DOI: http://dx.doi.org/10.13161/kibim.2015.5.2.019

BIM 정보 가시화를 위한 오픈소스 뷰어기반 BIM 질의 개발

Open source viewer-based BIM Query Development for BIM information visualization

강태욱[1)]
Kang, Tae Wook[1)]

Received May 12, 2015 / Accepted June 9, 2015

ABSTRACT: The purpose of this study is to design simple BIM query language for visualizing BIM information as the viewpoint of use-cases by using open source viewer cost-effectively. recently, the studies related to the facility management using BIM are being researched world-widely. In this studies, it's important to develop the BIM query to obtain the needed data from BIM database. To implement it, the trends in related studies were reviewed and the open source-based BIM viewer architecture was analyzed. And then the architecture considering the simple BIM Query Language (sBQL) was designed to represent the geometry and property of the BIM database. Lastly, the system by using the proposed sBQL grammar was developed.

KEYWORDS: BIM, Viewer, Visualization, Open source, Query

키워드: BIM, 뷰어, 가시화, 오픈소스, 질의 언어

1. 서론

1.1 연구의 배경 및 목적

최근 세계적으로 BIM을 기반으로 시설물 관리 등을 어플리케이션을 효과적으로 구현하기 위한 연구가 진행되고 있다. 이런 연구에서 사용자가 필요한 BIM 정보를 효과적으로 질의하기 위한 기술의 중요성이 높아지고 있다. 이러한 BIM 정보 질의 기술을 개발하기 위해서는 많은 개발 시간과 비용이 필요하므로 오픈소스를 기반으로 한 BIM질의 기술을 개발하는 방식은 적절한 대안이 될 수 있다.

본 연구는 비용효과적인 BIM질의 방법을 개발하기 위해, 오픈소스뷰어를 기반으로 한 도구, 질의 언어 및 처리 알고리즘을 제안한다. 본 연구의 결과는 BIM기반 시스템, 예를 들어, BIM기반 시설물 관리, 에너지 관리 시스템의 BIM정보 질의 및 가시화에 활용할 수 있다.

1.2 연구의 범위 및 방법

본 연구의 범위는 IFC(Industry Foundation Classes) 기반 BIM 데이터베이스에 대한 질의로 한정한다. 본 연구의 목적이 오픈소스뷰어 기반의 BIM 질의 방식 구현의 효과를 확인하는 것이므로, 질의 문법은 객체 형상과 속성으로만 한정하고, 다중 조건문을 연결하는 'AND', 'OR'문법은 지원하지 않는다.

본 논문의 연구 방법은 다음과 같다.

BIM질의와 관련이 있는 정보교환방식, 상호운용성 및 오픈소스 기반 BIM에 관련된 연구를 2장에서 살펴본다. 이를 바탕으로, 3장에서 표준 모델 및 오픈소스 분석을 통해, BIM질의 언어의 문법을 정의할 때 고려할 사항을 분석하고 정리한다. 4장에서 BIM 질의언어인 sBQL(simple BIM Query Language) 문법과 처리 시퀀스를 정의하고, 5장에서 이를 프로토타입으로 구현하여, 제안된 방식의 효과를 확인한다. 마지막으로, 연구한 내용을 정리하고, 향후 연구 내용을 도출한다.

2. 문헌조사

BIM질의 방식은 정보의 상호운용성과 많은 관련이 있다. 이와 관련된 여러 가지 연구들이 있다. BIM과 시설물 관리, GIS와

[1)]정회원, 한국건설기술연구원, 수석연구원, 공학박사 (laputa99999@gmail.com) (교신저자)

같은 도메인이 다루는 정보 모델 구조는 서로 상이한 부분이 많기 때문에(Hwang, 2012), 사용자가 BIM과 다른 스키마의 정보를 통합적으로 활용하기 위해서는 효과적인 상호운용성 지원 방법을 개발할 필요가 있다(Kang, 2012). BIM서버는 이러한 상호운용 기술 중 정보 모델 관리 등 핵심적인 기능들을 제공한다.

