

# IFC 기반의 건축구조물 자료교환을 위한 연계모듈 개발 방안

정종현<sup>1\*</sup>

<sup>1</sup>경남대학교 건축학부

## Development Method on Interface Module for IFC-Based Data Exchange of Building Structures

Jong-Hyun Jung<sup>1\*</sup>

<sup>1</sup>Division of Architecture, Kyungnam University

**요약** 본 연구에서는 IFC 기반으로 여러 응용소프트웨어들 사이의 건축구조물 자료교환을 지원할 수 있는 IFC 연계모듈을 보다 적은 시간과 노력으로 개발할 수 있는 방안을 제시하였다. 중요한 점은 연계모듈을 개발하는 과정에서 1) 건축구조물에 관련되는 지식과 경험에 부합되는 물리적, 개념적 대상을 활용할 수 있어야 하며, 2) 복잡한 IFC에 대한 구체적인 지식이 필요하지 않아야 한다는 점이다. 본 연구에서 제시한 방안은 1) 외부의 응용소프트웨어와 직접 연결되는 공용부분과 이를 지원하는 전용부분으로 구분하고, 2) IFC에서 건축구조물과 직접 관련되는 부분을 건축구조물 관련 지식과 경험에 잘 부합되는 공용부분으로 재구성하고, 3) IFC의 나머지 부분은 전용부분으로 재구성하는 것이다. 이러한 방안의 적용사례로 IFC 파일에 있는 자료들을 읽어 시각화, 부재 목록 작성, 물량 산출, Tekla와 연계 등을 수행하는 IFC 연계모듈을 개발하여 본 연구에서 제시한 방안의 타당성을 검토하였다.

**Abstract** The purpose of this study is to propose the method to develop IFC interface module with less time and effort, which can support the IFC-based data exchange of building structures. The core of the IFC interface module is that 1) the physical and conceptual objects well suited to the knowledge and experience of the building structures are exploited and 2) no in-depth knowledge about IFC is required. The proposed method is to 1) divide the IFC interface module into public part open to the external applications and the supporting private part, 2) reorganize the IFC entities related to the building structures into the public part, and 3) reorganize the other IFC entities into the private part. In this study, the IFC interface module was developed and then verified by the application of the module to visualization, member listing, quantity take-off and interface to Tekla of an IFC file.

**Key Words** : Data Exchange, IFC, Interface Module

### 1. 서론

건축구조 분야에서는 건축구조물을 대상으로 구조설계, 시각화, 구조물량 산출, 구조도면 작성 등 많은 업무들을 수행하며, 이 과정에서 다양한 응용소프트웨어(application software)를 활용한다. 그리고 업무의 생산성을 높이고 결과물의 품질을 높이기 위해서 이 응용소프트웨어들은 상호 연계, 즉 서로 필요한 자료들을 적절

하게 교환할 수 있어야 한다. buildingSMART international[1]에서 건축분야의 여러 응용소프트웨어들 사이의 자료교환을 목적으로 개발한 IFC(Industry Foundation Classes)[2]는 건축구조물 대상의 여러 응용소프트웨어들 사이의 자료교환에 활용할 수 있는 표준 자료모델이다.

하지만, 건축, 구조, 설비, 시공 등의 모든 분야와 기획, 설계, 시공, 유지관리에 걸친 모든 생애주기를 대상

\*Corresponding Author : Jong-Hyun Jung(Kyungnam Univ.)

Tel: +82-55-249-2681 email: [ironbell@kyungnam.ac.kr](mailto:ironbell@kyungnam.ac.kr)

Received October 1, 2013

Revised (1st March 3, 2014, 2nd March 18, 2014)

Accepted April 10, 2014

으로 하는 IFC에는 매우 많은 물리적, 개념적 대상이 엔티티(entity)로 정의되어 있는데, 이 중에서 많은 엔티티들이 건축구조물 대상의 전문지식이나 실무경험과는 직접 관계가 없거나 상이한 방식으로 정의되어 있다. 이러한 점 때문에 건축구조물 대상의 IFC 연계모듈 개발에는 IFC에 대한 구체적인 지식이 필요하고 많은 시간과 노력이 소요된다.

