

개방형 BIM 기반 라이브러리의 IFD 활용을 위한 기초연구

조동완*, 김인한**, 서종철***, 김종헌****

A Study on Usage of IFD of Open BIM-Based Library

Dongwan Cho*, Inhan Kim**, Jongcheol Seo*** and Jonghun Kim****

ABSTRACT

A variety of BIM libraries are required to apply BIM(Building Information Modeling) software and generate BIM data. However, enterprises are establishing their own libraries using BIM tools, the drawbacks are that these libraries are not open to other users, or they just can be used in some software packages. If IFC(Industry Foundation Classes), the standard format of open BIM, can be applied, the BIM library can be applied in all the BIM software packages. Although the enterprises can collaborate using IFC standard, it is difficult to represent all the property information of the construction materials by IFC. IFD(International Framework for Dictionaries) is used to solve these problems and add the property information of the construction materials easily. Therefore, Open BIM-based library needs to be developed using IFD. In this research, the development and application of BIM library and IFD in domestic and foreign countries were investigated and analyzed, and Open BIM-based IFD scenario was proposed for using, and finally by applying an IFD method, the IFD data was added within an IFC model. Consequently, Open BIM-based library can be extended with rich, formal, and semantic descriptions and enhance data interoperability among BIM software packages adopting Open BIM standard format, by applying the IFD.

Key words : BIM Library, IFC, IFD, Open BIM

1. 서 론

BIM(Building Information Modeling)은 건축물의 세부적인 표현을 디지털화하여 장기간 보관이 가능하도록 해주며, 건설과정에서 발생하는 다양하고 방대한 정보를 효율적으로 관리할 수 있도록 해준다^[1]. 현재 BIM은 국내 건설산업에서 활발히 도입 및 적용되고 있으며, 건설과정에 참여하는 업체들 간의 협력작업 방식으로의 전환, 4차원 공정 시뮬레이션, 간접체크, 시공성 검토 등 BIM이 주는 이점은 건설 프로세스에 많은 변화를 가져오고 있다. 그러나 BIM 소프트

웨어들 간의 BIM 데이터 호환성 부족의 문제는 건설과정에 참여하는 업체들 간의 협력작업 방식으로의 전환을 어렵게 하고 있다. 이를 해결하기 위하여, 빌딩스마트 국제연맹(buildingSMART International)에서는 국제표준 정보모델인 IFC(Industry Foundation Classes)^[2]를 지속적으로 개발하고 있으며, 이는 BIM 데이터의 공유 및 교환을 위한 개방형 BIM(Open BIM)으로 불리고 있다^[3].

이처럼, 건설산업에서 BIM의 도입 및 적용이 확산됨에 따라 설계사는 BIM 데이터를 생성하기 위하여 수많은 BIM 라이브러리를 필요로 하게 되었다. 실제로 빌딩스마트협회에서 건설업체들을 대상으로 한 설문조사에 의하면 BIM 도입 및 적용을 활성화하기 위해서는 BIM 라이브러리 콘텐츠의 보급이 필요하다고 조사된 바 있다^[4]. 하지만 자재업체 또는 정보시스템에서 BIM 라이브러리의 제공은 아직까지 해외 사례에 비하여 미비한 상태이고, BIM 라이브러리를 제공하는 업체 및 단체에서는 특정 BIM 소프트웨어의 BIM 라이브러리만 제공하고 있다. 이는 라이브러리

*학생회원, (사)빌딩스마트협회 기술연구소 연구원, 경희대학교 건축학과 석사과정

**교신저자, 종신회원, 경희대학교 건축학과

***정회원, 경희대학교 건축학과 학술연구교수

****비회원, 경희대학교 건축학과

- 논문투고일: 2011. 02. 10

- 논문수정일: 2011. 03. 11

- 심사완료일: 2011. 03. 15

를 제작하는데 드는 시간이 많이 소비되고, BIM 소프트웨어의 버전에 따라 라이브러리를 재제작 해야 되는 문제점을 발생시킬 수 있다. 이러한 문제를 해결하기 위해서는 개방형 BIM 포맷으로 라이브러리를 제공하여야 하며, 이러한 방법은 BIM 소프트웨어의 버전에 관계없이 BIM 라이브러리의 장기간 보관, 관리, 재활용을 가능하게 한다¹⁾.

그러나 개방형 BIM 포맷인 IFC는 건설 프로세스에서 활용되는 모든 정보를 내부적으로 관리할 수 없다. 현재의 기술적 수준에서 IFC는 외부 데이터 또는 데이터베이스와 연계하여 실제 건설정보를 활용 및 관리할 수 있으나, 응용소프트웨어들 간의 양방향 데이터 교환을 위해서는 IFD(International Framework for Dictionaries)가 필요하다. 물량산출 시스템, 카탈로그, 캐드 시스템, 에너지 분석 프로그램 등 여러 응용 소프트웨어들 간의 필요로 하는 속성정보를 IFD를 통하여 입력이 가능하다.

따라서 본 연구에서는 개방형 BIM 기반 라이브러리의 정보체계 구축을 위하여 IFD 활용방안을 제시하였다.

연구의 방법 및 범위는 다음과 같다.