국내 BIM기반 정보교환에 대한 연구로는 ORDB(Object Relationship Database)를 이용한 IFC(Industry Foundation Classes)모델 서버 개발을 통한 효율적인 BIM 데이터 교환에 대한 연구가 있다(Jeong, 2011). 이 연구에서는 데이터 공유 및 교환을 위해 IFC 스키마를 ORDB로 맵핑하여 BIM서버를 개발하였다. BIM 기반 BIM통합정보시스템 프레임워크에 관한 연구(Cho, 2013)에서는 협업이 가능하도록 BIM정보를 활용하기 위해 BIM서버를 고려한 프레임웍을 설계하고 개발하였다. BIM 기반 뷰어 개발 연구(Yoon, 2013)에서는 BIM서버를 기반으로 객체 정보를 추출해 뷰어에 객체를 표출하는 연구를 진행하였다.

해외의 경우 다자간 정보교환을 위한 프레임웍에 대한 연구(Tomo, 2011)에서는 엔지니어링 커뮤니케이션을 위한 프레임웍을 정의하고 BIM 정보 관리 및 교환을 위해 BIM 서버를 활용하고 있다. BIM기반 서비스 지향 아키텍처에 관한 연구(Umit, 2012)에서는 소프트웨어 공학의 SaaS(Software as a service) 개념과 디자인 패턴(Design pattern)적용한 아키텍처를 제안하고 이를 BIM서버로 구현하였다. 타 기종 시스템과 BIM정보를 통합하는 연구(EI, 2013)에서는 BIM과 GIS 모델링 소프트웨어 도구를 리뷰하고 각 도구의 포맷 변환 기능을 이용해 BIM 표준 파일 포맷인 IFC를 해당 시스템의 데이터 포맷으로 변경해, BIM 객체를 가시화하는 연구를 수행하였다.

앞서 언급한 대부분의 연구는 이기종 시스템 간의 정보교환 및 상호운용성에 관한 연구들이다. 몇몇 오픈소스 기반 BIM 기술이 연구가 있으나, 오픈소스기반 BIM질의 언어 개발과 관련된 연구는 매우 부족한 상황이다. 본 연구에서는 비용효과적인 BIM 질의 방법을 개발하기 위해, 오픈소스뷰어를 기반으로 한 도구, 질의 언어 및 처리 알고리즘을 제안한다.

3. 표준 모델 및 오픈소스 분석

3.1 표준 모델 IFC 분석

BIM질의언어의 문법을 정의하기 위해서는, IFC객체 구조를 파악하여, 이를 통해, 문법의 기본요소를 정의하고, 질의 시 정보탐색을 수행할 수 있어야 한다.

IFC에서는 객체간의 포함, 연관, 할당, 제약조건, 기하학적 관계 등 각 객체별 특성에 따른 관계들을 모두 명확히 표현하고 있다. 그래서 IfcObjectDefinition이란 개념을 두어서 이 관계들을 특별히 관리하고 있다. 특히, EXPRESS 언어는 INVERSE관계를 표현할 수 있어, 예를 들면 자동차가 엔진을 Aggregation 하고 있지만, 엔진관점에서도 자동차에 포함되어지는 관계를 역으로 표현할 수 있다.

객체 특성은 IfcPropertyDefinition 으로 기술되며 객체간 관계는 할당(Assign), 집합(Aggregation), 분해(Aggregation의 역관계), 정의(Defined), 연결과 같은 IfcRelationship에 의해 기술될 수 있다. IFC에서는 관계도 객체로써 취급된다.

IFC에서 건물이란 전체(Whole), 건물 공간, 건물 부재 간에 집합 관계로 구성된다. 이는 전체와 부분 관계(Whole/part relationship)을 의미하는 것으로 IfcRelAggregates는 이를 지원한다. 관계는 항상 두 객체 간에 발생되는 것이므로 집합체와 부분으로 관계가 정의된다. 집합체는 IfcObjectDefinition형식의 RelatingObject로 정의되며 부분들은 RelatedObjects 로 정의된다. 예를 들어 집합체는 건물이고 부분들은 건물의 각 층들이 될 수 있다.

IFC공간적 구조는 대지(IfcSite), 건물(IfcBuilding), 층(IfcBuilding-Story), 공간(IfcSpace) 4가지 개념으로 구분된다. IfcBuilding은 Figure 1과 같이 IfcBuildingStorey를 포함하고 지형 높이(ElevationOfTerriain)과 같은 속성을 가지고 있다.