IFC를 이용하여 자료를 교환하기 위한 연계모듈의 개발에 소요되는 시간과 노력을 줄이려는 연구가 수행되었고 여러 소프트웨어 라이브러리가 개발되었다[3-9]. 하지만 이들은 IFC 자료들을 엔티티 단위로 읽거나 쓰는 과정을 단순화하는 데에 중점을 두고 있다. 따라서 여전히 IFC에 정의된 엔티티에 대한 구체적인 지식이 필요하며 건축구조물 대상의 전문지식이나 실무경험에 부합되는 방식을 적극적으로 활용하지는 않는다.

이에 본 연구에서는 IFC 연계모듈 개발의 어려운 점을 구체적으로 분석하고, 이를 바탕으로 IFC에 대한 개략적인 지식, 적은 시간과 노력으로 건축구조물 대상의 IFC 연계모듈을 개발할 수 있는 방안을 제시하였다. 그리고 적용사례로 건축구조물에 대한 자료를 교환할 수 있는 IFC 연계모듈을 개발하여 본 연구에서 제시한 방안의 타당성을 검토하였다.

## 2. IFC 연계모듈 개발의 어려움

### 2.1 IFC의 전체적 구성

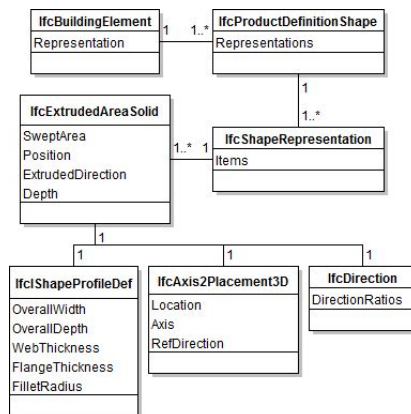
IFC의 자료구조는 IFC 스키마(IFC Schema)에 STEP[10]의 EXPRESS[11]를 이용하여 정의되며, 실제 자료들은 거의 대부분의 경우 IFC 파일(IFC file)에 저장된다. IFC에서는 건축의 모든 분야와 전체 생애주기에 걸친 방대한 자료들을 물리적 대상물과 개념 등으로 체계화하여 엔티티와 그에 포함되는 속성(attribute)으로 정의하고, 이를 Resource layer, Core layer, Domain layer로 계층화하여 구분한다. 가장 하위의 Resource layer에는 시간, 기하학적 형상, 재료, 단면, 물량, 단가 등에 대한 기본적인 엔티티들이, 중간의 Core layer에는 건축의 각 분야별로 공유할 수 있는 엔티티들이, 가장 상위의 Domain layer에는 건축, 구조, 설비, 유지관리 등 건축 각 분야에서 필요로 하는 엔티티들이 포함된다.

### 2.2 건축구조물 관점에서의 IFC

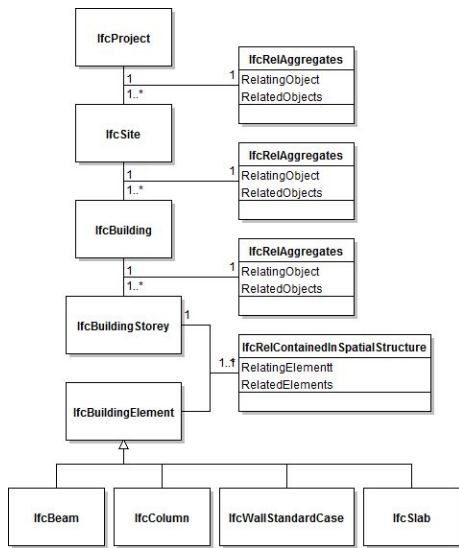
앞에서 기술한 바와 같이 IFC는 건축 분야의 많은 자료들을 정의한 것이므로 엔티티가 매우 많고 그 관계도 복잡하다. 따라서 건축구조물에 초점을 맞춘 관점에서 볼 때 IFC를 파악하고 자료교환을 위한 연계모듈을 개발하는 데에는 다음과 같은 어려움이 있다.

첫째 건축구조물을 대상으로 한정하면 교환할 필요가 없는 엔티티들이 있고, 필요한 엔티티에도 필요없는 속성들이 있다. 그리고 필요없는 엔티티나 속성들을 명확하게 분리하기 곤란한 경우가 많다. 예를 들어 IfcRoot 엔티티에는 IfcOwnerHistory 유형의 OwnerHistory 속성이 있다. 이 속성은 건축구조물의 관점에서는 활용도가 매우 낮아 필요없지만 필수적인 속성이기 때문에 상당히 복잡한 과정을 거쳐서 IfcOwnerHistory 인스턴스를 만들어 OwnerHistory 속성과 연결시켜야 한다.