- 첫째, BIM 라이브러리 및 IFD에 관한 국내의 개발 현황을 조사 및 분석하여 문제점 및 시사점을 도출하였다.
- 둘째, IFD의 개발방법을 적용한 개방형 BIM기반의 IFD 활용시나리오를 제시하였다.
- 셋째, 개방형 BIM 라이브러리의 정보체계 구축을 위한 IFD 활용방안을 제시하였다.

2. BIM 라이브러리 및 IFD의 개발 현황

2.1 국내외 라이브러리 관련 현황 분석

국외 BIM기반의 건설자재정보 제공시스템, 또는 BIM 라이브러리를 제공하는 기관은 'Autodesk Revit Library¹⁾', 'Revit City²⁾', 'Objects Online³⁾', 'Autodesk Seek⁴⁾', 'Form Fonts⁵⁾', 'BIM Highway⁶⁾' 등이 있다¹⁾. Table 1에서 조사된 바와 같이 BIM 라이브러리를 제공하는 기관들은 특정 BIM 소프트웨어의 자체포맷(Native format)으로 생성된 라이브러리를 개별적으로

Table 1. 국외 사례 분석

이름	검색 방법	소프트웨어	장점/단점
Autodesk RevitLibrary	Revit Category	Revit	· 장점: Revit 버전 업그레이드에 의한 지속적인 업데이트 · 단점: 검색방법 및 라이브러리가 Revit에 한정되어 있음
Revit City	MasterFormat, Keyword	Revit	· 장점: Revit의 모든 버전에 대한 라이브러리 제공, 사용자들의 라이브러리 공유, 분류체계에 의한 검색방법 · 단점: Revit에 한정되어 있음
Objects Online	ObjectsOnline Category	ArchiCAD, AutoCAD, SketchUp	· 장점: 여러 프로그램에 대한 라이브러리 제공 · 단점: 특정 소프트웨어 라이브러리 제공
Autodesk Seek	MasterFormat, UNIFORMAT, Manufacturer	Revit, AutoCAD, Microstation, SketchUp	· 장점: 자재정보 및 자재업체에 대한 정보 제공, 분류체계에 의한 검색 방법 · 단점: AutoCAD사의 프로그램에 대한 라이브러리만 제공
Form Fonts	MasterFormat, Keyword, Platform, Manufacturer	SketchUp, AutoCAD, Revit, ArchiCAD, 3DMax, Lightwave	· 장점: 다양한 소프트웨어의 라이브러리 제공, 분류체계에 의한 검색 방법 · 단점: 하나의 라이브러리를 각각의 프로그램에서 직접 제작해야 함
BIM Highway	Revit Category, Project, MasterFormat, UNIFORMAT, Keyword, Manufacturer	AutoCAD, Revit	· 장점: 웹 기반 프로그램을 통한 라이브러리 제공, 여러 분류체계를 통한 검색 방법 · 단점: 특정 소프트웨어 라이브러리 제공

¹⁾Autodesk Revit Library Web Site, <<http://revit.autodesk.com/library/html>>

²⁾Revit City Web Site, <<http://www.revitcity.com/downloads.php>>

³⁾Objects Online Web Site, <<http://www.objectsonline.com>>

⁴⁾Autodesk Seek Web Site, <<http://seek.autodesk.com>>

⁵⁾Form Fonts Web Site, <<http://www.formfonts.com>>

⁶⁾BIM Highway Web Site, <<http://digitalbuildingsolutions.com>>

제공하고 있다. 또한, 몇몇 기관들은 MasterFormat과 UNIFORMAT 분류체계를 직용하여 사용자들로 하여금 자체정보를 쉽게 검색 및 조회할 수 있도록 지원하고 있다.

국내 건설자재정보 제공시스템을 운영하는 기관 및 단체는 한국건설기술연구원의 건설자재 정보시스템과 조달청의 나라장터, 한국토지주택공사가 있다⁶⁾. 아직 BIM 라이브러리를 제공하는 기관 및 단체는 없으며, 개인적인 웹사이트 및 커뮤니티에서 사용자들이 제작한 라이브러리를 공유하고 있는 실정이다. 한국건설기술연구원의 건설자재 정보시스템에서는 사용자들에게 자재정보에 대한 상세한 자료 및 도면을 제공하고 전자카탈로그와 전자거래를 지원하고 있으나, BIM 라이브러리의 제공은 미흡한 실정이다. 조달청의 나라장터에서는 물품에 대한 분류체계와 자재에 대한 상세한 정보를 제공하고 있지만, 도면이나 BIM 라이브러리는 제공하지 않고 있다.

2.2 IFD 개발 현황 분석

IFD는 용어 라이브러리, 분류체계 또는 온톨로지를 매핑하기 위한 구성체계로서 ISO 12006-3 분류체계를 기반으로 한다. IFD 개발의 목적은 개방형 표준 환경에서 교환정보에 대한 표준적인 이해와 함께 정보의 명시적 교환, 다국어 변환, 풍부한 표준정보를 제공하는데 있으며, 현재 캐나다, 네덜란드, 노르웨이, 미국의 참여로 라이브러리 형태로 개발되고 있다⁷⁾.

이러한 IFD 라이브러리를 개방형 BIM 데이터에 활용함으로써 새로운 속성정보를 추가할 수 있을 뿐만 아니라, IFD의 표준속성들이 다양한 어플리케이션의 정보모델에서 활용될 수 있다(Fig. 1)⁸⁾.

