

항만 IFC Viewing을 위한 형상 및 속성 정보 추출 방법에 관한 연구

A Study on the Method of Extracting Shape and Attribute Information for Port IFC Viewing

김근호¹⁾, 박남규²⁾, 주철범³⁾, 김성훈⁴⁾

Kim, Keun-Ho¹⁾ · Park, Nam-Kyu²⁾ · Joo, Cheol-Beom³⁾ · Kim, Sung-Hoon⁴⁾

Received July 15, 2021; Received September 28, 2021 / Accepted September 28, 2021

ABSTRACT: An IFC file is dependent on the IFC schema. Because of this relationship, most IFC-using software reads and interprets the IFC File by employing an early binding method, which uses a standard IFC schema. In the case of most open sources, early binding methods using standard IFC schema have a problem that they cannot express extra information of IFC File out of extended IFC schema. Also, in the case of previous studies, they suggested schema extension, such as adding attribute information to the schema, rather than the interpretation of IFC File. This study research on method of extracting shape and attribute information was conducted by analyzing the IFC File produced through the Port schema, which is an extended IFC schema. Three objects were created using the reference relationship between the Port schema definition and the IFC entity, and, at the end, the three objects were combined into one object. It has been confirmed that the shape and property data were express properly while delivering the combined object to the viewer. The process is possible because of the method of matching IFC schema and IFC File, which is dependent on IFC schema but not early binding method. However, this method has some drawbacks, such that contemporaneously generated many objects consume many memory spaces. Future research to investigate that issue further is needed.

KEYWORDS: IFC, IFC Schema, Early Binding, IFC Entity, Viewer

키워드: IFC, IFC 스키마, 정적 바인딩, IFC Entity, 뷰어

1. 서론

1.1 연구의 배경 및 목적

건설 산업은 도로, 항만, 철도, 토목 등 다양한 분야가 존재한다. 이런 다양한 건설 산업 분야에서 요구되는 많은 정보를 처리할 수 있는 것이 BIM(Building Information Modeling)기술이다. 많은 건축 설계 소프트웨어들은 BIM 기술을 적용하였는데, 각자의 포맷으로 설계된 모델 간 호환이 되지 않는 문제가 발견됐다.

BIM 용 소프트웨어들 간의 정보교환, 즉 호환을 위해 나온 포맷이 IFC이다(buildingSMART, 2021). IFC란 Industry Foundation Classes의 약어로 'building SMART'가 개발하여 서로 다른 소프트웨어들 간의 정보 교환을 가능하게 하기 위한 포

맷이다. 이러한 이유로 IFC 포맷은 ISO에 의해 ISO/PAS 16739와 ISO/IS 16739 등록이 되었다.

IFC는 특정 소프트웨어 기업만의 포맷이 아닌 모두가 사용할 수 있는 데이터 모델 포맷으로서 BIM 설계 시 표준이 되었다.

현재 IFC의 가장 최신 버전은 IFC 4.3RC2로서 처음 IFC 1.0으로부터 지속해서 업그레이드되었다. 버전은 기술적인 문제 해결 및 새로운 건축 분야가 추가될수록 업그레이드되었다.

그러나 꾸준히 발전됨에도 불구하고 여전히 해결해야 할 문제가 다수 존재한다. IFC는 3D 형상과 속성정보를 표현하지만 많은 건설 산업 분야의 데이터를 담아내지 못하는 부분이 있다. 실제로 IFC의 데이터 일부 중 특정 속성 정보에 대응하지 않는 BIM 소프트웨어도 많다(Kim, 2017).

¹⁾정회원, ㈜태성에스엔아이 기술연구소 선임연구원 (khkim@tssni.com)

²⁾정회원, ㈜태성에스엔아이 기술연구소 기술연구소장 (nkpark@tssni.com) (교신저자)

³⁾정회원, ㈜태성에스엔아이 기술개발부 이사 (cbjoo@tssni.com)

⁴⁾정회원, ㈜태성에스엔아이 대표이사 (shkim@tssni.com)

또한, BIM 소프트웨어에서 IFC File Export 시 새로운 추가 속성 정보를 수동으로 입력하여 내보내는 것은 지원하지 않지만, IFC File Import 시 표준 IFC 스키마의 정의 외의 정보 존재할 경우 해당 정보를 보여주지 않는 경우가 대부분이다.

1.2 오픈소스 조사 및 분석

IFC에서 형상 정보 및 속성정보를 추출하는 과정을 파악하기 위해 관련 오픈소스들의 조사 내용은 아래와 같다.

BIMserver는 JAVA로 개발되었으며, IFC를 저장하고 프로젝트의 정보를 관리를 할 수 있는 오픈소스다. IFC 통합 관리 플랫폼으로 정의할 수 있으며, 기본 제공되는 기능 외 추가적인 기능은 플러그인을 설치할 경우 IFC의 구조물 속성정보 및 형상 정보를 볼 수 있다(BIMserver, 2021).

xbim은 .Net 오픈소스 소프트웨어 개발 BIM 도구 세트라 할 수 있다. 조작과 관련한 라이브러리는 C#으로 개발되었으며, 구조물의 형상 정보를 표현하는 기하학 엔진은 C++로 작성되어 있다(xbim, 2021).

apstex IFC Framework는 JAVA로 개발되었으며, IFC 정보에 접근하고 시각화하기 위한 개발 도구 세트다. IFC Java Toolbox, Geometry Tools, IFC Loader, IFC Java GUI Components, IFC Java Viewer, IFC WebGL Viewer 총 6가지 도구가 있으며 IFC Java Toolbox는 정적바인딩 방식으로 IFC Entity에 맞는 Java Classes를 이용해 객체화한다(apstex, 2021).