둘째, 부재(member)의 기하학적 형상을 표현하는 자료구조가 복잡하여 부재의 단면에 대한 자료를 얻으려면 여러 엔티티와 속성에 대한 접근이 필요하다. Fig. 1은 IFC에서 부재의 기하학적 형상의 자료구조 일부분을 나타낸 UML 클래스 다이어그램이다. 여기에서 부재는 IfcBuildingElement로, 부재 단면의 형상은 IfcIShapeProfileDef로 표현되는데 부재의 단면 형상에 접근하기 위해서는 총 4개의 속성과 4개의 엔티티를 거쳐야만 한다.



[Fig. 1] UML class diagram for geometry of member



[Fig. 2] UML class diagram for project, site, ..., member

셋째, 엔터티들의 관계를 정의할 때 관계를 표현하는 엔터티를 이용하므로 어떤 엔터티와 연결된 엔터티의 자료를 얻는 과정이 복잡하다. Fig. 2는 프로젝트, 대지, 건물, 층, 부재의 포함관계의 자료구조를 나타내는 UML 클래스 다이어그램이다. 프로젝트부터 층까지 포함관계는 IfcRelAggregates를 통해서, 층과 부재 사이의 포함관계는 IfcRelContainedInSpatialStructures를 통해서 표현되는데, 이는 복잡하고 일관성이 없으며 다소 특이한 방식이다.

넷째, IFC 파일에 저장된 실제 자료들에 접근하는 표준적인 방법인 SDAI[12]를 이용하는 경우, 엔터티 단위가 아니라 엔터티에 포함되어 있는 속성 단위로 자료들에 접근하기 때문에 엔터티에 포함되어 있는 다수의 자료들에 일일이 접근해야 한다. 예를 들어 Fig. 1의 IfcShapeProfileDef를 얻기 위해서는 먼저

IfcShapeProfileDef에 접근한 후 그에 포함되는 속성인 OverallWidth, OverallDepth, FlangeThickness, WebThickness, FilletRadius를 하나씩 차례로 접근해야만 한다.

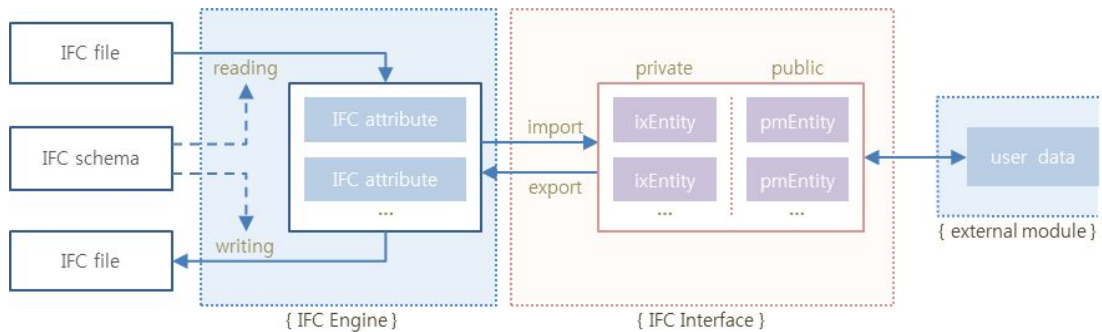
### 3. IFC 연계모듈 개발 방안

#### 3.1 연계모듈의 구조

Fig. 3은 본 연구에서 제안하는 IFC 연계모듈의 구조를 나타낸 것이다. 크게 IFC Engine 부분과 'IFC Interface' 부분으로 구분된다.

IFC Engine[7]은 기존의 소프트웨어 라이브러리(dll)로서, 본 연구의 IFC 연계모듈에서 자료들이 저장되어 있는 IFC 파일을 읽거나 생성 및 저장하는 기능을 수행하는 데에 활용하였다. SDAI를 이용하기 때문에 엔터티가 아니라 속성 단위로 자료에 접근한다. 따라서 reading을 수행하면 IFC 파일에 저장되어 있는 자료들을 읽어서 IFC Engine에 할당된 메모리(memory)에 속성 단위의 IFC attribute로 쓴다. 반대로 writing을 수행하면 IFC Engine에 할당된 메모리에 써어 있는 속성 단위의 IFC attribute들을 읽어 IFC 파일로 옮겨 저장한다.