IfcBuildingStorey는 벽체(IfcWall), 슬라브(IfcSlab), 기둥(IfcColumn)과 같은 건축 부재를 포함할 수 있는 데 이런 건축 부재는 IfcProduct로 일반화된 클래스로 표현된다. 공간과 건축 부재 객체들은 서로 관계를 가지고 있으며 예를 들어 다음 그림과 같이 IfcBuilding는 IfcBuildingStorey를 IfcRelAggregates객체를 이용해 집합관계로 포함하고 있다. 반대로 IfcBuildingStorey는 자기를 포함하는 객체를 역관계(INV)로 가리키고 있다. IfcBuildingStorey는 IfcProduct를 IfcRelContainedInSpatialStructure 관계객체를 이용해 관리한다.

Figure 2는 객체관계를 좀 더 자세히 표현한 다이어그램이다. IfcRelAggregates 관계객체를 이용해 Relating 객체와 Related 객체들의 정보를 관리하고 있다. IFC는 IfcProject에서 IfcBuildingStorey에 이르기까지 연결된 위상정보를 제공하며 이를 통해 계층적으

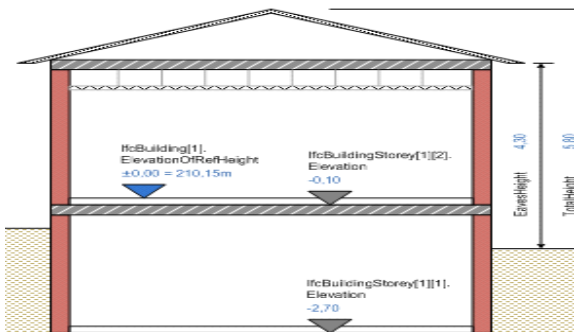

Figure 1 IfcBuilding concept diagram (buildingSMART)

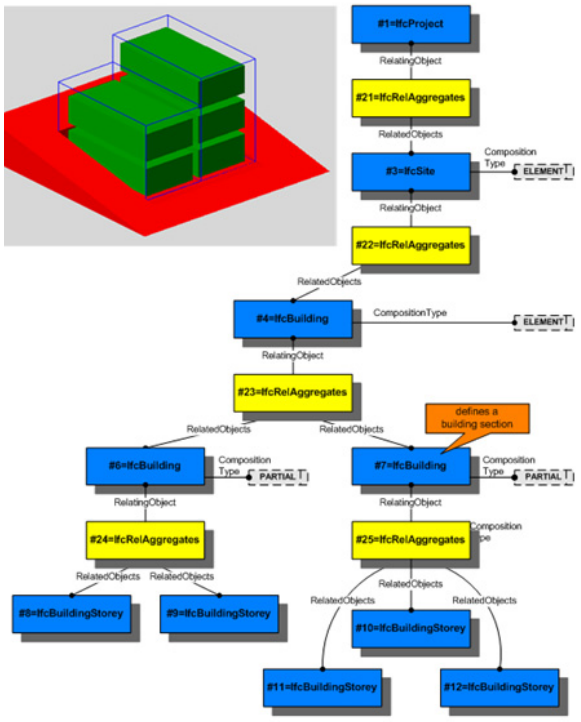

Figure 2 IfcRelAggregates relationship (buildingSMART)

로 서로 관련된 정보를 탐색할 수 있다.

각 공간객체는 자신만의 지역좌표계를 가지고 있는 데 이는 IfcLocalPlacement에 의해 표현된다. IfcLocalPlacement는 X, Y, Z축과 Orientation을 가질 수 있어 IfcLocalPlacement에 의해 정의된 공간 객체는 이 지역좌표계에 영향을 받는다. 즉, 지역좌표계가 (999, 210, 3)이라면 이 지역좌표계에 속해진 부재 객체들의 형상 좌표들은 이를 원점으로 취급되 상대좌표로 정의된다. 이는 컴퓨터 그래픽스에서 전통적으로 사용되는 좌표계 스택 프레임(Stack Frame)개념을 활용하기 위함이다. 참고로, 대부분의 컴퓨터 그래픽 좌표계는 전역좌표계와 지역좌표계들로 나뉘며 지역좌표계에 속해진 객체들은 좌표변환행렬을 통해 전역좌표계로 변환된다. 이를 통해 각 객체별로 효과적인 좌표 처리를 할 수 있다.