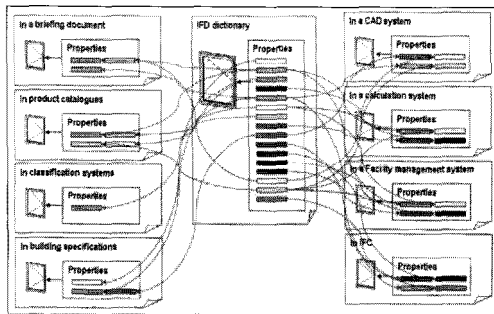


Fig. 1. IFD 매핑 개념도.

현재 IFD 웹사이트에서는 IFD의 소개, 프로그램 개발에 필요한 WSDL(Web Service Description Language) API(Application Program Interface) 문서

와 API 구현 방법 예시, IFD 응용소프트웨어와 IFC 모델에 대한 IFD 구현 및 활용 방법 등을 제공하고 있다. 하지만, 아직 국내에서 활용이 가능한 프로그램이 없는 실정이며 IFD 웹사이트에서 제공하는 WSDL API, IFD 응용소프트웨어를 활용한 프로그램의 개발이 필요하다.

IFC 모델내에 IFD의 정보를 추가하는 방법은 노르웨이의 표준협회⁹⁾에서 “SN/TS 3489⁹⁾”라는 기술문서로 제공하고 있으며, 여기에서는 총 4가지의 방법을 제시하고 있다(Table 2). 각각의 방법은 공통적으로 IFD 정보를 표현하기 위하여 IfcClassification을 생성한다. 첫 번째 방법은 IFC에서 표현되어지는 부재의 Entity에 대하여 IFD에서 정의하고 있는 부재의 Name과 GUID(Globally Unique Identifier)를 추가하는 방법이고, 두 번째 방법은 IFC에서 표현되는 부재의 타입객체를 생성한 후 이를 IFD에서 정의하고 있는 객체의 GUID를 추가하는 방법이다. 세 번째 방법은 IFC에서 정의하고 있는 부재의 Entity에 IFD에서 정의하고 있는 부재의 속성명과 GUID를 추가하는 방법으로, 부재를 속성명 표현하는데 있어서 미흡한 부분을 보완하기 위한 방법이다. 세 번째 방법이 IFD에서 정의하고 있는 객체의 속성명을 추가하는 것이 라면 네 번째 방법은 속성명 뿐만 아니라 속성값도 추가하는 방법이다.

2.3 분석결과 및 시사점 도출

국내의 사례 분석 결과 BIM 라이브러리를 제공하는 기관들은 BIM 소프트웨어별 자체포맷으로 생성된 BIM 라이브러리를 제공하고 있다. 즉, BIM 라이브러리 자체가 특정 BIM 소프트웨어의 버전별 자체포맷으로 생성되기 때문에, 이러한 BIM 라이브러리는 소프트웨어 버전 호환성의 문제를 야기시키며, 장기보관 및 관리와 재활용 측면에서 어려움을 줄 수 있다.

하지만 개방형 BIM의 표준 포맷인 IFC로 되어 있는 BIM 라이브러리는 기존의 BIM 라이브러리가 가지는 문제점을 해결할 수 있다. BIM 소프트웨어의 버전에 상관없이 다양한 BIM 소프트웨어에서 개방형 BIM 기반의 BIM 라이브러리를 사용할 수 있게 되며, 이는 BIM 라이브러리를 제작하는데 소비하는 시간과 비용을 절약시켜준다.

그러나 개방형 BIM 기반의 BIM 라이브러리는 건설 프로세스의 다양한 업무에서 활용되기에는 한계가 있다. 물량산출 및 에너지 분석 등과 같은 다양한 업무에서 활용되기 위해서는 업무에서 필요로 하는 속성정보를 입력하여야 하며, 현재의 기술적 수준에서

Table 2. IFC 모델에 IFD 정보를 추가하는 방법

	방법	설명	추가되는 Entity
1	IfcClassificationReference를 사용하여 IFD의 GUID와 Name을 연결	IFC의 객체(문, 벽, 창문, 기둥 등)와 IFD에서 정의하는 Name과 GUID를 연결	IfcClassification, IfcClassificationReference, IfcRelAssociatesClassification
2	IfcSpaceType, IfcDoorStyle, IfcCoveringType, IfcWindowStyle, IfcFurnishingElementType와 같은 비슷한 유형의 객체를 묶어주는 Entity를 추가하여 IFD의 GUID와 Name을 연결	IFC의 객체 중 비슷한 유형의 객체와 IFD에서 정의하는 Name과 GUID를 연결	IfcClassification, IfcClassificationReference, 객체의 비슷한 유형을 묶어주는 Entity, IfcRelDefinesByType, IfcRelAssociatesClassification
3	속성을 추가하기 위한 IfcPropertySingleValue와 이러한 속성을 묶어주기 위한 IfcPropertySet을 추가하고, IfcPropertyDependencyRelationship을 추가하여 IFD의 속성정보와 연결	IFC의 객체에 IFD에서 정의하는 객체의 속성명을 연결	IfcClassification, IfcRelDefinesByProperties, IfcPropertySet, IfcPropertySingleValue, IfcPropertyDependencyRelationship, IfcPropertyReferenceValue, IfcClassificationReference
4	속성을 추가하기위한 IfcPropertySingleValue와 이러한 속성을 묶어주기 위한 IfcPropertySet을 추가하고, IfcPropertyDependencyRelationship을 추가하여 IFD의 속성정보와 Value정보를 연결	IFC의 객체에 IFD에서 정의하는 객체의 속성명 및 속성값을 연결	IfcClassification, IfcRelDefinesByProperties, IfcPropertySet, IfcPropertySingleValue, IfcPropertyDependencyRelationship, IfcPropertyReferenceValue, IfcClassificationReference