IFC Interface는 건축구조물 대상으로 활용하는 전문 지식과 실무경험에 따른 물리적 대상물, 개념, 용어, 조건 등을 표현하는 pmEntity와 기하학적 형상, 특성(property), 유틸리티(utility), 소프트웨어개발 방법상 필요한 것 등을 표현하는 ixEntity로 구성된다. pmEntity와 ixEntity 모두 일부는 IFC 엔터티를 그대로 가져오고 일부는 여러 IFC 엔터티들의 속성들을 취합하여 재구성한다. pmEntity는 주로 건축구조물 대상의 상업용, 실무용 응용소프트웨어와 직접 연계되는 공용부분이 되며,



[Fig. 3] Structure of IFC interface module

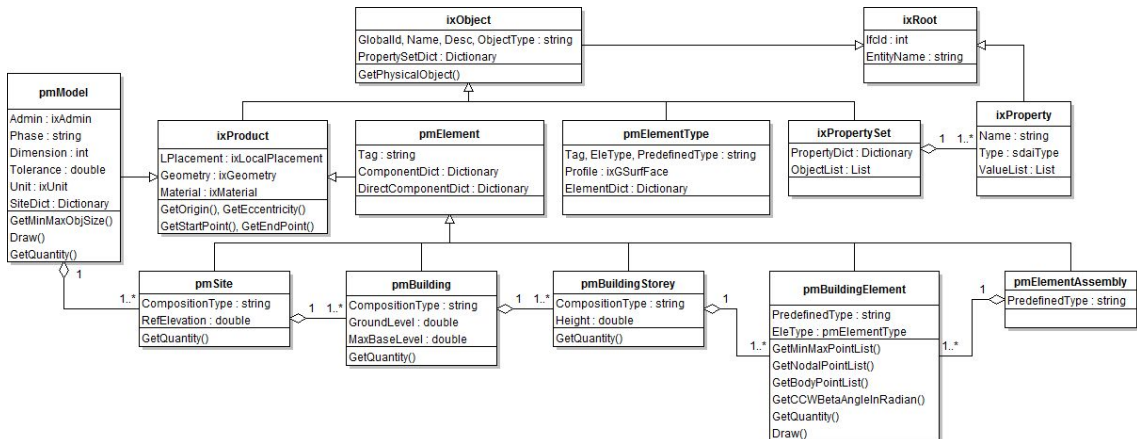
ixEntity들은 이를 뒤에서 지원하고 외부에 노출되지 않는 전용부분이 된다.

ixEntity와 pmEntity는 속성 단위가 아니라 엔티티 단위이며, 이에 따라서 import를 수행하면 IFC Engine의 메모리에 써어 있는 속성 단위의 IFC attribute들을 조합하여 IFC Interface에 할당된 메모리에 엔티티 단위의 ixEntity와 pmEntity로 쓴다. 이와 반대로 export를 수행하면 IFC Interface의 메모리에 써어 있는 ixEntity와 pmEntity에 포함되어 있는 각 속성들을 IFC Engine에 할당된 메모리에 속성 단위의 IFC attribute로 옮겨 쓴다.