BIMserver, xBIM은 IFC 스키마의 Entity 별로 정보를 저장할 파일들이 만들어져 있고 IFC를 로드할 시 정적바인딩 방식을 통해 IFC Entity를 객체화한다. apstex IFC Framework의 IFC Java Toolbox 역시 같은 방식으로 동작한다. 각각의 오픈소스의 소스 코드를 분석한 결과 IFC2.3부터 IFC4.2 버전의 스키마로 제작된 IFC를 로드할 수 있었지만 xBIM을 제외하고 최근 스키마인 IFC4.3을 지원하지 않는다. xBIM은 지속해서 업데이트되어 새로운 Entity 및 속성 정보가 추가되었으나 나머지 오픈소스의 경우 IFC의 업데이트가 반영되지 않았다. 이를 통해 IFC 소프트웨어 및 프로그램들은 스키마에 종속적인 관계를 갖는다는 점을 알 수 있다.

오픈소스 뷰어들의 IFC를 읽고 해석하는 모듈은 표준 스키마에 종속적인 관계가 있는 경우가 대부분이었다. 각 모듈은 표준 스키마 상의 정의된 Entity만을 표현할 뿐, 새로운 속성정보가 추가된 확장된 스키마를 활용해 만들어진 IFC File의 데이터는 전부 표현하지 못하는 문제가 있다. 이것은 IFC를 읽고 해석할 때, 프로그램상의 표준 스키마에 정의된 내용에 따라 Entity를 객체화시키기 때문이다. 오픈소스 뷰어들의 경우 IFC의 해석을 Figure 1과 같이 IFC 내의 여러 IFC Entity에 따라 분류한다. 이때, 표준 스키마의 IFC Entity 정의에 맞춰 IFC 상의 Entity를 객체화하는

방식을 정적바인딩(Early binding) 방식이라 한다. 오픈소스 및 BIM 소프트웨어들은 표준 스키마의 Entity 정의에 맞춰 생성될 객체의 정보를 입력할 파일들을 소프트웨어 배포 전 미리 만들어 놓는다. 이후 IFC를 읽고 해석할 때, 미리 만들어진 파일들을 이용해 Entity 명칭에 맞추어 정보들을 저장할 객체를 만든다. 이런 정적바인딩 방식은 프로그램 내의 스키마에 맞춰 객체화할 파일들을 만들어 놓기 때문에 새로운 스키마 및 확장된 스키마를 통해 제작된 IFC에 대한 대응이 어려운 문제가 있다.

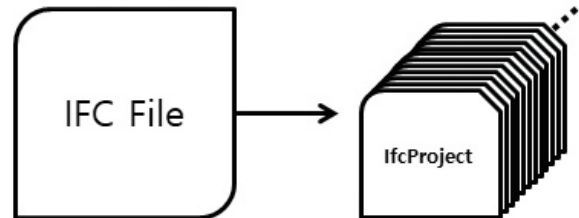


Figure 1. Early binding example

스키마에 의존적인 정적바인딩 방식이 아닌 다른 방법을 통해 Entity를 객체화시킨다면 Entity에 새로운 속성 정보가 추가될 경우에도 이를 표현할 수 있을 것이다.

1.3 선행 연구 고찰

IFC는 스키마의 정의에 따라 Entity의 속성정보가 결정된다. 표준 스키마는 모든 건설 분야의 정보를 담는 데 한계가 있다.

Table 1. Research on IFC schema extension

Researcher	Contents
Cho & Joo (2014)	Using the standard model converted through the new schema, It has established a delivery and verification system that fits the ordering system and showed various possibilities for use.
Kim et al. (2014)	The IFC data schema was extended to enable the representation of earthwork objects.
Won et al. (2019)	Based on the facility classification of the river WBS, components that can be used in common were derived, and an attribute set that complements the components was developed.
Han et al. (2016)	By developing the Extension IFC Viewer, which allows the BIM model to utilize design information at the construction site, a methodology to utilize the extended schema-based library was presented.
Song et al. (2013a)	Based on the IFC schema-based process information package configuration, the work and schedule using the process control information items are expressed. In addition, by defining information items, the standard construction period can be expressed.
Won et al. (2018)	The river sector schema extension plan was presented by classifying the river facility components according to the IFC class.
Song et al. (2013b)	The schema of the BIM model is analyzed to derive process information and to present a way to effectively utilize it.

이러한 부분을 극복하기 위해 표준 스키마의 Entity를 새로 정의하거나 속성정보를 추가하는 등의 방법과 같은 스키마 확장에 관한 연구와 IFC에서 형상 정보와 속성 정보를 추출하여 새롭게 가시화하는 등의 연구가 진행되었다. 스키마의 확장과 관련된 연구에 대한 내용은 Table 1과 같으며, IFC 분석 및 형상화 표출에 관련된 연구는 Table 2와 같다.