IFC는 외부 데이터 또는 데이터베이스와 연계가 가능하도록 되어 있다. IFD는 업무에서 필요로 하는 속성 정보를 IFD 라이브러리를 활용하여 IFC에 추가할 수 있도록 해준다.

3. 개방형 BIM 기반의 IFD 활용 시나리오

2.2절에서 언급된 IFD의 개발방법 중 본 연구에서는 BIM 라이브러리의 IFD 연계 및 활용에 있고, 이를 위하여 활용시나리오를 제안하였다(Fig. 2).

IFD를 활용하기 위해서는 IFD 웹사이트에서 제공하고 있는 WSDL API를 활용하는 방법과 IFD 응용 소프트웨어를 활용하는 방법이 있다. WSDL API를 활용하는 방법은 IFD 라이브러리 서버와 연동하여 프로그램을 개발하는 방법이다. IFD 응용소프트웨어를 활용하는 방법은 개발자가 직접 속성정보를 정의하여 IFC 모델내에 IFD 정보를 입력하고, 정의한 속성정보와 IFD 라이브러리 서버에 등록되어 있는 속성과의 매핑을 통하여 개발하는 방법이다. 본 연구에서는 IFD 응용소프트웨어 중 IFD Library Propertyizer를 사용하여 활용시나리오를 제시하였다. IFD Library Propertyizer는 사용자가 직접 속성정보를 정의하여

XML로 저장하도록 해준다. 따라서 운영자 및 개발주체는 IFD Library Propertyizer를 통하여 객체에 대한 공통된 표준속성정보를 정의하여야 하며, BIM 라이브러리 공급자는 표준속성정보를 BIM 라이브러리에 입력하여 건설자재 정보시스템에 업로드하게 된다. 사용자는 표준속성정보가 입력된 정보로부터 건설프로젝트에 활용하게 되며, 생성된 BIM 데이터는 물량산출 및 에너지 분석, 견적서와 같은 다양한 업무에서 활용할 수 있다.

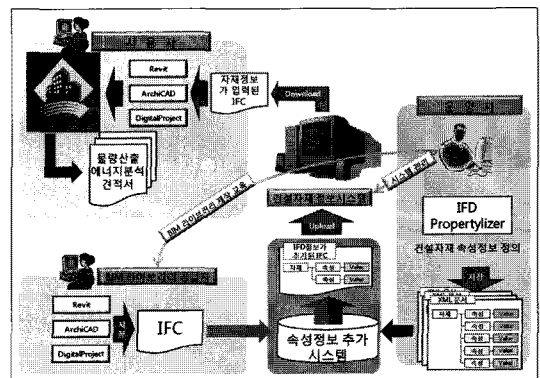


Fig. 2. 건설자재 정보시스템에서 개방형 BIM 기반의 IFD 활용시나리오.

4. 개방형 BIM 기반 라이브러리의 IFD 활용을 위한 기초연구

4.1 연구의 개요

본 연구에서는 IFD Library Propertyizer를 통하여 객체의 속성과 Value를 정의하고 이를 IFC 모델에 입력하는 과정을 저했다(Fig. 3).

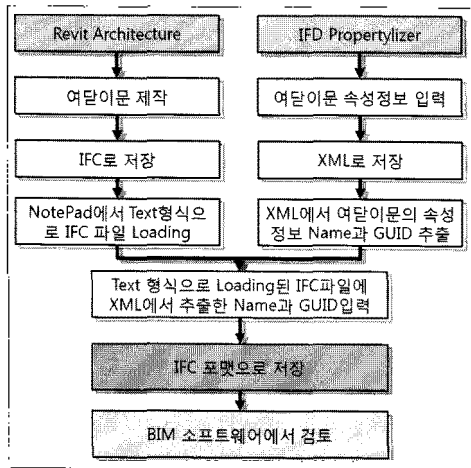


Fig. 3. IFC에 IFD 속성 입력을 위한 연구절차.

이를 위하여 IFD 정보를 입력하기 위한 대상을 선정하였고, Table 2의 IFC 모델에 IFD 정보를 추가하는 방법 중 네 번째 방법을 활용하였다.

4.2 IFC 모델에 IFD 속성 입력 및 활용

4.2.1 연구대상 선정

IFD Library Propertyizer를 통하여 표준속성정보를 정의하기 위하여 본 연구에서는 조달청의 분류체계^[10]를 참고하였다. 조달청에서는 물품에 대한 분류체계를 대분류부터 속성정보까지 정의하고 있으며, 이 중에서 본 연구에서는 여담이문에 대한 속성정보를 참고하였고, 다음 Table 3은 여담이문에 대한 속성명과 속성값을 표현하였다.

