를 구성할 목적으로 사용되는 엔티티로, 현재 그 하위요소로는 IfcSite, IfcBuilding, Ifc Building Story, 그리고 IfcSpace 등이 정의되어 있다. 본 연구에서는 개별 개체로 실제 형상을 표현할 수 있는 IfcSpace 엔티티에 대해 형상데이터를 추출한다. 이러한 공간구조를 정의하는 엔티티들간의 관계 또는 위에서 언급한 IfcBuildingElement가 어떤 공간적 구성요소에 포함되는 지 등의 관계는 IfcRel Aggregates 엔티티들에 의해 정의된다.

표 1 IfcBuildingElement의 하위 엔티티들

엔티티 이름	건축적 요소	각 엔티티에 해당하는 건물부재 종류
IfcBeam	보	수평에 가까운 부재-beam, joist, lintel, T_beam 등
IfcColumn	기둥	수직에 가까운 부재-column
IfcMember	선형부재	임의방향의 선부재-brace, chord, collar, member, mullion, plate, post, purlin, rafter, stringer, strut, stud 등
IfcPlate	면부재	임의방향의 면부재-curtain_panel, sheet 등
IfcWall	벽체	수직에 가까운 면요소-standard, polygonal, shear 등
IfcSlab	슬라브	수평에 가까운 면요소-floor, roof, landing, baseslab 등
IfcRamp	경사로	층간 이동을 위한 계단없는 통행로-여러 종류의 경사로 형태가 제공됨
IfcStair	계단	층간이동을 위한 계단-여러 종류의 계단 형태가 제공됨
IfcRoof	지붕을 위한 복합요소	건물을 상부를 이루는 구조체-flat_roof, shed_roof, gable_roof 등 여러 종류의 지붕 형태가 제공됨
IfcStairFlight	계단의 경사부분	계단부분-straight, winder, spiral, curved, freeform 등
IfcRampFlight	경사로의 경사부분	경사로의 구성요소로 사용-straight, spiral 등
IfcCurtainWall	커튼월	건물외부면을 감싸는 하중을 받지 않는 벽
IfcRailing	난간, 통행보호시설	통행이나 안전을 위한 보조시설-handrail, guardrail, balustrade 등
IfcWindow	창문	벽체에 포함되는 창문이나 또는 천창
IfcDoor	문	개구부를 열고 닫는 부재
IfcBuildingElementProxy	범용 건물요소	특히 분류되지 않는 건물요소로 IFC에서 다루지 않는 건물요소에 관한 정보를 교환하기 위하여 사용됨

* 밑줄친 엔티티 이름들은 본 연구에서 다루어질 대상임

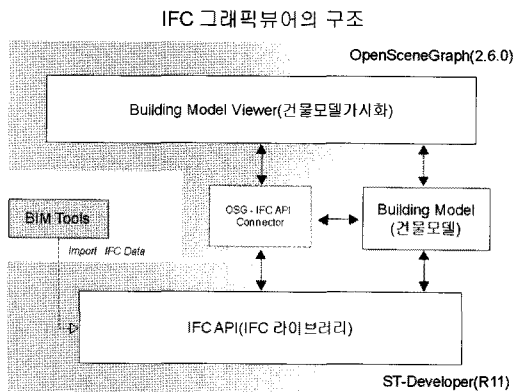


그림 4 IFC 뷰어의 구조

3. IFC 그래픽뷰어의 구현

3.1 시스템 개요

IFC 물리적 파일로부터 건물형상을 추출하여 가시화하기 위하여 본 연구에서 개발한 IFC 그래픽뷰어를 구성하는 주요 모듈들은 IFC 라이브러리, 건물모델, 그리고 건물모델가시화 등이다(그림 4 참조). 본 시스템을 구현하기 위한 프로그래밍환경으로 Visual Studio.Net 2005을 사용하였으며, 프로그래밍언어는 C++이다.

3.1.1 ST-Developer에 의한 IFC 라이브러리(IFC API)

IFC 라이브러리는 IFC 파일에 저장되어 있는 데이터를 읽어들이어 IFC 스키마에서 정의하는 내용에 따라 그 데이터를 해석하고, 각 건물요소의 형상에 관련된 정보를 발췌해 빌딩모델 부분으로 전달할 수 있는 함수들로 구성된다. 이러한 라이브러리를 구축하기 위해서는 IFC 스키마에 정의된 EXPRESS 엔티티들을 C++나 Java 등의 프로그래밍 언어에 따른 클래스로 변환하고 각 클래스의 속성값들을 다룰 수 있는 인터페이스 함수들을 구성할 수 있는 개발환경이 필요하다.