3.2 오픈소스 조사

BIM정보 질의는 사용자가 원하는 정보를 획득하기 위해, 미리 정해진 질의 문법을 이용하여, 질의내용을 기술하고, 이 내용을 BIM데이터베이스 서버에서 추출하는 과정 전부를 수행해야 한다. 이 장에서는 이와 관련된 오픈소스를 조사하고, 정리한다.

bimServer(Jakob, 2010)는 다중 사용자간 형상관리를 지원하는 오픈소스 기반 BIM 모델 협업 서버이다. 기본적으로 버클리 파일 DB(Database)상에서 동작하므로 일반적인 RDBMS(Relational

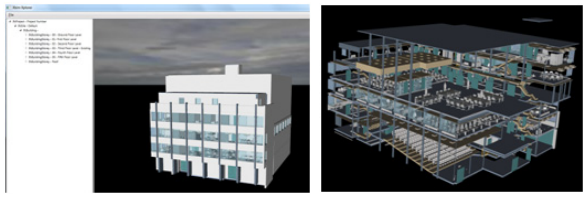

Figure 3 xBIM Xplorer screen image

Database Management System)에 비해 BIM파일을 등록하거나 인출하는 등의 연산속도가 빠르고 다양한 응용 도구와 연계를 할 수 있도록 웹서비스(Web service)를 지원한다. 이런 웹서비스를 이용해 다양한 방식의 BIM 정보쿼리를 포함한 응용 도구를 개발할 수 있다. 다양한 모델러 및 시스템과 연동을 위해 약 470개 (1.1 버전)의 Web-service를 지원하고 Map 기능 구현을 위해 Google 등 클라우드 기반 시스템을 활용하고 있다.

오픈소스기반 BIM뷰어로 널리 알려진 xBIM(eXtensible Building Information Modeling) Toolkit은 오픈소스 기반 소프트웨어 개발 BIM 도구이며 IFC를 지원한다. xBIM은 IFC포맷을 읽고 쓰는 기능을 지원하며 형상 표현을 위한 메쉬 프로세싱, 위상 정보 연산 처리 등을 지원한다. xBIM은 IFC 기반 어플리케이션 개발 시 IFC 처리 엔진으로써 동작될 수 있다.

xBIM은 C#으로 개발되어 있어 기능을 확장하기가 용이하며 객체지향적으로 구조화되어 있다. 3D 모델 가시화는 솔리드 모델링 엔진 오픈소스인 Open CASCADE 를 이용해 3D 가시화를 처리하고 IFC 파서는 기존의 IFC파서 대신에 Gardens Point Parser Generator 를 이용한 파서를 활용하고 있다. 이 파서는 다른 오픈스스 기반의 파서와는 다르게 처리속도가 빠른 장점이 있다. 그래서 다른 오픈소스기반 뷰어와는 다르게 IFC파일 로딩 속도가 빠르다.

xBIM은 BIM모델을 읽고 생성하는 기능과 더불어 그래픽 가시화처리 기능을 포함하고 있다. xBIM을 기반으로 제공되는 Xplorer는 IFC 파일 보기, IFC 객체 속성 확인, 객체 가시화 처리, ifcXML, ifcZIP파일 출력 등을 지원한다.

3.3 오픈소스 분석

본 장에서는 xBIM을 이용한 실습을 위해 필요한 핵심 구조를 설명한다. xBIM 컴포넌트 구조는 다음과 같다.

xBIM은 핵심적인 컴포넌트가 4개가 있는 데 Xbim.Ifc, Xbim.IO, Xbim.ModelGeometry.Scene, Xbim.Presentation이다.

XbimXplorer BIM뷰어는 앞서 설명한 컴포넌트를 모두 사용해 개발된 것이다.

BIM뷰어의 로직 처리 구조를 살펴보기 위해 xBIM 프로젝트의 소스들 중 "XplorerMainWindow.xaml.cs" 소스파일을 다음과

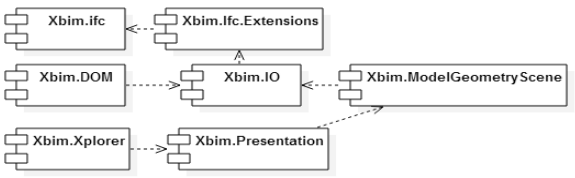

Figure 4 xBIM component diagram (UML)