4.2.2 IFD 데이터 생성

Table 3의 속성정보를 IFD 데이터로 생성하기 위하여 IFD Library Propertyizer를 활용하여 XML로 정의하는 과정을 저했다(Fig. 4). IFD Library Propertyizer는 속성명 및 속성값을 정의할 수 있는 응용프로그램으로써 최상위 분류부터 최하위 분류까지 정의할 수 있다. 따라서 본 연구의 대상이 되는 여담이문의 최상위 분류인 전자제품부터 다음 분류인 창호 및 유리 그리

Table 3. 여담이문의 속성명 및 속성값

품명	속성명	표기방법	속성값
여담이문	물품목록번호	시설자재구분코드 + 물품식별코드	30171511-12345678
	물품분류번호	시설자재구분코드	30171511
	물품식별번호	물품식별코드	12345678
	영문명	Text	Swing door
	상품원산지국가명	국가코드	KR
	품목등록일	날짜	2010.12.28
	모델명	Text	SD_123
	한글품목명	Text	여담이문, 제조업체명, SD 123, 1735×2053×44 mm. 외여담이문·양소대
	제조업체명	Text	제조업체명
	제조업체 사업자번호	제조업체사업자코드	987654321
	문두께	숫자(mm)	44
	문재질	Text	Wood
	표면처리	Text	친환경 페인트 마감
	폭	숫자(mm)	1735
높이	숫자(mm)	2053	

고 문, 여담이문 순서로 속성정보의 상위정보를 정의하였고, 여담이문의 속성명 및 속성값을 정의하였다.

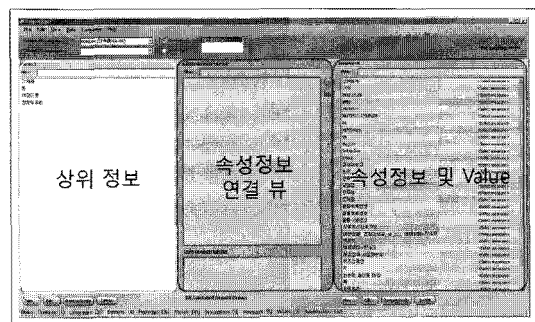


Fig. 4. IFD Library Propertyizer를 활용한 여담이문의 속성 정보 정의.

IFD Library Propertyizer를 통하여 정의한 모든 속성정보에는 고유의 ID 값인 GUID가 포함되어 있다. GUID는 IFC에 IFD정보를 입력하기 위하여 필요하며, 이를 위해서 IFD Library Propertyizer를 통하여 정의한 속성명 및 속성값의 GUID를 추출하였다

(Fig. 5). 추출한 GUID는 사용자의 컴퓨터에서 생성된 GUID로써 추후 IFD 라이브러리 서버에 등록되어 있는 속성과의 매핑에 관한 연구가 이루어져야 한다.

```

<preferred_by />
</IfdName>
<IfdName id="48e167e756624f6a9ca1ac15756922c4" IfdNameTypeEnum="FullIz"
<name>역달이문</name>
<language>
<IfdLanguage ref="1VzhOcobahsm00W01U3_W0" />
</language>
<preferred_by />
</IfdName>
<IfdName id="531d0ff2db814ac0a1ecbf3058110e68" IfdNameTypeEnum="FullIz"
<name>물품목록번호</name>
<language>
<IfdLanguage ref="1VzhOcobahsm00W01U3_W0" />
</language>
<preferred_by />
</IfdName>
<IfdName id="9c9a51a3b67849b08d5692aaef52558" IfdNameTypeEnum="FullIz"
<name>물품목록번호</name>

```

물품목록번호에 대한 GUID

Fig. 5. 물품목록번호에 대한 GUID.

4.2.3 IFC 모델에 IFD 속성 입력 및 검증

Autodesk사의 Revit Architecture™를 이용하여 여단이문의 IFC 파일을 생성한 후, IFD Library PropertyLizer를 통하여 생성된 GUID와 속성정보의 이름을 입력하기 위하여 Text형식으로 불러왔다. IFC를 Text형식으로 불러왔을 경우 IFC에 저장되어 있는 정보가 스키마 형식으로 불러와지며, 이를 통하여 IFD 정보를 스키마 형식으로 입력이 가능해진다.

Table 4는 Text형식으로 Loading된 IFC파일에 IFD 정보를 추가하는 과정을 나타낸다. 첫 번째로, IFD의 속성정보 표현을 위하여 IfcClassification을 생성한다. IfcClassification의 Attribute는 Source와 Edition, EditionDate, Name을 가지며, EditionDate를 제외하고는 필수적으로 표기해야 한다. Source에서는 IFD 속성의 출처를 기재하고, Edition에서는 사용한 언어를 표기한다. Name에서는 IIFD Library라는 것을 명시하여야 한다.

두 번째로, 여단이문과 속성정보를 연결시키기 위하여 IfcRelDefinesByProperties를 생성한다. IfcRelDefinesByProperties의 Attribute는 GUID, OwnerHistory, Name, Description, Object, IfcPropertySet으로써 Name과 Description을 제외하고는 필수적으로 표기해야 한다. GUID는 생성된 IfcRelDefinesByProperties의 고유 GUID를 생성하여야 하고, OwnerHistory는 IFC 파일에서 IfcOwnerHistory의 넘버값인 #33을 입력하였다. Object는 대상이 되는 객체로서 본 연구에서는 여단이문의 Entity인 IfcDoor의 넘버값 #207을 입력하였고, IfcPropertySet은 이후 추가될 IfcPropertySet의 넘버값이 입력된다.