본 연구에서 그림 5와 같이 단계별로 IFC API를 생성하여 사용한다. 우선 그림 5의 1단계에서는 ST-Developer R11의 ST-Lib Wizard를 사용하여 IFC의 EXPRESS 스키마 파일인 IFC2x3_Final.exp로 정적바인딩(early binding)방식¹⁾의 C++로 변환된 IFC 라이브러리 파일을 생성한다. 이 라이

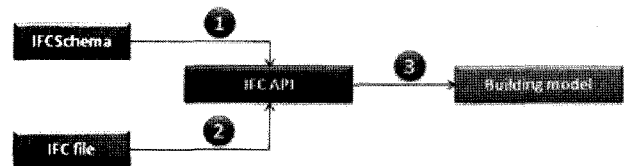


그림 5 IFC 라이브러리의 구성과 활용 단계

브러리는 컴퓨터메모리 상에서 건물정보를 다룰 수 있도록 구성자(constructor), 데이터 필드(field) 및 함수(method) 등을 제공한다(Steptools, 2008). 일단 이러한 IFC API가 생성되고 나면, 2단계처럼 IFC 스키마에 따른 물리적 파일을 읽어들이어 프로그래밍 언어로 메모리내의 정보를 접근할 수 있다. 3단계로 사용자는 IFC API의 함수나 매크로로 읽혀진 정보를 원하는 형태로 가공할 수 있게 된다.

3.1.2 건물요소의 형상표현을 위한 클래스(Building Model)

본 연구에서는 IFC의 주요 건물요소들인 보, 기둥, 벽체, 슬라브, 그리고 공간 등의 형상 및 단면속성 정보를 다루기 위한 단면변수, 좌표계, 3차원 형상 등의 정보를 저장하기 위한 클래스들을 정의한다(자세한 내용은 3.4절 참조). 현재 각 건물요소는 3차원 형상을 표현하는 형상객체(Shape)과 구조해석을 위한 해석모델(Analysis Model)을 주요 속성으로 가진다. 여기서, 형상객체는 건물의 3차원 형상을 가시화하기 위해 사용되었으며, 해석모델은 IFC 요소들로부터 구조해석모델(보와 기둥은 선형요소, 벽체와 슬라브는 면요소로 표현됨)을 생성시키기 위해 사용될 수 있다.

3.1.3 OpenSceneGraph에 의한 건물모델 가시화 (Building Model Viewer)

OpenSceneGraph는 씬그래프에 의한 그래픽요소의 관리와 렌더링 최적화 기능을 기본적으로 제공하는 오픈소스 라이브러리로, ANSI C++로 되어 있어 이식성이 뛰어나며 OpenGL을 기본으로 한 미들웨어이기 때문에 수 많은 그래픽 하드웨어상에서 효율적으로 렌더링할 수 있다. 이러한 OSG는 3차원 그래픽스, GIS, CAD, 모델링과 디지털 콘텐츠 생성, 데이터베이스 개발, 가상현실, 애니메이션, 게임, 그리고 오락 분야 등에서 복잡한 2, 3차원 그래픽을 위해 활용되고 있다.

여기서, 씬그래프는 효율적인 렌더링을 위해 공간데이터를 체계화하기 위한 계층적인 트리형태의 데이터구조이다. 씬그래프 트리는 최상수준이 루트노드로부터 시작되며, 그 루트노드 아래에는 그룹노드들이 기하요소와 그 외양을 제어하기 위한 렌더링상태를 조직화하기 위하여 사용된다. 루트노드와 그

1) IFC파일에 저장된 데이터를 해석하기 위한 기법 중의 하나로 IFC 엔티티와 대응되는 클래스를 미리 생성하여 그 클래스에 해당 인스턴스 정보를 담고, 클래스 멤버함수를 이용하여 그 정보를 처리하는 방식이다.

룹노드들은 임의 개수의 자식노드들을 가질 수 있다. 이러한 그룹노드로는 그 자식노드를 가능/불가능하게 전환하는 스위치노드(Switch Node), 시점으로부터 거리에 따라 자식노드를 선택하는 LOD(Level of Detail)노드, 자식노드 기하의 변환상태를 수정하는 변환노드(Transformation Node) 등이 대표적이다. 씬그래프의 끝에 붙는 말단노드(Geode)들은 씬그래프 내의 객체들을 구성하는 실제 기하요소를 포함한다.

본 연구에서의 건물요소들은 이러한 OSG의 개념에 따라 구조화하여 가지화된다. 그러므로 IFC 파일로부터 입력된 형상데이터들은 말단노드인 Geode에 기하요소로 저장된다. 본 연구에서는 OpenSceneGraph 2.6.0을 사용하였다(참고 문헌 8 참조).