Table 1 xBIM component description

Component	Description
Xbim.Ifc	IFC 객체들을 관리한다. IFC의 클래스인 IfcProduct 등을 구현하고 있다.
Xbim.IO	FC파일과 입출력을 위한 기능을 제공한다. XbimModelServer 등의 클래스를 구현하고 있다.
Xbim.ModelGeometry.Scene	IFC 형상정보를 가시화하여 그래픽 장면(Scene)을 만들기 위한 기능이 구현되어 있다. 주로 메쉬처리와 좌표변환을 담당한다.
Xbim.Presentation	윈도우에서 3D를 가시화할 수 있도록 DrawingView3D 클래스를 제공하고, 3D모델 배경이 되는 Skybox와 모델 탐색을 위한 Pan 등 기능을 지원한다. 또한, 이 도면 뷰에서 마우스 클릭 등 이벤트가 발생하였을 때 선택된 객체를 탐색해 리턴해 주는 기능을 제공한다.

같이 분석해 보았다. 이 소스는 OpenIfcFile이 정의되어 있으며, 어떻게 IFC파일을 파싱(parsing)하고, BIM가시화와 관련된 인터페이스를 설정하는 지가 구현되어 있다.

```
void OpenIfcFile(object s,
        DoWorkEventArgs args)
{
  BackgroundWorker  worker = s
      as BackgroundWorker;
  string ifcFilename = args.Argument
      as  string;

  IModel model =
      new bimFileModelServer();
      model.Open(ifcFilename,
      worker.ReportProgress);
  XbimScene geomEngine =
      new XbimScene(model);
  ModelProvider.Scene =
      geomEngine;
}
```

OpenIfcFile 함수 정의.

IFC로드 시 처리 진행 표시를 위한 백그라운드 작업 생성.
XbimFileModelServer 객체 생성.
모델서버 객체를 로드할
IFC이름을 받아 Open 함.
모델서버에 저장된 BIM정보를 화면에 표시하기 위한
XbimScene 객체 생성.

모델 정보를 제공하는 객체에 XbimScene객체를 할당.

앞의 코드를 분석해 보면 다음과 같은 방식으로 동작함을 알 수 있다. 우선, IFC로드를 위해서는 IFC문법에 맞게 Text를 파싱(Parsing)해야 한다. XbimFileModelServer는 IfcInputStream객체를 이용해 IFC파일 TEXT 데이터 스트림을 로드한 후 파서를 이용해 IFC문법을 체크하고 IFC정보를 읽어들인다.

xBIM은 앞에서 언급한 XbimFileModelServer에서 Open함수를 이용해 IFC를 로드한다. XbimFileModelServer는 IFC파일을 Open하거나 IFC를 출력하는 ExportIfc를 제공하며 Open함수로 로드된 객체들을 IfcProject 를 루트(Root)객체로 관리하고 있다. 이 IfcProject객체를 이용해 모든 IFC객체 계층 구조를 탐색하고 필요한 정보를 추출할 수 있다. 각 객체는 IFC 구조에서 다루었듯이 IfcObjectDefinition의 IfcRelationship 객체들을 통해 탐색할 수 있도록 되어 있다.

다음은 이러한 과정들을 표현한 클래스 다이어그램이다.

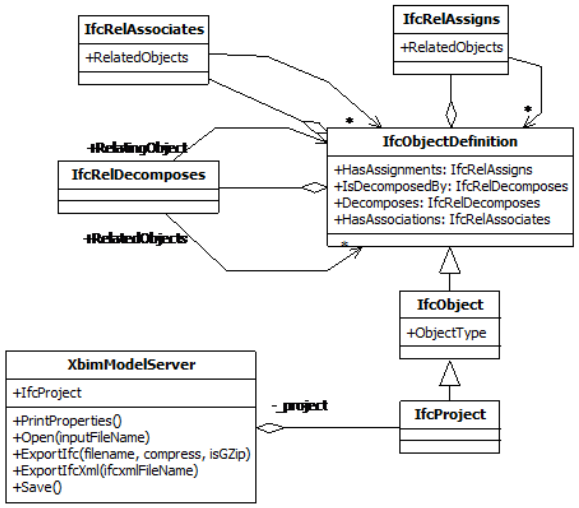

Figure 5 Relationship between XBimModelServer and IFC (UML)

앞의 객체 관계도를 참고해 XbimModelServer 객체에 저장된 IfcSite에 접근하는 코드는 다음과 같다. IFC객체의 관계도와 비교해 보면 코드가 이를 반영해 관련 정보를 탐색하였음을 알 수 있다.

```
private     string  GetIfcSiteName(IModel
                                   model)
{
   string    SiteName = "";
   foreach (var relation
   in model.IfcProject.IsDecomposedBy)
   {
      foreach(IfcObject obj
          in relation.RelatedObjects)
      {
         SiteName = obj.Name;
         break;
      }
   }
   return SiteName;
}
```

XbimFileModelServer객체의 인터페이스를 받아 IfcSite 이름을 얻는 함수.

