

오픈 소스 BIM서버를 활용한 비용 효과적인 GIS기반 BIM 객체 가시화 시스템 아키텍처 설계

GIS-based BIM Object Visualization System Architecture Design using Open source BIM Server Cost-Effectively

강태욱* · 홍창희**

Tae Wook Kang · Chang Hee Hong

요약 본 연구의 목적은 오픈소스 BIM서버를 활용해 GIS(Geographic Information System)기반으로 BIM(Building Information Modeling)객체를 가시화하는 비용 효과적인 시스템 아키텍처를 디자인하는 것이다. 최근 세계적으로 GIS와 BIM을 기반으로 시설물 관리 등을 효과적으로 구현하기 위한 연구가 진행되고 있다. 관련 연구에서 GIS와 같은 이기종 시스템과 연계가 가능한 BIM서버가 개발될 필요성이 부각되고 있다. 이런 관점에서 오픈소스 BIM서버는 비용 및 유지보수성 측면에서 효과적인 대안이 될 수 있다. 본 연구는 오픈소스 BIM서버를 활용한 GIS기반 BIM객체 가시화 시스템의 아키텍처를 디자인한다. 이를 구현하기 위해 BIM서버 관련 연구 동향을 분석하고, 최근 해외에서 활용되기 시작한 오픈소스 기반 BIM서버인 bimServer의 구조를 분석한다. BIM서버에 저장된 BIM객체의 데이터인 형상 및 속성을 GIS 상에 가시화할 수 있는 방안을 도출하고 아키텍처를 제안한다. 마지막으로 제안된 아키텍처의 적용성을 보이기 위해 시스템을 구현하고 결과를 도출한다.

키워드 : BIM, GIS, 오픈 소스, BIM Server, 객체 가시화

Abstract The purpose of this study is to design the GIS (Geographic Information System)-based BIM (Building Information Modeling) Objects Visualization System (OVS) Architecture by using Open source BIM Server Cost-Effectively. Recently, the studies related to the facility management using GIS and BIM are being researched world-widely. In this studies, it's important to develop the BIM Server to link the heterogeneous system such as GIS and represent GIS-based BIM objects. To implement it, the trends in related studies were reviewed and the open source-based BIM server architecture was analyzed. And then the architecture considering the BIM server was designed to represent the geometry and property of the BIM object which was stored in BIM server. Lastly, the system by using the proposed architecture was developed.

Keywords : BIM, GIS, Open source, BIM Server, Object visualization

1. 서 론

최근 세계적으로 GIS와 BIM을 기반으로 시설물 관리 등을 효과적으로 구현하기 위한 연구가 진행되고 있다. 이런 연구에서 GIS에서 BIM객체를 표현하기 위해 GIS와 같은 이기종 시스템과 연동을 할 수 있는 BIM서버 개발이 필요해 지고 있다.

BIM서버 개발을 위해서는 몇 가지 필수적인 요구 사항을 만족해야 한다. BIM서버는 BIM객체를 보관 및 관리하고 클라이언트가 필요한 정보를 질의할 수

있는 기능을 포함해야 한다. 또한, 다중 사용자가 접속해 BIM객체에 대한 CRUD(Create, Read, Update, Delete)연산을 수행하였을 경우 Checkin, Checkout등을 통한 형상관리(Configuration Management)와 단위 작업의 트랜잭션(Transaction)이 처리되어야 한다. 또한, GIS에서 수많은 BIM객체를 관리하고 표현하기 위해서는 적절한 질의 성능을 가지고 있어야 한다.

이러한 BIM서버를 개발하기 위해서는 많은 개발 시간과 비용이 필요하므로 오픈소스를 기반으로 한 BIM서버는 적절한 대안이 될 수 있다.

† This work was researched by the supporting (2014) Development of BIM/GIS Interoperability Open-Platform

* Tae Wook Kang, Senior Researcher, ICT Lab, Korea Institute of Construction Technology, ktw@kict.re.kr

** Chang Hee Hong, Senior Researcher, ICT Lab, Korea Institute of Construction Technology, chhong@kict.re.kr

bimServer[4]는 다중 사용자간 형상관리를 지원하는 오픈소스 기반 BIM 모델 협업 서버이다. 기본적으로 버클리 파일 DB(Database)상에서 동작하므로 일반적인 RDBMS(Relational 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.

본 연구 목적은 오픈소스 기반 BIM서버를 이기종 GIS 클라이언트 시스템과 연계해 GIS기반 BIM객체 가시화 시스템을 개발하는 것이다. 본 연구에서는 최근 해외에서 활용되기 시작한 오픈소스 기반 BIM서버인 bimServer의 구조를 분석한 후 GIS와 연계하기 위한 아키텍처를 설계한다. 아키텍처의 적용성을 확인하기 위해 GIS기반에 BIM객체를 가시화하는 시스템을 개발한다. 이를 통해 제안된 방법이 BIM객체 가시화를 위해 BIM서버가 비용 효과적으로 구현되어 활용될 수 있음을 보인다.

본 논문의 연구 방법은 다음과 같다.

2장에서 BIM객체 가시화를 위한 BIM서버와 GIS연계에 관한 최신 연구동향을 살펴본다. 이 후 3장에서는 오픈소스 기반 BIM서버인 bimServer의 구조를 컴포넌트, 웹서비스, 프로젝트 관리 방법, 파일 호환성, 좌표계 설정 방법, 정보 질의 방법을 분석한다. 이를 통해 4장에서 GIS기반 BIM객체 가시화를 위한 BIM서버 연계 아키텍처를 설계한다. 설계된 아키텍처의 적용성을 확인하기 위해 5장에서 GIS기반에 BIM객체를 가시화하는 시스템을 개발하여 BIM객체가 BIM서버를 통해 GIS와 효과적으로 가시화되고 통합될 수 있음을 보인다. Figure 1은 연구 흐름도이다.