3.2 건물요소의 형상데이터 분석

3.2.1 IfcProduct 엔티티의 형상표현

IFC의 IfcProduct 엔티티는 그 하위 엔티티인 IfcBuildingElement의 위치와 형상을 표현하는 기본적인 데이터구조를 갖고 있다. 즉, 각 요소의 형상을 표현하기 위하여 IfcProduct의 Representation 속성은 IfcProductDefinitionShape 엔티티를 가지는데, 이 엔티티의 Representation 속성은 다시 IfcShapeRepresentation 엔티티들의 리스트를 값으로 가지며, 각 IfcShapeRepresentation의 Items 속성은 최종적으로 IfcGeometricRepresentationItem 인스턴스들을 집합을 가지도록 설계되어 있다(그림 6 참조). 또한, 각 요소의 위치정보를 나타내기 위하여 Ifcproduct의 ObjectPlacement 속성은 해당 요소의 국부좌표계를 정의하는 IfcLocalPlacement 인스턴스를 가진다. 이 IfcLocalPlacement는 그림 6에서 보는 바와 같이 여러 개의 로컬좌표계들로의 변환을 지원하기 위해서 중첩되어 정의될 수 있다. 이 엔티티는 국부좌표계에서 정의되는 요소형상을 전체좌표계로 변환할 수 있는 정보를 제공한다.

3.2.2 건물요소별 형상표현 방법

IFC의 IfcProduct를 위한 3차원 형상표현은 IfcSolidModel, IfcShellBasedSurfaceModel, IfcFaceBasedSur-

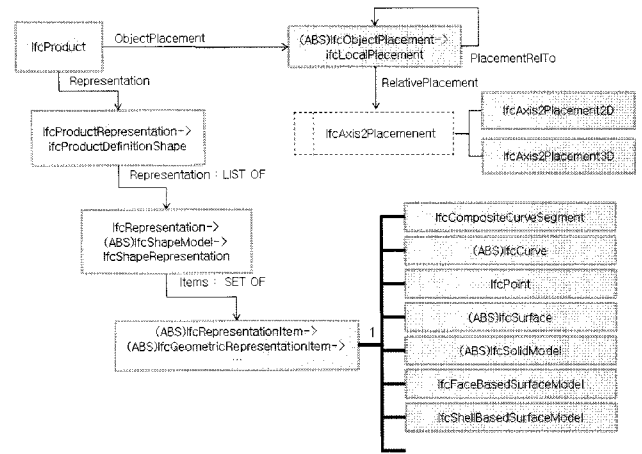


그림 6 IfcProduct 엔티티의 형상표현을 위한 좌표계 및 기하요소

faceModel, IfcCsgPrimitive3D, IfcHalfSpaceSolid, 그리고 불리언연산에 의한 조합형태인 IfcBooleanResult 등 다양한 방식이 가능하다. 건물의 물리적 구성요소의 상위 엔티티인 IfcBuildingElement에 대해 정의된 형상표현방식은 IfcShapeRepresentation 엔티티의 두 개의 속성들, RepresentationIdentifier와 RepresentationType에 의해 정의되며, 이 속성값들에 의해 실제로 사용가능한 기하요소인 IfcGeometricRepresentationItem의 하위 엔티티가 결정된다(표 2 참조).

IFC는 이러한 상위 엔티티 IfcBuildingElement에서 정의한 형상표현방식들 이외에도, 각 건물요소에 대하여 표 3과 같이 추가로 사용가능한 표현방식들을 정의하고 있다.

IFC가 제시한 건물요소의 형상표현방식들이 위에서 언급한 바와 같이 다양함에도 불구하고, 현재 실무에 많이 사용되는 BIM용 프로그램들은 건물의 특성을 고려하여 IfcSolidModel에 의한 방법들 중 IfcSweptAreaSolid의 IfcExtrudedAreaSolid표현, IfcFaceBasedSurfaceModel 표현, IfcBooleanResult표현, 그리고 IfcMappedItem에 의한 매핑표현 등을 많이 사용하고 있다. 본 연구에서도 이러한 형상표현방식들을 우선적으로 구현하고자 한다.

표 2 IfcBuildingElement에 대한 형상표현 방법

RepresentationIdentifier 속성의 값	RepresentationType 속성의 값	IfcGeometricRepresentationItem로 사용가능한 하위 엔티티
'Body'	'BoundingBox'	IfcBoundingBox
'Body'	'SurfaceModel'	IfcShellBasedSurfaceModel 또는 IfcFaceBasedSurfaceModel
'Body'	'Brep'	IfcManifoldSolidBrep
'Body'	'MappedRepresentation'	IfcRepresentationMap의 MappedRepresentation 속성으로 재사용될 형상표현을 나타냄