Table 2. Research on IFC file analysis and shape expression

Researcher	Contents
Park [2012]	Shape information was created in IFC File using open source and analyzed with spatial network analysis tool.
Ryu et al. [2015]	A new visualization method was proposed by extracting the shape information of the IFC file.
Goh et al. [2009]	It shows how to visualize and extract shape information from IFC File using a program. However, since it is a method through early binding, the dependencies and limitations were mentioned.

IFC의 구조물의 정보 표현의 한계를 지적한 연구들은 IFC 제작에 필요한 스키마에 집중하였다. 해당 연구들은 IFC Entity에 새로운 속성을 추가함으로써 표현할 수 있는 정보의 확장을 얻고자 했다.

선행 연구들은 표준 스키마의 표현 한계와 같은 부족한 부분을 개선하려는 시도가 많았다. 그러나 개선된 스키마를 활용해 만들어진 IFC File을 뷰어 상에서 읽고 해석할 때 발생하는 문제점을 지적하지 않았다.

이와 마찬가지로 본 연구에선 사용한 ‘한국건설기술연구원’에서 표준 스키마에 향만 건축물과 해당 구조물의 속성 정보를 추가하여 제작한 스키마를 통해 제작된 IFC File은 일반적인 상용 뷰어에서 해당 구조물 및 구조물의 정보를 확인할 수가 없었다.

1.4 연구의 범위 및 방법

본 연구는 현재 수행 중인 “생애주기별 향만시설 통합 운영관리를 위한 BIM 기반기술 개발” 연구에서 “향만 IFC 뷰어 개발” 연구의 일부이다. ‘한국건설기술연구원’에서 표준 스키마에 향만 건축물과 해당 구조물의 속성 정보를 추가하여 제작한 스키마와 이를 통해 제작된 IFC File을 사용한다.

본 연구는 향만 구조물의 형상 및 속성 정보를 표현하는 것으로 한정하고 지형의 형상 및 속성 정보 표현은 지원하지 않는다.

본 논문의 연구 방법은 다음과 같다. 첫째, 정적 바인딩 방식 외의 방법으로 IFC의 Entity들을 객체화시킨다. 둘째, 뷰어를 통해 IFC의 구조물 형상 및 속성정보가 표현되는지 상용 및 오픈 소스 뷰어와 비교하여 향만 스키마를 통해 제작된 IFC 확인한다.

2. IFC 스키마와 IFC File에 대한 고찰

2.1 IFC 스키마의 Entity 정의

IFC File은 수십 개에서부터 많게는 수백만 개의 IFC Entity로 구성되어 있다. Entity들은 스키마의 Entity 정의에 의해 서로 참조 관계로 구성되어 있다. Figure 2는 스키마에 정의된 IfcSite의 Entity 정의이다. IfcSite는 RefLatitude, RefLongitude, RefElevation, LandTitleNumber, SiteAddress를 속성으로 갖고 있는데, 이들은 각기 IfcCompoundPlaneAngleMeasure, IfcCompoundPlaneAngleMeasure, IfcLengthMeasure, IfcLabel, IfcPostalAddress에 대응된다.

```
ENTITY IfcSite
SUBTYPE OF(IfcSpatialStructureElement);
RefLatitude : OPTIONAL IfcCompoundPlaneAngleMeasure;
RefLongitude : OPTIONAL IfcCompoundPlaneAngleMeasure;
RefElevation : OPTIONAL IfcLengthMeasure;
LandTitleNumber: OPTIONAL IfcLabel;
SiteAddress : OPTIONAL IfcPostalAddress;
END_ENTITY;
```

Figure 2. IfcSite entity definition

또한 IfcSite Entity는 SUBTYPE OF 뒤 괄호 안의 IfcSpatialStructureElement의 속성 정의를 계승 받는다. Figure 3은 IfcSite의 Entity 계승 구조이다. IfcSite는 자신의 Entity 속성 정의뿐만 아니라 계승 구조상의 상위, IfcSpatialStructureElement부터 IfcRoot까지의 Entity들의 속성 정의를 자신의 속성으로 갖게 된다.

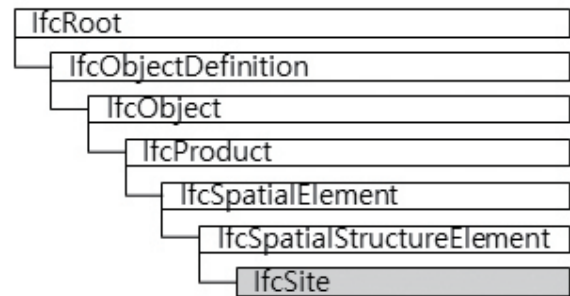


Figure 3. Inheritance diagram of IfcSite entity

IfcSite가 계승 받는 IfcSpatialStructureElement의 스키마에서의 정의된 속성은 CompositionType다. IFC에서의 IfcSite의 속성들 중 CompositionType 속성은 Entity IfcSite의 정의된 속성 앞에 자리하게 된다. 따라서 IFC에서 IfcSite의 속성은 최종적으로 Figure 3에 나온 모든 Entity의 정의된 속성을 갖게 된다. 이처럼 IFC에서 IfcSite의 속성은 스키마에 정의된 Entity의 정의된 속성뿐만 아니라 계승도 상의 상위 Entity의 정의된 속성들로 채워지게 된다.