IfcProject의 IsDecomposedBy 관계들을 획득.
획득한 관계 객체의 RelatedObjects에 포함된 객체들을 얻음.
Whole/Part에서 Part에 해당함.
Part에 해당하는 객체를 얻은 후 이름을 리턴. IfcProject의 경우 Part 객체는 IfcSite임.

얻은 객체 이름을 리턴함.

지금까지 xBIM의 구조를 살펴보았다. 이런 이해를 바탕으로 오픈소스뷰어 기반 BIM질의 언어를 개발해 본다.

4. 질의 언어 디자인

4.1 질의 언어 문법 정의

특정 객체를 탐색할 쿼리 문자열을 입력받아 탐색된 객체들의 유형만 화면에 표시하는 예제를 만들어 본다. 이를 위해서는 IFC 구조에 대한 이해가 필수적이다. 정보 추출을 위한 쿼리를 정의하기 위해 간단한 문법을 가진 sBQL 언어를 정의한다. 이

연구에서는 좀 더 정규화된 방식의 질의 언어를 개발할 때 이용하는 LEX(LEXical analyzer)나 YACC(Yet Another Compiler Compiler)같은 토큰 분석기와 파서 생성기를 사용한다.

이런 방식은 언어의 문법성을 정규화함으로써, 문법 오류 등을 파싱과정을 통해 체크할 수 있다. 다음은 LEX/YACC 로 정의된 sBQL 문법이다.

```
%token PROPERTY_NAME
%token STRING
%%

Variable
    : PROPERTY_NAME

Value
    : STRING

AssignOP
    : '='

SimpleExpression
: Variables AssignOP Value

LogicOp
: '&'

Expression
: SimpleExpression
| SimpleExpression LogicOp Expression
%%
```

이러한 문법의 결과로 아래와 같은 질의문을 문법적으로 정의하는 것이 가능하다.

```
sBQL Query: Type=IfcWindow&Name=Window xyz
```

이 예의 의미는 BIM객체의 속성명이 Type이고 값이 IfcWindow인 객체와 그 객체의 Name 속성값이 Window xyz 인 조건에 부합하는 모든 객체를 추출해 리턴하라는 뜻이다. 이런 방식으로 BIM데이터베이스 내 객체를 효과적을 추출해 낼 수 있다.

4.2 sBIMql 처리 알고리즘 디자인

알고리즘 구조는 크게 세 부분으로 구분된다.

첫번째 단계는 입력받은 sBQL 쿼리 문자열에서 문자열 연산을 이용해 토큰들을 분리하는 단계이다.

두번째 단계는 BIM객체를 입력 받아 PSet 속성 리스트를 얻고 sBQL 토큰들에 포함된 속성 조건에 맞는 지 체크하는 단계이다.

세번째 단계는 sBQL 쿼리문 조건에 부합된 객체 리스트를 리턴받아 화면에 표시해 주는 단계이다. 시퀀스 구조도는 다음과 같다.