표 3 IfcProduct 엔티티의 형상표현을 위한 좌표계 및 기하요소

건물요소	IFC 엔티티	RepresentationIdentifier /RepresentationType의 속성값들	IFC의 형상표현방식(하위객체) *밑줄:필수적, 나머지:선택적	비고
보와기둥	IfcBeam IfcColumn	표준방식: 'Body'/'SweptSolid'	솔리드표현방식 : <u>IfcExtrudedAreaSolid</u> 프로파일 : - <u>IfcArbitraryClosedProfileDef</u> (하위객체 : <u>IfcArbitraryProfileDefWithVoids</u>) - <u>IfcParameterizedProfileDef</u> (하위객체 : <u>IfcCircleProfileDef</u> , <u>IfcEllipseProfileDef</u> , <u>IfcRectangleProfileDef</u> , <u>IfcCShapeProfileDef</u> , <u>IfcIShapeProfileDef</u> , <u>IfcLShapeProfileDef</u> , <u>IfcTShapeProfileDef</u> , <u>IfcUShapeProfileDef</u> , <u>IfcZShapeProfileDef</u>)	-국부좌표계상에서 임의 방향으로 전개됨
		고급방식: 'Body'/'Clipping'	솔리드표현방식 : 'SweptSolid'와 동일 불리언연산 : IfcExtrudedSolid와 IfcHalfSpaceSolid(s)간의 <u>IfcBooleanResult</u>	
		재사용방법: 'Body'/'MappedRepresentation'	위의 'SweptSolid'와 'Clipping'방식을 재사용함	
슬라브	IfcSlab	표준방식: 'Body'/'SweptSolid'	솔리드표현방식 : <u>IfcExtrudedAreaSolid</u> 프로파일 : - <u>IfcArbitraryClosedProfileDef</u> - <u>IfcParameterizedProfileDef</u> (<u>IfcCircleProfileDef</u> , <u>IfcEllipseProfileDef</u> , <u>IfcRectangleProfileDef</u>)	-국부좌표계상에서 임의 방향으로 전개됨
		고급방식: 'Body'/'Clipping'	솔리드표현방식 : 'SweptSolid'와 동일 불리언연산 : IfcExtrudedSolid와 IfcHalfSpaceSolid(s)간의 <u>IfcBooleanResult</u>	
		재사용방법: 'Body'/'MappedRepresentation'	위의 'SweptSolid'와 'Clipping'방식을 재사용함	
벽체	IfcWall	표준방식: 'Body'/'SweptSolid'	솔리드표현방식 : <u>IfcExtrudedAreaSolid</u> 프로파일 : - <u>IfcArbitraryClosedProfileDef</u>	-사각형 단면형태가 아니거나 벽체의 길이방향으로 두께가 달라지는 경우 -국부좌표계 또는 전체 좌표계의 z방향으로 프로파일에 수직으로만 전개됨
		고급방식: 'Body'/'Clipping'	솔리드표현방식 : 'SweptSolid'와 동일 불리언연산 : IfcExtrudedSolid와 IfcHalfSpaceSolid(s)간의 <u>IfcBooleanResult</u>	
	IfcWallStandardCase	표준방식: 'Axis'/'Curve2D'	벽체중심선 표현방식 : IfcPolyline 또는 IfcTrimmedCurve (두께와 offset은 IfcMaterialLayerSet로 제공됨)	-벽체 전체가 동일한 두께를 가지며, SweptSolid로만 형상이 표현됨 -국부좌표계 또는 전체 좌표계의 z방향으로 프로파일에 수직으로만 전개됨
		표준방식: 'Body'/'SweptSolid'	솔리드표현방식 : <u>IfcExtrudedAreaSolid</u> 프로파일 : - <u>IfcArbitraryClosedProfileDef</u> (하위객체 포함) - <u>IfcRectangleProfileDef</u>	
		고급방식: 'Body'/'Clipping'	솔리드표현방식 : 'SweptSolid'와 동일 불리언연산 : IfcExtrudedSolid와 IfcHalfSpaceSolid(s)간의 <u>IfcBooleanResult</u>	

3.3 요소형상의 전체좌표계로의 변환

IfcProduct의 전체좌표계에서의 형상데이터는 Representation 속성에 의해 국부좌표계에서 정의된 3차원 형상데이터를 ObjectPlacement 속성에 의해 정의되는 변환좌표계를 적용함으로써 얻어지게 된다. 이 절에서는 IfcRectangleProfileDef을 전개단면으로 가지는 IfcExtrudedAreaSolid인 형상표

현에 대해 이 과정을 자세히 설명한다.