스키마에 새로운 Entity가 추가되거나 기존의 정의된 Entity에 새로운 속성이 추가되게 된다면 해당 스키마를 활용해 만들어지는 IFC의 구조도 변경되게 된다. IFC를 해석할 때, IFC 제작에 사용된 스키마가 있어야만 정확한 해석이 가능하다. 정적 바인딩 방식의 경우 스키마에 Entity 및 새로운 속성이 추가될 때마다 해당 Entity의 정보를 바인딩할 파일을 Entity의 변경사항과 똑같이 변경시켜줘야 하는 문제점이 발생한다. 따라서 IFC를 제작할 때 사용한 스키마를 활용해 해당 IFC와 스키마 상의 각각의 Entity 속성과 정의를 직접적으로 매칭시키는 방식을 제안한다. 이를 통해 시스템의 변경 없이 스키마 상에 추가된 새로운 항만 구조물의 Entity 및 속성 정보가 있는 IFC File을 출력할 수 있다.

2.2 IFC Entity 매칭

IFC File을 TEXT 데이터 스트림으로 로드한 이후 TEXT 상의 Entity 번호를 Key 값으로 그리고 Entity 번호 뒤에 해당하는 정보를 Value로 두는 Map 자료구조를 생성한다. 이후 해당 IFC를 제작에 사용된 스키마를 같은 방식으로 로드시킨 후 Key로는 Entity 명, Value는 해당 Entity의 정의로 두는 Map 자료구조를 생성한다. 이후 IFC의 Entity 정보를 저장한 자료구조와 스키마 정보를 저장한 자료구조를 같은 Entity Name 맞게 매칭시킨다. 매칭될 경우 IFC의 Value 값에 해당 Entity의 스키마 정의를 추가한다. Figure 4는 매칭 작업 이후 생성되는 IFC Entity 객체 개념도 그림이다. 매칭 이후 해당 객체는 IFC File상의 정보와 스키마 상의 정의를 갖게 된다.

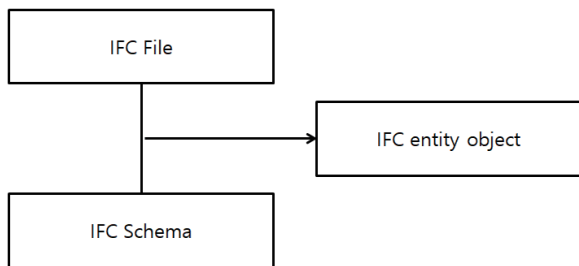


Figure 4. IFC entity object concept

2.2.1 Entity 매칭 모듈

2.2절에서 설명한 방법과 같이 Entity 매칭 모듈은 IFC에 저장된 정보에 스키마의 정의를 추가하였다. 이를 위해 IFC Entity를 저장할 클래스를 만들고 IFC를 TEXT 데이터 스트림으로 라인별로 읽어 들일 때마다 객체화한다. 이후 매칭 모듈에 의해 해당 IFC Entity의 정보를 Parameters에 저장하고, 스키마의 Entity 정의를 ParameterNames라 저장한다. 이 Entity를 참조하는 객체들과 이 Entity가 참조하는 객체들을 ReferencingNumbers와 ReferencedNumbers에 저장한다. Figure 5는 IFC Entity 객체를

구성하고 있는 변수들을 보여준다. IFC와 스키마를 이용하여 만들어진 IFC Entity 객체는 Key 값으로는 IFC 상의 Entity 번호가 들어가고, Value 값으로는 IFC Entity 객체가 들어가는 Map 자료 구조로 메모리에 저장된다.

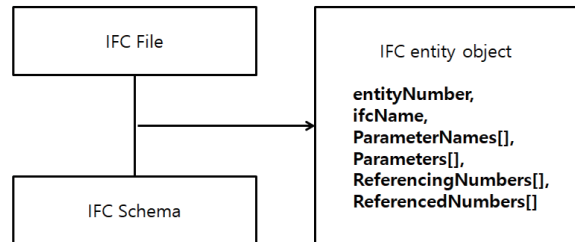


Figure 5. IFC entity object construction

2.2.2 객체 관계 정의 모듈

객체 관계 정의 모듈은 앞 절에서 정의한 Entity 매칭 모듈에 의해 만들어진 객체의 참조 관계를 이용하여 IFC의 구조물 간의 관계를 정의한다. IFCProject부터 시작하는 트리 구조 관계를 표현할 Relation 객체를 Figure 6과 같이 정의한다. 해당 객체는 Entity 번호 레벨, 이전 Entity 번호, 다음 Entity 번호 및 속성 정보 등을 멤버변수로서 갖는다. Figure 6은 Relation 객체를 구성하고 있는 변수들을 보여준다.