세 번째로, 추가되는 IFD의 속성정보를 묶어주기 위하여 IfcPropertySet을 생성하였다. IfcPropertySet의

Attribute는 GUID, OwnerHistory, Name, Description, IfcPropertySingleValue로써, GUID는 생성된 IfcPropertySet의 고유 GUID를 생성한다. OwnerHistory는 두 번째로 추가되는 속성과 동일하게 IfcOwnerHistory의 넘버값인 #33을 입력하고, Name은 표현되는 속성들을 묶어주기 위한 이름을 표기한다. Description은 Name에서 표기한 이름을 정의하기 위하여 사용되어지며, IfcPropertySingleValue는 이후 추가될 IfcPropertySingleValue의 넘버값이 입력된다.

네 번째로, 속성정보를 표현하기 위하여 IfcPropertySingleValue를 생성하였다. IfcPropertySingleValue의 Attribute는 Name, NominalValue, Type, Unit이 있으며, NominalValue와 Unit을 제외한 나머지 Attribute는 필수적으로 표기해야 한다. Name에서는 IFD에서 정의한 속성정보의 이름이 입력되고, Type에서는 IFD에서 정의한 속성정보의 속성값이 입력 된다.

다섯 번째로, IfcPropertySingleValue와 이후 추가될 IFD 속성명 및 속성값의 정보구분을 위한 IfcPropertyReferenceValue를 연결시켜 주기 위하여 IfcPropertyDependencyRelationship을 생성한다. IfcPropertyDependencyRelationship의 Attribute는 DependingProperty, DependantProperty, Name, Description, Expression이 있으며 Name, Description, Expression을 제외한 다른 Attribute는 필수적으로 표기해야 한다. DependingProperty는 IfcPropertySingleValue의 넘버값을 입력하게 되고, DependantProperty는 이후 추가될 IFD 속성명 및 속성값의 정보구분을 위한 IfcPropertyReferenceValue의 넘버값을 입력하게 된다.

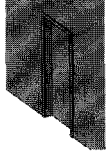
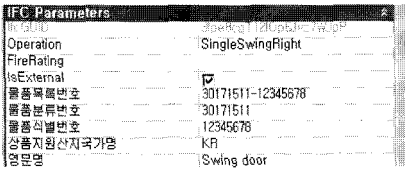

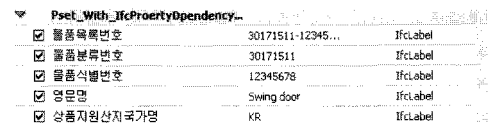

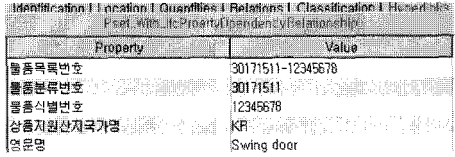

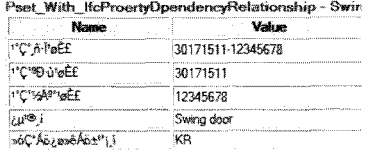
여섯 번째로, IFD의 속성명 또는 속성값의 구분을 위한 IfcPropertyReferenceValue를 생성한다. IfcPropertyReferenceValue의 Attribute는 Name, Description, UsageName, PropertyReference이 있으며, Description과 UsageName를 제외한 나머지 Attribute는 필수적으로 표기해야 한다. Name에서는 IfdProperty 또는 IfdValue를 입력하여 IFD정보를 구분하게 되며, PropertyReference에서는 이후 추가되는 IfcClassificationReference의 넘버값이 입력된다.

일곱 번째로, 마지막으로 추가된 속성정보에 IFD에서 정의한 속성정보의 GUID를 입력하고 첫 번째로 생성한 IfcClassification과 연결하기 위하여 IfcClassificationReference을 생성한다. IfcClassificationReference의 Attribute는 Location, IFD GUID, Name, ReferencedSource이 있으며, Location에서는 IFD 정보의 출처를 입력한다. IFD GUID는 추가된 속성명 및 속