3.3.1 국부좌표계에서의 형상데이터의 구성

IfcExtrudedAreaSolid에 의한 형상표현에 관한 정보는 전개시킬 단면형상을 나타내는 SweptArea 속성, 단면형상의 국부좌표계에서의 위치를 나타내는 Position 속성, 전개될 방향을 나타내는 ExtrudedDirection 속성, 그리고 전개

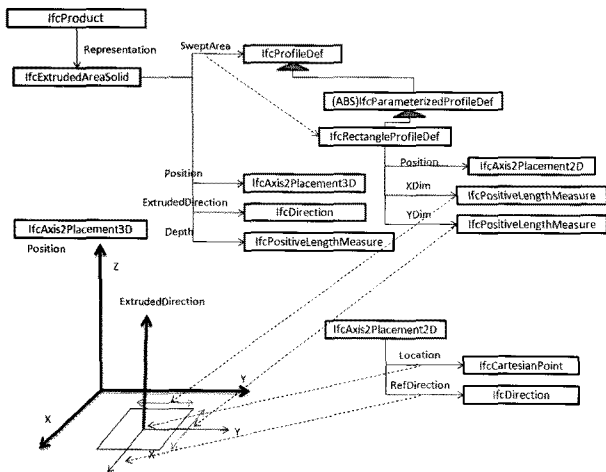


그림 7 IfcExtrudedAreaSolid의 IfcRectangleProfileDef에 의한 국부좌표계에서의 형상 표현

깊이를 위한 Depth 속성등으로 표현된다. 여기서 단면형상은 다시 그 위치를 나타내는 Position 속성, 두 방향 치수를 위한 XDim과 YDim 속성으로 정의된다(그림 7 참조). 그러므로, 현재의 단면형상의 프로필이 사각형단면이므로 XDim과 YDim, ExtrudedDirection과 Depth로부터 육면체를 이루는 면들을 주어진 국부좌표계에서 구성할 수 있다.

이러한 IfcProfileDef에서 정의한 형상들의 좌표값들을 P_D 라 하고, IfcProfileDef 내의 IfcAxis2Placement2D를 나타내는 행렬을 M_1 , IfcExtrudedAreaSolid 내의 IfcAxis2Placement3D를 나타내는 행렬을 M_2 라 하면, 최종적으로 각 요소의 국부좌표계에서의 단면형상들의 좌표값들(P_L) 위치는 아래와 같이 계산된다.

$$P_L = M_2 * M_1 * P_D \quad (1)$$

3.3.2 전체좌표계의 형상데이터의 변환

IfcProduct의 ObjectPlacement속성에 의한 국부좌표계의 표현은 최종적으로 전체좌표계에 도달할 때까지 IfcLocalPlacement가 중첩되는 형태로 정의된다. 이러한 변환좌표계를 차례로 M^1, M^2, \dots, M^n 이라고 정의하면 전체좌표계에서 요소형상의 좌표값들 P_W 는 식 (2)와 같이 정의된다.

$$P_W = M^n * \dots * M^2 * M^1 * P_L \quad (2)$$

3.4 주요 클래스들과 상호관계

그림 8은 IFC 파일로 읽어들이 추출된 형상데이터를 저장하기 위한 건물모델 모듈을 위해 정의된 C++ 클래스들과 그 상호관계를 보여준다. 클래스들은 IFC모델 해석부분(Model

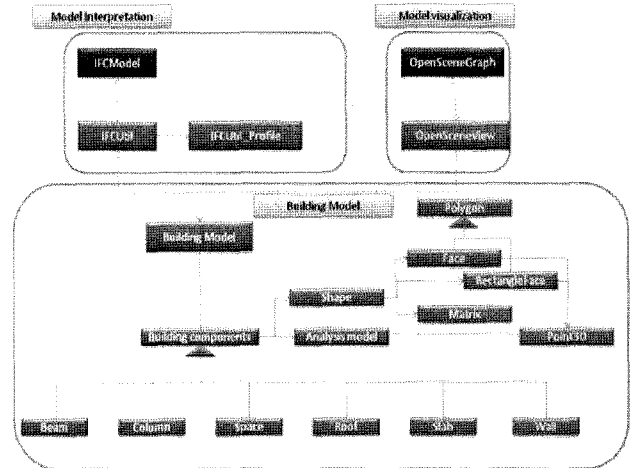


그림 8 Building Model의 주요 클래스 구조

Interpretation), 형상데이터 저장부분(Building Model), 그리고 OSG에 의한 가시화 부분(Model Visualization) 등, 3개 부분들로 나뉘어진다(그림 8 참조).

- IFC 모델해석부분은 IFC 파일에 저장된 인스턴스들로부터 형상에 관한 데이터를 추출하는 기능을 수행한다. IfcUtil 클래스는 연구에서 다루는 각 건물요소별로 국부좌표계와 형상데이터를 읽어 전체좌표계로 전환하기 위한 여러 함수들을 가지며, IfcUtil_Profile 클래스는 IfcArbitraryClosedProfileDef, 파라메트릭 단면을 표현하는 IfcRectangleProfileDef와 IfcIShapeProfileDef 등의 국부좌표계에서 정의되는 3차원 형상데이터를 생성하는 역할을 수행한다.
- 빌딩모델(Building Model) 부분은 IFC 파일로부터 추출된 형상정보를 IFC 뷰어를 위해 저장하기 위한 클래스들로 구성되며, IFC 건물요소에 대응하는 클래스 이름들을 가진다. 현재, 이러한 클래스들로는 보, 기둥, 공간, 지붕, 슬라브 그리고 벽체 등이 정의되어 있다. 이들 클래스들은 상위클래스인 건물기본요소 클래스(Building components)에서 속성들로 정의된 형상클래스(Shape)와 해석모델 클래스(Analysis Model)를 모두 가지게 된다. 여기서, 형상 클래스는 건물요소의 형상을 표현하기 위한 3차원 다각형들과 전체좌표계로의 변환을 위한 변환행렬을 속성으로 가진다. 해석모델 클래스는 각 요소의 중심선을 나타내기 위한 선 또는 면을 표현하기 위한 점들의 리스트를 가진다.
- 가시화부분은 OpenSceneGraph와 빌딩모델 사이에 정보를 전달하기 위한 클래스들이다. 이 클래스들은 빌딩 모델에 있는 데이터를 OSG내의 기하요소로 전환시키는 역할을 한다.