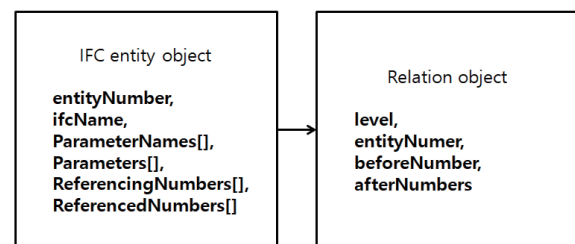


Figure 6. Relation object construction

Entity 매칭 모듈과 객체 관계 정의 모듈을 통해 만들어진 객체들을 이용하여 속성 정보 추출 모듈, 형상 정보 추출 모듈을 만든다. 이를 통해 IFC에서 구조물의 속성 정보와 형상 정보를 추출하여 원하는 형태로 가공할 수 있게 된다.

2.2.3 속성 정보 추출 모듈

속성 정보 추출 모듈은 Entity 매칭 모듈과 객체 관계 정의 모듈에서 만들어진 자료구조들을 참조하여 IFC의 구조물 속성 관련 정보를 저장하는 자료구조를 생성한다. Figure 7은 속성 정보 객체의 멤버 변수를 나타낸 그림이다. 구조물 객체의 이름과 속성 propertySet을 참조하여 객체 번호와 propertySet 안에 있는 property 등을 추가하여 속성 정보 객체를 만든다. 속성 정보 추

출 모듈은 이러한 속성 정보 객체들을 Map에 저장하는 자료구조를 생성한다. 이후 생성된 객체들은 결합 모듈에서 합쳐진다.

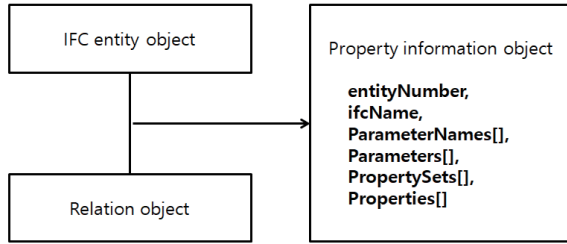


Figure 7. Property information object construction

2.2.4 형상 정보 추출 모듈

구조물 객체인 Relation 객체를 IFC Entity 객체의 Map 자료구조에서 검색한다. 해당 객체의 Entity 속성정보 중 ObjectPlacement 그리고 Representation을 가졌는지 확인한다. 이후 여기서 생성되는 좌표 점들을 모아 Map 형태의 자료구조에 저장한다. 이때의 Key 값은 구조물의 객체번호, Value는 좌표 점들이다. 해당 형상 정보 객체 자료구조는 이후 결합 모듈에서 합쳐진다.

2.2.5 형상 정보 좌표계 변환

구조물은 참조 하는 IfcAxis2Placement3D를 이용하여 위치와 방향을 정한다. IfcAxis2Placement3D Entity 속성 정보를 통해 3차원 공간에서 구조물의 방향 및 위치 전환 정보를 얻을 수 있다.

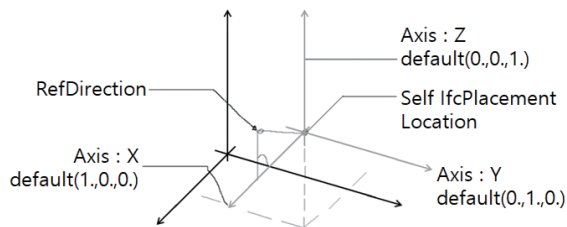


Figure 8. Axis2placement3D concept

Table 3. IfcAxis2Placement3D property information

Attribute	Description
Location	The geometric position of a reference point, such as the center of a circle, of the item to be located.
Axis	The exact direction of the local Z axis.
RefDirection	The direction used to determine the direction of the local X axis.

Table 3은 IfcAxis2Placement3D의 속성정보에 대한 설명이다. 해당 속성정보에선 구조물의 위치정보와 Z축, X축의 방향벡터만을 제시한다.

Z축의 방향벡터 (Z_x, Z_y, Z_z)와 X축의 방향벡터 (X_x, X_y, X_z)를 이용하여 Y축의 방향벡터 (Y_x, Y_y, Y_z)는 식 (1)을 통해 구할 수 있다.

$$\begin{aligned} Y_x &= X_y \times Z_z - X_z \times Z_y \\ Y_y &= X_z \times Z_x - X_x \times Z_z \\ Y_z &= X_x \times Z_y - X_y \times Z_x \end{aligned} \quad (1)$$

구조물은 참조 하는 IfcCartesianTransformationOperator3D Entity 속성정보를 이용해 해당 구조물이 참조하는 좌표 점들의 위치를 바꾸어 준다. IfcCartesianTransformationOperator3D Entity는 3차원 공간에서의 기하 변환을 정의한다. Table 4는 IfcCartesianTransformationOperator3D의 속성 정보이다. 해당 속성 정보에선 x, y, z 축의 방향벡터와 구조물의 기준점, 배율 정보를 알려준다. IfcFile의 구조물의 형상 정보 중 같은 좌표 점을 공유하는 경우가 많다. 구조물의 모양은 같지만, 위와 같은 좌표 변환 Entity에 의해 위치정보와 방향이 달라진다.

Table 4. Ifccartesiantransformationoperator3D property information

Attribute	Description
Axis1	The direction used to determine the derived X axis direction.
Axis2	The direction used to determine the derived Y axis direction.
LocalOrigin	The required translation, specified as a cartesian point. The actual translation included in the transformation is from the geometric origin to the local origin.
Scale	The scaling value specified for the transformation.
Axis3	The exact direction of the derived Z axis direction.