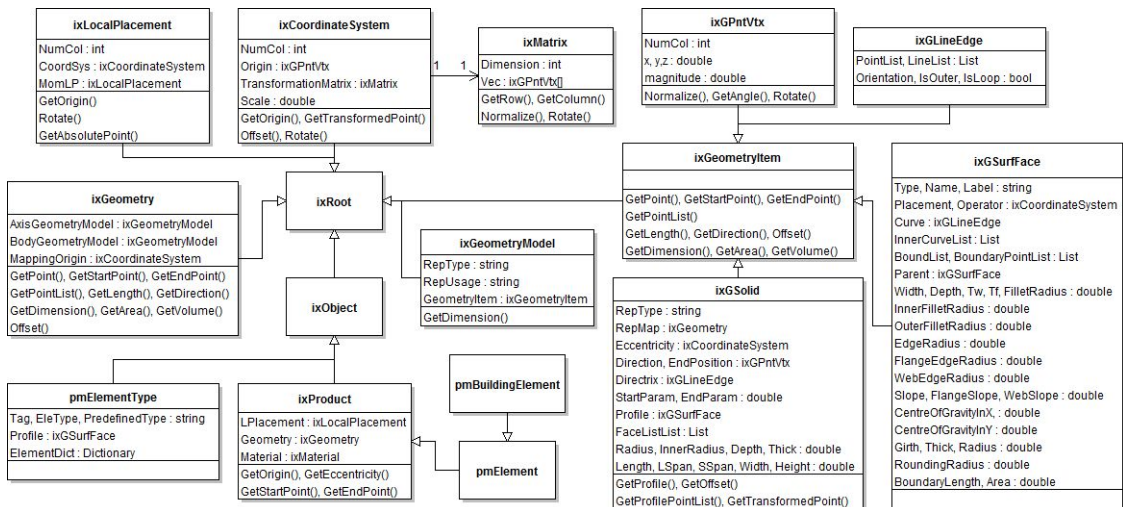
### 3.2 ixEntity와 pmEntity 상세

여기에서는 IFC Interface에서 주요 물리적 대상물

과 기하학적 형상에 대해서만 구체적으로 기술한다. Fig. 4는 주요 물리적 대상물 부분의 UML 클래스 다이어그램인데, 프로젝트(pmModel)-대지(pmSite)-건물(pmBuilding)-층(pmBuildingStorey)-부재(pmBuildingElement)와 물리적 포함관계가 직접적으로 표현되어 있다. Fig. 2와 비교해 보면 포함관계를 표현하는 엔티티들이 제외하고 자료들의 집합체(collection)를 관리하는 방법인 dictionary를 이용하였다. 이러한 자료구조는 보다 직관적이고 단순하며 건축구조물 대상의 전문지식과 실무경험도 잘 부합된다. Fig. 4에서 ixEntity들은 소프트웨어 개발 방법론의 하나인 객체지향설계법의 일반화/특수화 관계를 표현하기 위한 것으로 IFC Interface의 외부에서 직접 활용



[Fig. 4] UML class diagram for main part of ixEntities and pmEntities



[Fig. 5] UML class diagram for geometry part of ixEntities and pmEntities

할 필요가 없다.

Fig. 5는 부재의 기하학적 형상 부분의 UML 클래스 다이어그램이다. 여기에서 보면 부재들을 표현하는 pmBuildingElement가 ixProduct를 통해서 부재의 상대적 위치를 표현하는 ixLocalPlacement와 기하학적 형상을 표현하는 ixGeometry를 직접 갖는다. 하지만 부재의 단면에 대한 자료를 얻기 위해서는 3개의 단계, 즉 ixGeometry-ixGeometryModel-ixGeometryItem을 거쳐야 하는데, Fig. 1과 비교해 보면 1단계가 감소되었으며, ixGeometry에 GetPoint(), GetStartPoint(), ..., GetVolume() 등의 함수가 있어 부재의 위치와 단면의 치수 등에 바로 접근할 수 있다. 이렇게 되면 ixGeometry 뒤의 엔터티들에 대해서 파악할 필요가 없다. 따라서 이러한 자료구조 역시 보다 직관적이고 단순하며 건축구조물 대상의 전문지식이나 실무경험과 잘 부합된다.

## 4. 사례 적용

### 4.1 구현

본 연구에서는 3장에서 기술한 방안에 따라 IFC 연계 모듈을 개발하였으며, 그 타당성을 확인하기 위해서 IFC 파일에 저장된 여러 자료들을 추출하는 응용소프트웨어인 IFC Manager를 개발하였다. IFC 연계모듈과 IFC Manager는 모두 Microsoft Visual Studio 2010 환경에서 C#으로 개발하였다.

그리고 IFC 연계모듈을 기반으로 IFC Manager를 개발하는 과정에서 IFC에 대해 파악해야 하는 범위 및 상세한 정도, 건축구조물 대상 전문지식과 실무경험에 부합되는 물리적, 개념적 대상을 활용할 수 있는지 검토하였다. 대상 구조물은 26층 규모의 RC 벽식구조 아파트이며, 보, 기둥, 벽체, 슬래브가 포함되어 있다.