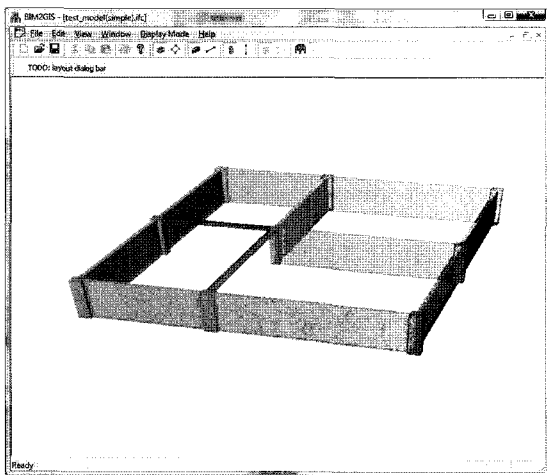
4. IFC 뷰어의 실행 사례

건물의 구성요소들을 효과적으로 가시화하기 위하여 본 프로그램은 각 부채단위로 3차원 형상표현과 1, 2차원의 중심선 표현을 각각 제공한다. 그리고, 건물요소들의 중첩에 의해 내부상태를 보다 효과적으로 파악하기 위하여 투명도를 부여하였다.

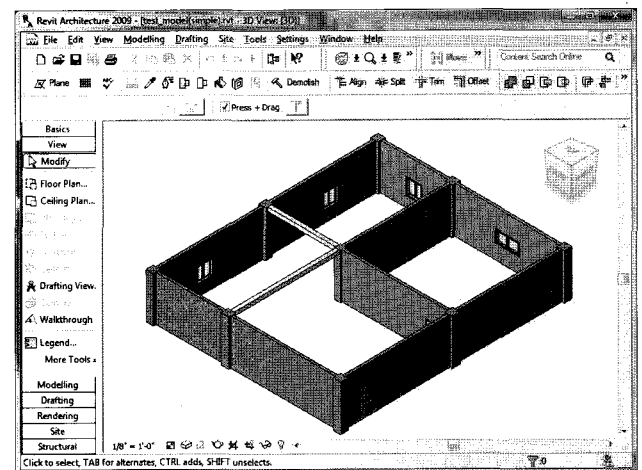
4.1 실행 사례 1 - 간단한 건물

그림 9는 벽체, 보, 기둥, 공간, 그리고 문과 창문 등의 건물요소들로 구성된 간단한 예제용 건물을 AutoDesk Revit에서 모델링 것을 IFC 파일로 저장한 뒤에 본 프로그램으로 읽어들이어 가시화한 것이다.

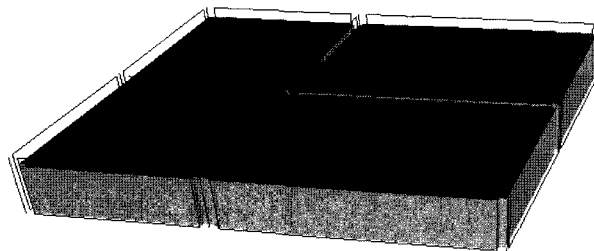
그림 9(a)와 (b)는 본 논문의 그림 1과 그림 2와 같이 IFC 파일을 읽어들이어 본 프로그램과 Revit에서 가시화한 화면을 각각 보여준다. 그림 9(c)는 공간요소(IfcSpace) 엔티티를 3차원으로 보여주고, 기타의 건물요소들은 중심선을 보여준다. 본 사례와 같이 본 프로그램은 아직 구현되지 않은 문과 창문을 제외한 벽체, 보, 그리고 기둥 등의 형상모델을 제대로 보여주고 있다.



(a) 본 연구의 IFC 뷰어의 실행 화면



(b) Revit의 실행화면



(c) 공간요소(IfcSpace)의 가시화 화면(다른 건물요소는 중심선으로 표시)

그림 9 Building Model의 클래스 구조

4.2 실행 사례 2 - 실제 건물

그림 10은 Revit으로 모델링된 건물을 IFC 파일로 내보낸 다음, ArchiCAD로 읽어들이는 전체 건물의 모습을 보여준다.