구조물의 좌표 점(pointX, pointY, pointZ)는 아래의 식에 의해 좌표가 변환된다. 이때, S는 Scale 값, X축, Y축, Z축의 방향벡터는 각각 (X_x, X_y, X_z), (Y_x, Y_y, Y_z), (Z_x, Z_y, Z_z)이며, 구조물의 기하학적 기준점은 ($localX, localY, localZ$)이다. 이때 최종적으로 변환된 구조물의 좌표 점(X, Y, Z)는 아래 식 (2)와 같다.

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} localX \\ localY \\ localZ \end{pmatrix} + \left(S \times \begin{pmatrix} X_x & Y_x & Z_x \\ X_y & Y_y & Z_y \\ X_z & Y_z & Z_z \end{pmatrix} \right) \times \begin{pmatrix} pointX \\ pointY \\ pointZ \end{pmatrix} \quad (2)$$

2.2.6 결합 모듈

객체 관계 정의 모듈에서 생성된 자료구조의 Key 값, 객체 번호를 기준으로 다른 모듈에서 만들어진 자료구조 및 정보를 하나로 합친다. 결합 모듈은 Map 자료구조로서 Key로는 객체 번호, Value로는 결합 객체를 저장한다. Figure 9의 결합객체의 멤버 변수는 다른 객체들의 멤버들을 포함하며, IfcCartesianTransfor

mationOperator3D와 IfcAxis2Placement3D의 속성 정보와 식 (1) 과 (2) 를 통해 나온 정보를 transform과 rotation 변수에 담는다.

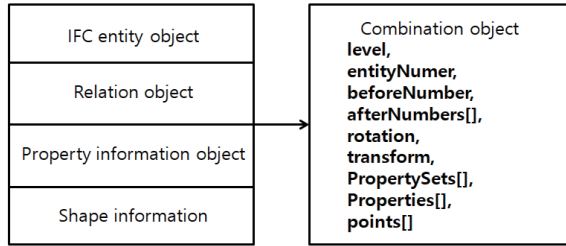


Figure 9. Combination object construction

이후 결합 모듈에 의해 만들어진 객체들을 Json 객체로 변환 및 압축하여 뷰어로 전달한다.

3. IFC 뷰어 구현

3.1 시스템 개발환경

IFC로부터 추출된 정보를 보여주기 위해 본 연구에서 개발한 뷰어의 구성은 Entity 매칭 모듈, 속성 정보 추출 모듈, 형상 정보 추출 모듈, 결합 모듈 및 뷰어 Javascript 등이 있다. Figure 10은 뷰어 시스템의 모듈 간의 데이터의 전달 순서를 그린 그림이다.

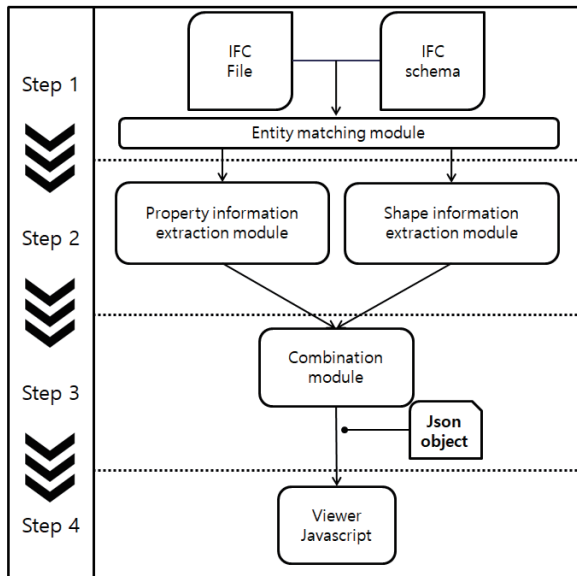


Figure 10. Viewer system flowchart

본 시스템의 주요 개발언어는 Java와 Javascript이며 시스템 개발 환경은 다음과 같다.

- Eclipse v2020-03 (4.15.0)
- Java 1.8
- Apache Tomcat v8.5

뷰어 상의 Javascript는 오픈소스 라이브러리인 JQuery-3.6.0.js와 Web 기반의 그래픽 라이브러리인 Three.js를 활용하여 만들었다.

3.1.1 뷰어 Javascript

Three.js란 오픈소스 Javascript 라이브러리를 이용하여 사용자의 화면상에 IFC의 구조물의 형상을 3차원 그래픽으로 표현한다. 서버 측에선 결합객체들을 Json 객체로 변환하여 사용자 화면의 웹페이지로 전달한다. 이후 사용자 화면의 뷰어 스크립트 모듈은 전달받은 Json 객체를 이용하여 IFC의 구조물을 표현한다. 또한, IFC의 좌표축과 웹 브라우저 환경의 좌표축 표현방식이 다르므로 그것에 맞게 좌표축을 변경한다. 또한, 결합객체 안의 transform과 rotation 정보를 통해 구조물 좌표 점들을 회전 및 이동 시켜 표현한다.