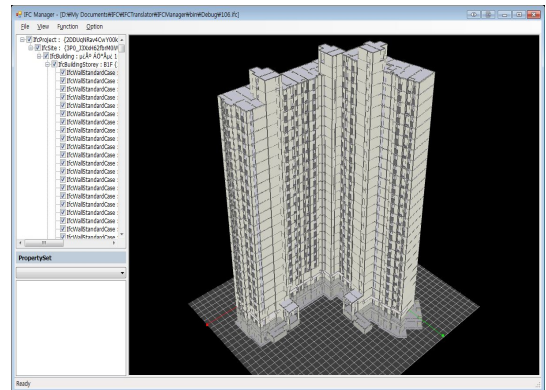
### 4.2 시각화 및 부재의 목록 표시

시각화를 위해서는 먼저 시각화 대상물의 전체 크기를 파악해서 뷰포트(viewport)를 설정한다. 대상물의 전체 크기는 Fig. 4의 pmModel에서 함수 Get MinMaxObjSize()를 호출해서 얻을 수 있다. pmModel에서는 그에 포함되는 각 부재의 절대좌표를 얻고 그 최대값과 최소값으로 대상물의 전체 크기를 결

정한다. 다음에는 개별 부재들의 형상을 실제로 시각화한다. 역시 pmModel에서 함수 Draw()를 호출한다. pmModel에서는 그에 포함되는 대지-건물-층-부재 순으로 Draw()를 호출함으로써 모든 부재 형상의 시각화가 완료된다.

부재의 목록은 pmModel이 ixObject로부터 상속받은 GetPhysicalObject()를 호출하면 얻을 수 있다. pmModel에서는 그에 포함되어 있는 대지-건물-층-부재의 순서로 GetPhysicalObject()를 차례로 호출하여 부재의 목록을 프로젝트-대지-건물-층-부재의 위계체계에 따라 생성한다.

Fig. 6는 IFC 파일을 OpenGL[13]로 시각화하고 부재의 목록을 트리(tree)형식으로 나타낸 결과이다.



[Fig. 6] The visualization and tree structures of members

### 4.3 물량 산출

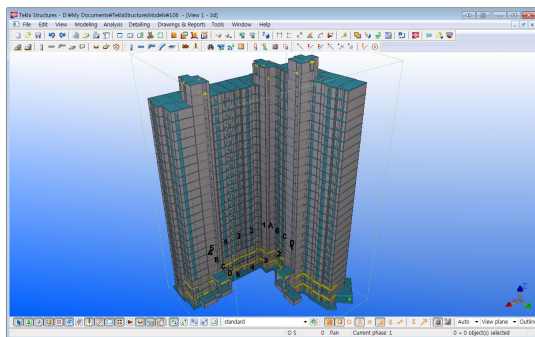
물량은 부재의 무게, 부피, 면적 등을 부재의 종류, 강도, 단면 크기, 그룹, 층 등으로 구분하여 더한 값으로서, pmModel의 GetQuantity()를 호출하면 얻을 수 있다. pmModel은 다시 그에 포함되는 대지-건물-층-부재 순으로 GetQuantity()를 호출하여 해당 부재의 무게, 부피, 면적을 계산하고 더한 후에 QTOBuilding, QTOStorey, QTOGroup, QTOMember에 저장한다. 이 엔터티들은 각각 건물, 층, 부재 그룹, 부재의 물량 합계와 그에 포함되는 부분의 물량을 단면 크기, 길이, 부피, 중량 등으로 표현하며 pmEntity에 포함된다. Fig. 7은 이렇게 얻은 물량을 MS-Excel로 저장한 결과이다.

story	group	height(H) (mm)	width(B) (mm)	length(L) (m)	quantity (ea)	concrete (m <sup>3</sup> )	form(m <sup>2</sup> )	rebar(m)
1F		2000	3,656.4	3.00	111	243.51	2,568.34	
		4000	2000	1.16	2	0.18	2.31	
		4000	2000	1.04	2	0.17	2.08	
		4000	2000	1.09	1	0.09	1.09	
		4000	2000	1.22	1	0.10	1.22	
		4000	2000	1.20	2	0.19	2.40	
		4000	2000	1.22	1	0.10	1.22	
		4000	2000	1.61	1	0.13	1.61	

[Fig. 7] Quantity Take-Off of the 1st floor

#### 4.4 Tekla와 연계

건축구조물 대상의 대표적 상업용 응용소프트웨어인 Tekla[14]와 연계하기 위해서는 프로젝트-대지-건물-층-부재의 포함관계, 부재의 식별을 위해 pmBuildingElement의 GlobalId, Name, Tag, 부재의 형상과 위치를 위해서 ixGeometryItem의 GetStartPoint()와 GetEndPoint(), ixLocalPlacement의 GetAbsolutePoint(), ixGSolid의 GetProfile(), 재료를 위해서 ixMaterial의 Name 등을 알고 있어야 한다. 그러면 Fig. 8에 나타난 바와 같이 IFC 파일에 저장되어 있는 자료를 변환하여 Tekla의 모델로 변환할 수 있다.