그림 11은 동일한 IFC 파일을 본 프로그램으로 가시화한 화면들을 보여준다. 그림 11(a)는 그림 10과 비슷한 위치에서의 건물모습이며, 그림 11(b)는 벽체와 슬라브를 감춘 결

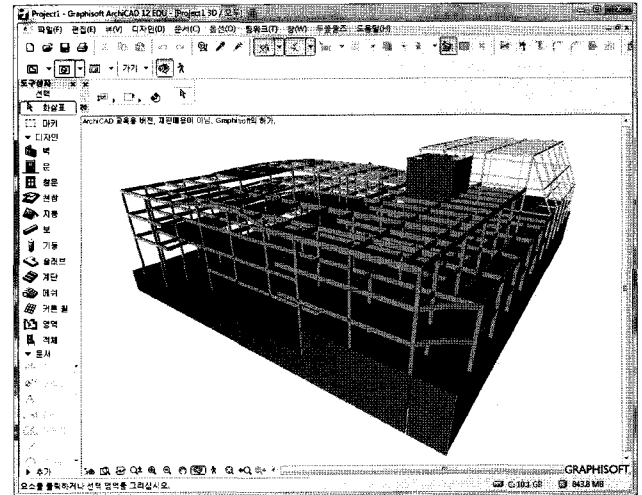
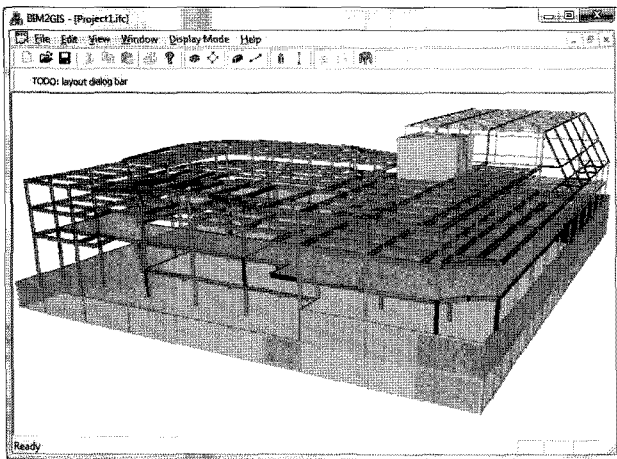
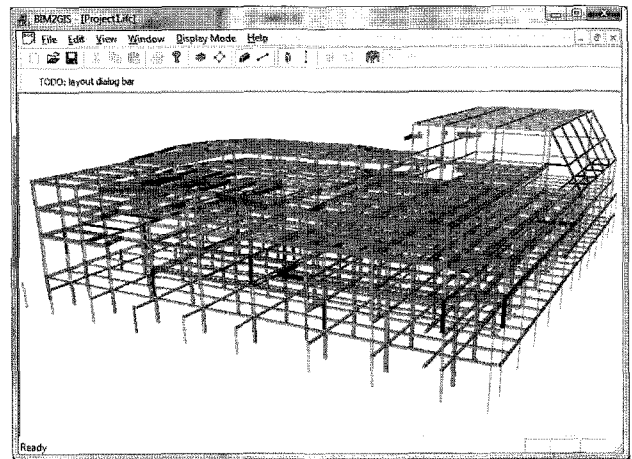