3.2 적용 사례

항만 스키마를 통해 제작된 IFC File을 오픈 소스 및 상용 뷰어를 통해 출력을 확인해 본 결과 제대로 실행되지 않거나 표준 스키마에 정의된 구조물의 형상 정보 및 속성 정보만을 출력된다. 이를 확인하기 위해 항만 구조물 중 하나인 안벽(Quay) IFC File을 만들어 테스트 하였다. 안벽을 구성하고 있는 요소 중 하나인 슬릿케이스를 IfcMarineFacility라 정의하였다. Figure 11과 Figure 12는 상용 IFC 뷰어 중 하나인 Solibri 뷰어와 오픈소스 IFC 뷰어 중 하나인 BIMserver에서 항만 스키마로 제작된 안벽 IFC File을 출력한 결과이다. 각기 오류 메시지가 출력되어 실행이 되지 않았다.

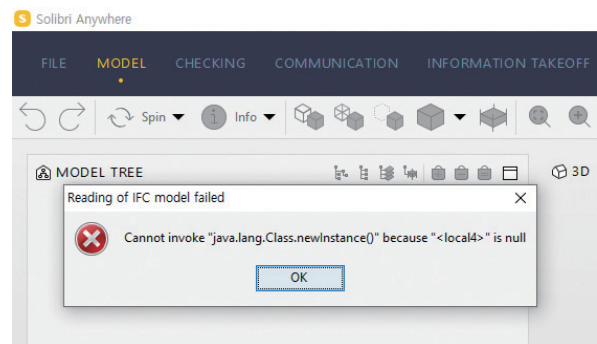


Figure 11. Solibri viewer

Figure 13은 상용 3D 설계, 엔지니어링 및 건설 소프트웨어 벤더 중 하나인 Autodesk에서 제공하는 무료 Viewer이다. 항만 스키마로 제작된 안벽 IFC File을 실행한 결과 슬릿 케이스가 출력되지 않는 것을 확인 할 수 있다.

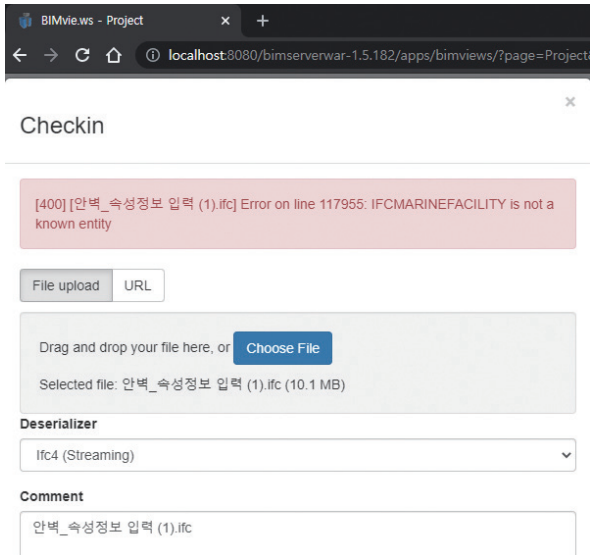


Figure 12. BIMserver viewer



Figure 13. Autodesk Online Viewer

Figure 14와 15는 본 연구에서 개발한 뷰어 화면의 일부이다. 향한 스키마로 제작된 안벽 IFC File을 출력한 결과, Figure 13과 달리 향한 구조물 Entity 중 하나인 IfcMarineFacility의 형상과 속성들이 출력되는 것을 확인 할 수 있다.

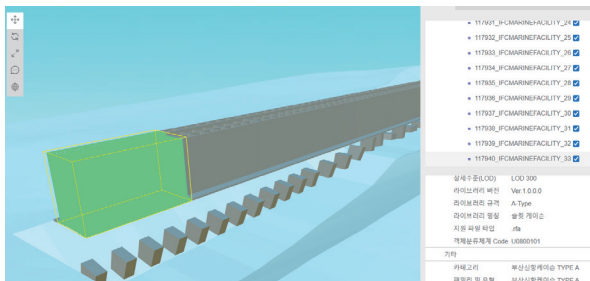


Figure 14. Output of shape and property information

Globalid	14sNwwdVbEGOU_RtQnBZJJ
Name	IfcMarineFacility
Description	0b5e6055-c1a7-4701-ab12-31cd82898ff2-00041f4c
기타	
유형	부산신항케이스 TYPE A
패밀리	부산신항케이스 TYPE A
유형 ID	부산신항케이스 TYPE A
카테고리	부산신항케이스 TYPE A
호스트 ID	부산신항케이스 TYPE A
패밀리 및 유형	부산신항케이스 TYPE A
문자	
상세수준(LOD)	LOD 300
라이브러리 규격	A-Type
지원 소프트웨어	Revit
라이브러리 버전	Ver.1.0.0.0
객체분류체계 Code	U0800101

Figure 15. Output of property information

4. 결론 및 향후 연구 진행 방향

본 연구는 기존의 연구 및 오픈소스 뷰어에서 IFC File을 해석할 때 사용하는 정적 바인딩 방식이 아닌 다른 방법을 사용해 해석하는 방법을 제시하였다. 상용 뷰어 프로그램 및 오픈소스들의 IFC File 해석 모듈들은 대부분 IFC 표준 스키마를 기반으로 정적 바인딩 방식으로 제작되었다. 따라서 이들 프로그램들에선 해석 모듈에 없는 Entity 등에 대응하지 못해 오류가 나거나 구조물을 시각화하지 못하는 것을 확인하였다. 반면 본 연구에서 제작된 시스템을 통해 IFC File의 Entity들과 IFC 스키마의 Entity 정의를 TEXT 스트림으로 메모리에 저장한 후 해당 데이터들을 Entity 객체, Relation 객체, 속성 정보 객체, 형상 정보 등으로 분류하였으며, 최종적으로 분류된 객체들을 결합하여 구조물의 형상 정보와 속성정보를 시각화하여 IFC File의 향한 구조물과 해당 구조물의 속성 정보를 보여주는 것을 확인하였다.