[Fig. 8] Interface between IFC file and Tekla

### 5. 결론

본 연구에서는 건축구조물 대상의 IFC 연계모듈 개발의 어려운 점을 분석하고, 이를 해결할 수 있는 방안을 제시하였다. 그리고 제시한 방안이 따라 적용사례로 건축구조물 대상의 IFC 연계모듈을 개발하고 그 타당성을

검토하였다.

이 과정에서 본 연구의 IFC 연계모듈을 구성하는 ixEntity와 pmEntity를 이용하여 건축구조물 대상의 전문지식과 실무경험에 부합되는 방식으로 IFC 파일에 저장되어 있는 자료들에 접근할 수 있었다. 그러므로 IFC에 대한 많은 지식이 필요 없었으며, 그에 따른 복잡함도 상당히 감소하였다.

이러한 검토결과로 볼 때, 본 연구에서 제안한 IFC 연계모듈 개발방안을 이용하면 IFC에 대한 개략적인 지식, 보다 적은 시간과 노력으로 IFC와 건축구조 분야의 여러 응용소프트웨어 사이에서 자료를 교환할 수 있는 다양한 연계모듈을 개발할 수 있을 것으로 기대된다.

### References

- [1] buildingSMART international, <http://www.buildingsmart.org>, (accessed Sep., 16, 2013)
- [2] buildingSMART international, IFC2x Edition 3 Technical Corrigendum 1, International Alliance for Interoperability, <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/index.htm>, (accessed Sep., 16, 2013)
- [3] Jakob Beetz, Léon van Berlo, Ruben de Laat, Pim van den Helm, "BIMserver.org - An open source IFC model server", *Proceedings of the CIB W78 2010, 27<sup>th</sup> International Conference-Cairo, Egypt, 15-18 November, 2010.*
- [4] Antti Karola, Hannu Lahtela, Reijo Hänninen, Rob Hitchcock, Qingyan Chen, Stephen Dajka, Kim Hagström, "BSPro COM-Server - interoperability between software tools using industrial foundation classes", *Energy and Buildings*, 34, pp.901-907, 2002. DOI: [http://dx.doi.org/10.1016/S0378-7788\(02\)00066-X](http://dx.doi.org/10.1016/S0378-7788(02)00066-X)
- [5] EDMmodelServer, <http://www.epmtech.jotne.com/edmmodelserver-ifc>, (accessed Feb. 24, 2014)
- [6] ifc-dotnet, <https://code.google.com/p/ifc-dotnet/>, (accessed Feb. 24, 2014)
- [7] IFC Engine, <http://www.ifcengine.com>, (accessed Feb., 24, 2014)
- [8] ST-Developer, <http://www.steptools.com/>, (accessed Feb. 24, 2014)
- [9] xBIM Toolkit, <http://xbim.codeplex.com/>, (accessed Feb. 24, 2014)
- [10] ISO, *ISO 10303-1:1994 Industrial Automation Systems and Integration - Product Data Representation*

*and Exchange - Part 1: Overview and Fundamental Principles*, ISO/TC 184/SC 4, 2004.

- [11] ISO, *ISO 10303-11:2004 Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 11: Description Methods: The EXPRESS Language Reference Manual*, ISO/TC 184/SC 4, 2013.
- [12] ISO, *ISO 10303-22:1998 Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 22: Implementation Methods: Standard Data Access Interface*, ISO/TC 184/SC 4, 2008.
- [13] OpenGL, <http://www.opengl.org>, (accessed Sep., 16, 2013)
- [14] Tekla, <http://www.tekla.com/products/tekla-structures>, (accessed Sep., 16, 2013)

---

## 정 종 현(Jong-Hyun Jung)

[정회원]



- 1993년 2월 : 서울대학교 건축학과 (공학석사)
- 1998년 8월 : 서울대학교 건축학과 (공학박사)
- 1998년 11월 ~ 1999년 10월 : 한국건설기술연구원 위촉연구원
- 1999년 11월 ~ 2002년 12월 : 현대건설 기술연구소 선임연구원
- 2003년 1월 ~ 2003년 8월 : 포스코건설 기술연구소 선임연구원
- 2003년 9월 ~ 현재 : 경남대학교 건축학부 부교수

<관심분야>

초고층구조, BIM