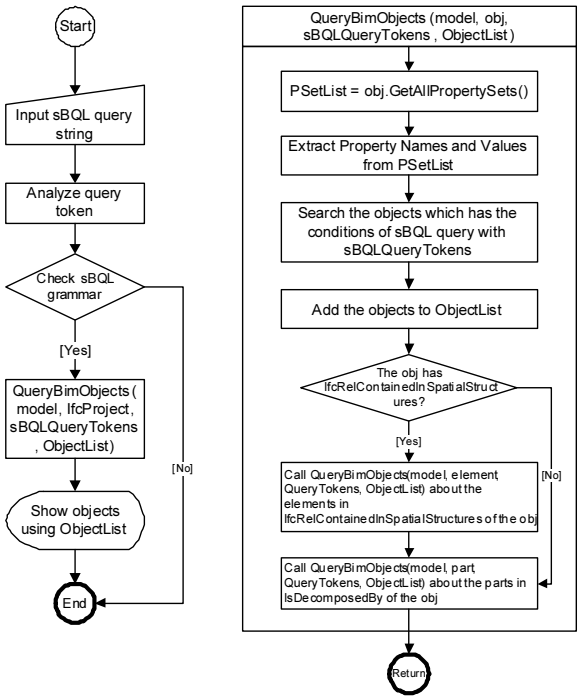

Figure 6 sBQL algorithm

5. 프로토타입 개발 및 효과 확인

이 장에서는 오픈소스뷰어 기반 BIM 질의언어의 효과를 확인해 보기 위해, 프로토타입을 개발한다. 이를 위해, c#언어를 이용해 프로토타입을 개발한다.

Figure 6에 기술된 알고리즘을 고려해, xBIM 엔진을 이용해 c#에서 구현하기 위해서는 몇가지 절차가 필요하다.

첫번째로 xBIM 엔진에서 필요한 라이브러리를 사용한다고 "using"문을 이용해 선언해야 한다. 이렇게 해야 라이브러리 안에 구현되어 있는 객체와 기능을 사용할 수 있다.

두번째로 sBQL 을 입력받는 입력 UI를 만들어야 한다. 이 부분은 본 책의 주제와 큰 관계가 없는 것이므로 자세히 설명하지 않는다. 이미 제공되는 소스에 InputQueryStringBox 함수로 구현해 놓았으므로 사용만 하면 된다.

세번째로 xBIM에서 IFC 모델을 관리하는 IModel을 이용하기 위해 ModelProvider.Scene.Graph.Model 을 통해 IModel 객체

를 획득해야 한다. sBQL 쿼리 결과 얻은 BIM객체들만 화면에 표시하기 위해 DrawingControl 객체의 Hide, Show 함수를 호출한다.

이를 고려해, 개발된 프로토타입을 실행하여, SimpleWallWindow.ifc 파일을 로드하고, BIM 질의 메뉴를 실행하여 다음의 sBQL 쿼리문을 수행해 보았다.

"FireExit=False&SmokeStop=False"

BIM 질의 수행 결과는 다음과 같이 "FireExit"속성값이 "False"이며 "SmokeStop"속성값이 "False"인 BIM객체가 쿼리되어 해당 객체만 화면에 표시된다.

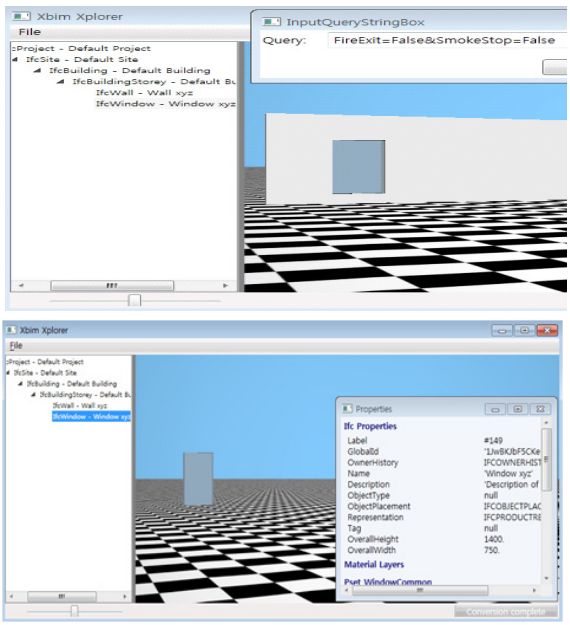

Figure 7 BIM query result (top: before query, bottom: after query)

제안된 방식의 의미를 확인해 보면 다음과 같다.

첫 번째, 보통, BIM으로 모델링된 건물은 LOD(Level Of Detail)에 따라 부재가 수십만개에서 수백만개 이상 있을 수 있다. 이를 간단한 질의언어로 검색할 수 있고, 이런 질의언어를 기반으로 시스템을 개발해 나간다면, 특정 언어에 종속된 코딩으로 구현된 시스템 보다, 시스템의 유연성과 유지관리성이 개선될 것이다.