본 연구에서 사용된 방식의 이점은 시스템의 변경 없이 새로운 향한 구조물의 추가 및 변경에 대응할 수 있다는 점이다.

본 연구 중 도출된 문제점은 다음과 같다.

첫째, 모든 IFC Entity를 파악하지 못해 선형으로 표현된 지형 정보를 객체화하지 못하였다.

둘째, IFC 스키마의 전체적인 내용을 파악하지 못해 Entity의 정의만을 활용하였다. Entity 정의 외의 정보들에 대한 분석이 필요할 것으로 판단된다.

셋째, IFC File을 TEXT 스트림으로 메모리에 저장함에 따라 많은 객체 생성으로 인해 발생하는 메모리 공간 점유 문제가 있다. 또한 최종적으로 생성된 객체를 시각화를 위해 뷰어로 전달하는데, IFC File의 크기가 클수록 생성된 객체 역시 커지는 문제가

생긴다. 이를 해결하기 위해 데이터의 압축 및 경량화 및 처리에 관한 후속 연구가 필요해 보인다.

향후 연구에선 이와 같은 문제점들에 대한 고려가 필요할 것으로 판단된다.

감사의 글

이 논문은 2021년 해양수산부 재원으로 해양수산과학기술진흥원의 지원을 받아 수행된 연구임(생애주기별 항만시설 통합 운영 관리를 위한 BIM 기반기술 개발).

References

- apstex, apstex IFC Framework, <https://www.apstex.com/info.html> (May, 13, 2021).
- BIMserver, BIMserver, <https://github.com/opensourceBIM/BIMserver> (May, 13, 2021).
- buildingSMART, Industry Foundation Classes (IFC) – An Introduction, <https://technical.buildingsmart.org/standards/ifc> (May, 13, 2021).
- Cho, G., H., Joo, K., B. (2014). Extension of the IFC Schema for Road Subsidiary Facility. *Journal of the Korea Academia-Industrial cooperation Society*, 15(12), pp. 7385-7392.
- Goh, I., D., Choi, J., H., Kim, E., D., Lee, J., M. (2009). Extracting Building Element Geometry from BIM/IFC Physical Files. *Journal of the computational structural engineering institute of Korea*, 22(2), pp. 163-172.
- Han, J., H., Lee, C., W., Kim, T., R., Nam, S. H. (2016). A development Viewer System for extension of IFC Model Schema geometrylibrary Review. *Proceedings of the Society of CAD/CAM Conference*, pp. 760-761.
- Kim, C., Y., Moon, H., S., Won, J., S., Ju, K., B. (2014). Extension of IFC Core Schema for Earth Work Object. *Proceedings of the Society of CAD/CAM Conference*, pp. 727-730.
- Kim, M. G. (2017). IFC(Industry Foundation Classes), <http://www.ancnews.kr/news/articleView.html?idxno=4887> (May, 2, 2021).
- Park, Y., S. (2012). A Study on Development of the Spatial Network Analysis Tool based on Open BIM Technologies. *Korean Journal of Computational Design and Engineering*, 17(1), pp. 7-16.
- Ryu, J., R., Mun, S., K., Choo, S., Y. (2015). A Study on the Development and Utilization of Indoor Spatial Information Visualization Tool Using the Open BIM based IFC Model. *Journal of Korea spatial information society*, 23(5), pp. 41-52.
- Song, J., K., Choi, W., S., Won, J., S., Kim, N., G. (2013a). A Basic Study for Accumulation of IFC-based Schedule Information. *Journal of Korea Academia-Industrial cooperation Society*, 14(11), pp. 5890-5896.
- Song, J., K., Moon, H., S., Ju, K., B. (2013b). The Method on Utilization of IFC Schema to accumulate the Process Information of BIM for Bridge. *Journal of the Korean Society of Civil Engineers*, pp. 76-79.
- Won, J., S., Shin, J., Y., Moon, H., S., Ju, K. B. (2018). The Development Method of IFC Extension Elements using Work Breakdown Structure in River Fields. *Journal of Korea Academia-Industrial cooperation Society*, 19(4), pp. 77-84.
- Won, J., S., Shin, J., Y., Moon, H., S., Ju, K., B. (2019). Extraction and Verification Method of IFC Schema Elements to Represent BIM for River Facility. *The Journal of Korean Institute of Communications and Information Sciences*, 44(2), pp. 317-325.
- xbim, xbim toolkit, <https://docs.xbim.net/> (May, 13, 2021).