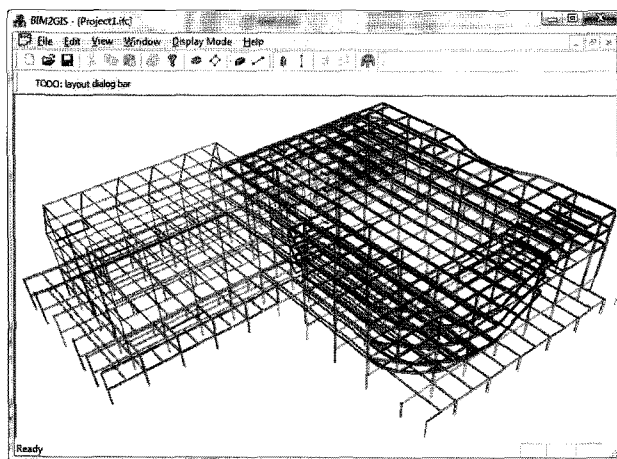
그림 10 ArchiCAD로 본 건물모양



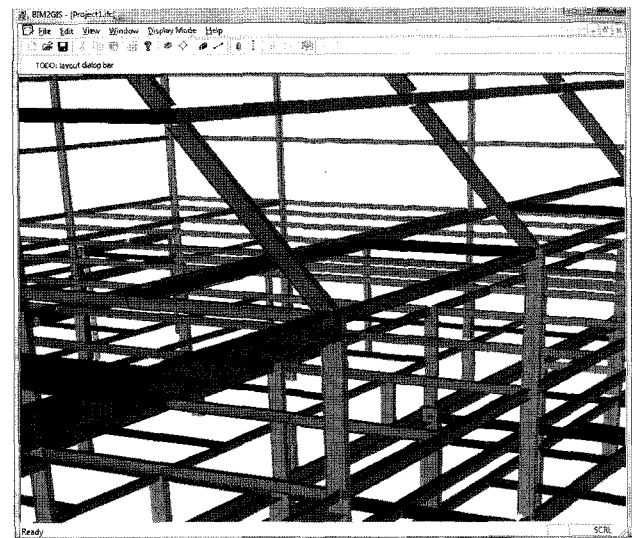
(a) 본 프로그램에서의 전체 건물



(b) 벽체와 슬라브를 감춘 건물모델



(c) 다른 각도에서의 건물골조



(d) 건물골조의 확대화면

그림 11 본 프로그램의 실행화면

파이다. 그림 11(c)는 다른 위치에서의 모습이며, (d)는 건물골조를 확대한 모습이다.

5. 결 론

IFC는 관련 기술자들간에 건물모델링 정보를 교환하기 위하여 국제표준화된 데이터모델로, 대부분의 BIM용 프로그램들은 이러한 IFC 스키마에 따른 물리적 파일의 입출력을 지원하고 있다. 본 연구에서는 이러한 BIM 기술을 적용하여 건축분야에서의 생산성향상을 시도하고자 IFC 그래픽뷰어를 구축하기 위한 세부사항을 설명하고, 구현된 시스템의 실행 결과를 제시하였다.

그러나 BIM용 프로그램들에 의한 IFC 파일의 입출력은 현실적으로 아직 많은 발전을 기대해야 할 단계이다. 즉, BIM용 프로그램으로 수행할 수 있는 업무에 비해 IFC 스키

마에 표현될 수 있는 정보는 부족한 형편이다. 그리고, BIM용 프로그램들이 IFC 파일형태로 데이터를 내보내거나 읽어들이는 기능이 완전한 수준이 아니다. 즉, 해당 프로그램이 내보낸 IFC 파일도 다시 읽어들이면 원래대로 모든 정보가 복구되지 않는다. 그리고 IFC 스키마에 따른다 하더라도 각 요소의 형상모델의 표현이 해당 BIM용 프로그램에 종속적이기 때문에 아직은 일관성이 부족하다. 이러한 IFC에 의한 정보교환의 질적문제나 안정성은 앞으로 각 분야별 활용을 통하여 개선되어야 할 것으로 보인다.

추후 연구에서는 이번 연구에서 제외된 문과 창문, 계단, 경사로, 커튼월 등의 2차적인 건물요소들의 형상표현을 가능하도록 할 뿐만 아니라 솔리드모델링 커널을 사용하여 다양한 IFC 형상표현을 지원할 수 있도록 프로그램의 성능을 확장함으로써, 건물공간 분석, 구조해석 및 설계, 에너지해석, 시설물유지관리, CityGML모델 생성 등에 실제 적용하여 그

활용가능성을 모색할 계획이다.

감사의 글

본 연구는 국토해양부 첨단도시기술개발사업 - 지능형국토 정보기술혁신 사업과제의 연구비지원(08국토정보C04)에 의해 수행되었습니다.

참 고 문 헌

- 김이두, 정연석 (2007) 건물정보모델링의 실무적용 사례, 전산구조공학, 20(1), pp.29~36.
- 김인한, 변수진 (2007) 복합엔지니어링분야의 프로젝트모델 기반 정보공유 기술 개발, 대한건축학회, 대한건축학회 논문집-구조계, 23(10), pp.123~130.
- 이상현, 김인한 (2003) 건설산업 분야의 IFC 모델을 활용한 시

설물 유지관리 적용방안에 관한 연구, 대한건축학회, 대한건축학회 학술발표대회 논문집-계획계, 23(1), pp.783~786.

황영삼 (2004) IFC 모델을 통한 도면으로부터의 물량산출 자동화 연구, 대한건축학회, 대한건축학회 논문집-계획계, 20(12), pp.89~97.

Eastman, C. M., Teicholz, P., Sacks, R., Liston, K. (2007) BIM Handbook, Booksurge, Charlestown, GSA(U.S. General Services Administration), Building Information Modeling Guide Series, <http://www.gsa.gov/bim>, <http://www.ifcwiki.org> <http://www.openscenegraph.org>.

IAI(International Alliance for Interoperability) IFC/ifcXML Specifications(IFC2x3), <http://www.iai-international.org>.

Steptools, Inc. (2008) ST-Developer R11, <http://www.steptools.com>.