Table 4. 여닫이문의 IFC 파일에 IFD의 속성정보 중 물품목록번호와 속성값 입력

순서	Entity	Entity에 입력되는 Attribute
	#33=IFCOWNERHISTORY(#32,#2,\$,NOCHANGE,\$,\$,\$,0); #207=IFCDOOR('3Ipe8cq112I0p6Jvz7WJpP',#33,'M_Single-Flush:1735 x 2053mm:1735 x 2053mm:113156,S,1735 x 2053mm',#206,#201,113156,2053,1735);	IFC 파일에서 IFD 정보를 입력하기 위한 정보 추출
1	#293=IFCCLASSIFICATION('IFD Library PropertyLizer','kr',\$,IFD Library');	Source, Edition, EditionDate, Name
2	#294=IFCRELDEFINITIONBYPROPERTYSET('35YdWMmwr4rQ61AZPsisfp7',#33,\$,\$,#207),#295);	GUID, OwnerHistory, Name, Description, Object, IfcPropertySet
3	#295=IFCPROPERTYSET('m35YaasdfwrQ61AZPsisfp7',#33,'Pset_With_IfcPropertyDependency','Swing Door',(#296,#303,#310,#317,#324));	GUID, OwnerHistory, Name, Description, IfcPropertySingleValue
4	#296=IFCPROPERTYSINGLEVALUE('X\B9\X\B0\X\C7\X\B0\X\B8\X\0\X\B8\X\F1\X\B7\X\CF\X\B9\X\F8\X\C8\X\A3',\$,IFCLABEL('30171511-12345678'),\$);	Name, NominalValue, Type, Unit
5	#297=IFCPROPERTYDEPENDENCYRELATIONSHIP(#296,#299,\$,\$,\$); #298=IFCPROPERTYDEPENDENCYRELATIONSHIP(#296,#300,\$,\$,\$);	DependingProperty(IfcProperty), DependantProperty(IfcPropertyReference Value), Name, Description, Expression
6	#299=IFCPROPERTYREFERENCEVALUE('IfdProperty',\$,\$,#301); #300=IFCPROPERTYREFERENCEVALUE('IfdValue',\$,\$,#302);	Name, Description, UsageName, PropertyReference(IfcClassificationReference)
7	#301=IFCClassificationReference('IFD Library PropertyLizer','531ddffbb814a60a1c6bf3058f1be80',\X\B9\X\B0\X\C7\X\B0\X\B8\X\F1\X\B7\X\CF\X\B9\X\F8\X\C8\X\A3',#293); #302=IFCClassificationReference('IFD Library PropertyLizer','9b7277f38e544dba85d1a64244693fc1',30171511-12345678,#293);	Location, IFD GUID, Name, ReferencedSource(IfcClassification)

Table 5. IFD정보가 추가된 IFC파일의 검토

	출력 화면	속성 화면	비고
Revit Architecture™			IFD의 속성명 및 속성값이 IfcPropertySet에서 정의한 Pset Name에 표현되지 않고, IFC Parameters에 포함되어 있음
ArchiCAD™			이상 없음
SMC			이상 없음
DDS CAD Viewer			소프트웨어의 한글 지원 문제로 인하여 한글이 이상하게 보임

성값의 GUID를 입력하고, Name은 추가된 속성명 및 속성값을 입력한다. ReferencedSource는 IfcClassification의 넘버값을 입력한다. 이러한 순서로 속성 정보와 Value정보를 순차적으로 입력하고 IFC 포맷으로 저장하게 되면 입력하고자 하는 객체의 IFD 정보가 추가된다.

IFD 속성정보가 추가된 IFC파일을 BIM 소프트웨어를 통하여 검토하는 과정을 거쳤다(Table 5). Revit과 ArchiCAD, SMC(Solibri Model Checker)에서는 추가된 속성정보와 Value정보가 자세히 표현되었다. 하지만 Revit에서는 IFD 정보를 추가하는 과정에서 IfcPropertySet에서 정의한 Pset_With_IfcProertyDependencyRelationship이 표현되지 않고, 추가된 속성정보가 IFC Parameters에 표현된 것을 확인할 수 있었다. 이는 Revit의 새로운 Pset 추가기능이 미흡한 부분이며 소프트웨어사의 해결이 필요하다. 또한 DDS CAD Viewer에서는 속성정보의 한글표현에 문제가 발생하여 한글이 이상하게 보이는 현상을 발견하였다. 이는 DDS CAD Viewer의 한글지원에 대한 문제로서 소프트웨어사의 한글지원에 대한 해결이 필요하다.

IFC 모델에 대한 IFD 정보를 추가하는 과정을 거쳤고, 개방형 BIM을 지원하는 소프트웨어에서 검토를 하였다. 이를 통하여 IFD 정보가 IFC 모델에 입력된 것을 확인할 수 있었으며, 몇몇 소프트웨어의 기능 개선이 필요한 것을 확인할 수 있었다.

5. 결 론

국내에서 BIM의 적용과 인식이 확산되는 상황에서, 본 연구는 개방형 BIM 기반 라이브러리의 IFD 활용을 위한 기초연구를 진행하였다. 국내의 사례를 조사 및 분석한 결과 개방형 BIM을 기반으로 하는 BIM 라이브러리 제공 시스템은 거의 없었고, BIM 라이브러리를 제공하는 시스템은 특정 소프트웨어에 국한되어 있었다. 이를 통하여 개방형 BIM의 필요성을 도출하였고, 개방형 BIM이 갖는 한계점을 해결하기 위하여 개방형 BIM 라이브러리의 IFD 활용 필요성을 도출하였다. 이를 기반으로 개방형 BIM 기반 IFD 활용시나리오를 제시하였으며, 활용시나리오를 기반으로 본 연구에서는 IFD 웹사이트에서 제공하는 응용소프트웨어를 활용하여 IFC 모델에 IFD 속성정보를 입력하는 방법에 대하여 연구하였다. IFC 모델에 IFD 속성을 추가하기 위해서는 여러 Entity가 추가되어야 하며, 이러한 Entity는 서로 연관관계를 가지면서 IFD 정보를 표현하게 된다. 이러한 방법으로 IFD 정보가

추가된 IFC 파일을 여러 BIM 소프트웨어에서 검토한 결과 IFD의 속성명과 속성값이 표현된 것을 확인할 수 있었으며, 몇몇 소프트웨어에서는 소프트웨어의 기능 개선이 필요한 것을 확인할 수 있었다. 이와 같은 연구를 통하여 IFD 정보가 추가된 개방형 BIM 라이브러리를 유통시킨다면 서로 다른 BIM 소프트웨어에서 개방형 BIM 라이브러리를 사용함으로써, 보다 효율적으로 BIM 데이터를 확보할 수 있고, 물량산출 및 견적과 에너지 분석 등 다양한 업무에서 활용이 가능할 것이다.