두 번째, 앞서 sBQL을 개발한 과정을 보면, 개발 시 많은 부분을 오픈소스뷰어를 이용함으로써, 개발비용을 줄이고 있는 것을 확인할 수 있다. 예를 들어, BIM객체 렌더링 엔진, BIM뷰어 인터페이스, BIM객체 브라우저 등은 BIM객체의 정보검색을 위해 필요한 것들이다. 직접 개발하기 위해서는 많은 비용이 소모되지만, 오픈소스뷰어를 기반으로 개발함으로써, BIM 질의언어를 정의하고, 구현하는 것에만 초점을 맞출 수 있다. 이는 BIM기반 시설물 관리, 에너지 관리 등의 시스템 구현 시 개발 노력을 효과적으로 줄일 수 있다는 것을 의미한다.

6. 향후 연구 진행 방향 및 결론

본 연구는 비용효과적인 BIM질의 방법을 개발하기 위해, 오픈소스뷰어를 기반으로 한 도구, 질의 언어 및 처리 알고리즘을 제안하였다. 단, 본 연구의 범위는 IFC(Industry Foundation Classes) 기반 BIM 데이터베이스에 대한 질의로 한정하였으며, 질의 문법은 객체 형상과 속성으로만 한정하고, 다중 조건문은 지원하지 않는다.

연구는 BIM질의와 관련이 있는 정보교환방식, 상호운용성 및 오픈소스 기반 BIM에 관련된 연구를 살펴보고, 이를 바탕으로, 표준 모델 및 오픈소스 분석을 통해, BIM질의 언어의 문법을 정의할 때 고려할 사항을 분석하고 정리하였다. 그리고, BIM 질의언어인 sBQL(simple BIM Query Language) 문법과 처리 시퀀스를 정의한 후, 프로토타입으로 구현하여, 제안된 방식의 효과를 확인해 보았다.

향후, 공간 겹침, 공간 내 포함 등과 같은 다양한 공간질의를 고려해, BIM질의언어를 확장할 계획이다.

감사의 글

본 연구는 한국건설기술연구원 2015년 주요시드사업(기존 시설물의 효율적인 유지관리를 위한 건축 MEP 객체 역설계 기술 개발)의 연구비지원에 의해 수행되었음.

References

buildingSMART, "IFC2x Editioin 4 alpha version"

Cho, J. K, Kim Y. S, Mha Y. K (2013), "A Framework of Server-Based BIM System for Architectural Design firm", Korea Institute BIM Society, Vol. 3, No. 2, pp. 10-19.

El, M. R, Rezoug, M, Hijazi, Ihab (2013), "Integrating and Managing BIM in GIS, Software Review", International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 2, No. 2, pp. 31-34.

Hwang, J. R, Kang, T. W, Hong, C. H, "A Study on The Correlation Analysis Between IFC and CityGML for Efficient Utilization of Construction Data and GIS Data", Korea Spatial Information Society, Vol. 20, No. 5, pp.

49–56.

Jakob, B, Leon, V. B, Ruben, D. L, Pim, D. H, (2010), “An open source IFC Model Server”, Proceedings of the CIB, Vol. 27, No. 78, pp. 1–8.

Jeong, J. Y, (2011), “A study of efficient BIM data exchange and sharing”, Yonsei University.

Kang, T. W, Hong, C. H, Hwang, J. R, Choi, H. S 2012, “The External BIM Reference Model Suggestion for Inter–operability Between BIM and GIS”, Korea Spatial Information Society, Vol. 20, No. 5, pp. 91–98.

Lee, G, Jeong, J, Won, J, Cho, C, You, S, Ham, S, Kang, H (2014), “Query Performance of the IFC Model Server Using an Object–Relational Database Approach and a Traditional Relational Database Approach”, Journal of Computing in Civil Engineering, Vol. 28, No. 2, pp. 210–222.

Tomo C (2011), “A review and outlook for a Building Information Model(BIM): A multi– standpoint framework for technological development”, Advanced Engineering Informatics, Vol. 25, No. 2, pp. 224–244.

Umit I (2012), "Design patterns for BIM–based service–oriented architectures", Automation in Construction, Vol. 25, pp. 59–71.

Yoon, S. W, Kim, B. G, Choi, J. M, Gwon, S, W (2013), “A Prototype BIM Server based viewer for Cloud Computing BIM Services”, Korea Society of Civil Engineers, Vol. 33, No. 4, pp. 1719–1730.