본 연구는 조달청의 분류체계를 참고하여 연구를 진행하였을 뿐 속성에 대한 정의는 언급하지 않았다. 추후 Library에 대한 표준 속성을 정의하는 연구가 이루어져야 한다. 또한, IFD 응용소프트웨어를 활용하여 속성정보를 정의하여 사용하였을 뿐 IFD 라이브러리 서버에 등록되어 있는 속성정보를 사용하지는 않았다. 추후 IFD 응용소프트웨어를 통하여 발생한 속성정보와 IFD 라이브러리 서버에 등록되어 있는 속성정보와의 매핑에 관한 연구가 필요하다. 또한, IFD 응용소프트웨어를 활용하는 방법 이외에 IFD 웹사이트에서 제공하는 WSDL API를 활용하는 방법에 대한 연구가 이루어져야 한다.

감사의 글

본 연구는 국토해양부 첨단도시개발사업의 연구비 지원(과제번호#09 첨단도시A01)에 의해 수행되었습니다.

참고문헌

1. Chuck Eastman *et al.*, "BIM HANDBOOK," John Wiley & Sons, Inc., New Jersey, 2008.
2. IFC Web Site, <<http://www.buildingsmart-tech.org/>>
3. 서종철, 김인한, "국내 건설 공공발주에서 BIM의 도입 및 적용을 위한 기본방향에 관한 연구," 대한건축학회 논문집, Vol. 25, No. 9, pp. 21-30, 2009.
4. 최중식, 김인한, 조찬원, 최중현, "국내건설산업의 개방형 BIM적용 현황 및 발전 방향", 한국 CAD/CAM학회 논문집, 제14권, 제6호, pp. 355-363, 2009.
5. Owolabi, A., Anumba, C. J. and El-Hamalawi, A., "Architecture for Implementing IFC-based Online Construction Product Libraries," ITcon Vol. 8, Special Issue IFC - Product models for the AEC arena, pp. 201-218, 2003.
6. 주기범, 한충한, 송종관, "건설자재정보의 유기적 활

용을 위한 역할기반 자재정보 운영방안,” 한국건설
관리학회 학술발표대회 논문집, pp. 565-570, 2009.
11.

- 7. buildingSMART, “IFD Library White Paper” <<http://www.ifd-library.org/>>
- 8. 노르웨이의 표준협회 Web Site, <<http://www.standard.no>>

- 9. 노르웨이 표준협회, “Implementation of IFD Library support in IFC,” <<http://www.standard.no/Global/PDF/Standarder%20p%20C3%A5%20h%20C3%B8ring/SN-TS%203489%20Implementation%20of%20IFD%20Library%20support%20in%20IFC.pdf>>
- 10. 조달청 북록정보시스템 Web Site, <<http://www.g2b.go.kr:8100/index.jsp>>



조 동 완

2009년 경희대학교 건축학과 학사
2009년~현재 경희대학교 일반대학원
건축학과 석사과정
관심분야: BIM(Building Information
Modeling), IFC, IFD, 건축정보
기술



김 인 한

1988년 서울대학교 건축학과 졸업
1991년 미국 Carnegie-Mellon 대학
건축학 석사
1994년 영국 Strathclyde 대학 건축학
박사
1996년~현재 경희대학교 공과대학
건축학과 교수
2002년~현재 한국CAD/CAM 학회 이사
2004년~2008년 사단법인 STEP센터
회장, 지식경제부
2008년~현재 빌딩스마트협회 수석
부회장
2010년~현재 대한건축학회 이사
관심분야: BIM(Building Information
Modeling), CAAD, 데이터모델링
및 통합 전산설계환경(STEP,
IFC), 건축정보기술, Digital
Design Media



서 종 철

2008년 경희대학교 건축공학과 공학박사
2006년~2008년 (사)스택센터 건설
표준팀 선임연구원
2008년~2011년 (사)빌딩스마트협회
기술연구소 부연구위원
2011년~현재 경희대학교 건축학과
학술연구교수
2011년~현재 한국CAD/CAM학회 논문
편집위원
관심분야: 건축계획 및 디자인이론, 데
이터 시뮬레이션 & 최적화, 객체
지향데이터모델(BIM, IFC표준)
및 건축 IT



김 종 훈

1984년 경희대학교 건축공학과 졸업
1989년 이탈리아 밀라노 공과대학
건축과 졸업
1990년 이탈리아 국가건축사 자격증 취득
1990년 I. Gianola 건축설계사무소 근무
1991년~1994년 DESITECH 산업디자인
사무소 근무
1994년~현재 경희대학교 공과대학
건축학과 조교수
연구분야: 건축설계