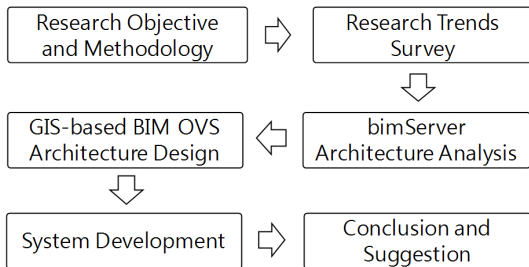


Figure 1. Research flow

2. 관련 연구 동향

BIM과 GIS이 다루는 정보 모델 구조는 서로 상이한 부분이 많기 때문에[3], 사용자가 BIM과 GIS 정보를 통합적으로 활용하기 위해서는 효과적인 상호운용성 지원 방법을 개발할 필요가 있다[6]. BIM서버는 이러한 상호운용 기술 중 정보 모델 관리 등 핵심적인 기능들을 제공한다.

국내 BIM서버에 관련된 연구로는 ORDB(Object Relationship Database)를 이용한 IFC(Industry Foundation Classes)모델 서버 개발을 통한 효율적인 BIM 데이터 교환에 대한 연구가 있다[5]. 이 연구에서는 데이터 공유 및 교환을 위해 IFC 스키마를 ORDB로 맵핑하여 BIM서버를 개발하였다. BIM서버 기반 BIM통합정보 시스템 프레임워크에 관한 연구[1]에서는 협업이 가능하도록 BIM정보를 활용하기 위해 BIM서버를 고려한 프레임워크를 설계하고 개발하였다. BIM서버 기반 뷰어 개발 연구[10]에서는 BIM서버를 기반으로 객체 정보를 추출해 뷰어에 객체를 표출하는 연구를 진행하였다.

해외의 경우 다자간 협업 및 데이터 교환을 위한 프레임워크에 대한 연구[8]에서는 엔지니어링 커뮤니티 이션을 위한 프레임워크를 정의하고 BIM 정보 관리 및 교환을 위해 BIM 서버를 활용하고 있다. BIM기반 서비스 지향 아키텍처에 관한 연구[9]에서는 소프트웨어 공학의 SaaS(Software as a service) 개념과 디자인 패턴(Design pattern)적용한 아키텍처를 제안하고 이를 BIM서버로 구현하였다. GIS상에서 BIM정보를 통합하는 연구[2]에서는 BIM과 GIS 모델링 소프트웨어 도구를 리뷰하고 각 도구의 포맷 변환 기능을 이용해 BIM 표준파일 포맷인 IFC를 ArcGIS 포맷으로 변환하여 ArcGIS상에 BIM객체를 가시화하는 연구를 수행하였다.

앞서 언급한 대부분의 연구는 단일 건물에 대해 BIM서버를 이용하여 BIM객체 데이터를 교환하고 뷰어를 통해 가시화하는 데 초점이 맞춰져 있다.

최근에 GIS를 기반으로 BIM객체를 가시화하는 연구가 진행되고 있으나 아직은 특정 상용 소프트웨어 도구에 의존된 파일 변환 방식으로 진행되고 있는 경우가 대부분이다.

몇몇 연구에서는 다음 표와 같은 상용 BIM서버를 사용하고 있으며, 오픈소스 개발 방식과는 비용적인 부분 등 여러 가지 차이점이 있다. 여기서 BIM서버라 함은 자체적으로 제공하는 DBMS에 IFC와 같은 BIM 모델을 업로드하고 질의방식을 통해 사용자가 원하는

정보를 추출하거나 갱신할 수 있는 시스템을 말한다. 현재 연구를 통해 알려진 오픈소스 기반 BIM서버는 많지가 않다. 이들의 특징을 Table 1과 같이 비교하였다.

Table 1. Open source-based BIM server characteristic

BIM server	Characteristic
bimServer	To improve the query performance, bimServer uses the Berkeley DB. it supports the Open API such as Web service and it's easy to analysis the BIM server architecture because of Open source. Recently, BIMql as the query language that has the query grammar is added to bimServer.
ORDB-based IFC server[7]	Object-based BIM Database server which uses the open source, CUBRID. This system has the high performance about the data query. But, there is no Open API such as Web service. Because of that, it is difficult to integrate between the heterogeneous system such as FM, GIS etc.

앞서 언급한 연구 동향과 같이 오픈소스 기반 BIM 서버를 통해 GIS클라이언트에 BIM객체를 가시화하는 비용 효과적인 방법에 관한 연구는 부족하다.

본 연구에서는 오픈소스 BIM서버를 이용해 비용 효과적으로 GIS기반에 BIM객체를 가시화하는 방법과 시스템을 개발한다.

3. bimServer 아키텍처 분석

본 장에서는 bimServer를 본 연구 목적에 맞게 활용하기 위하여 아키텍처를 분석한다. 아키텍처 분석 대상은 bimServer의 컴포넌트 프레임워크 구조, 웹서비스 구조, 프로젝트 관리 구조, 파일 호환성, 정보 질의 방법 등이다.

3.1 bimServer 컴포넌트 프레임워크 구조 분석

bimServer는 IFC파일을 데이터베이스로 관리할 수 있도록 지원해주는 오픈소스기반 BIM서버이다. bimServer는 BIM Data store로서 역할을 하며 협업을 위한 형상관리 기능을 지원한다. 다양한 모델러 및 시스템과 연동을 위해 약 470개(1.1 버전)의 Web-service를 지원하고 GIS Map 기능 구현을 위해 Google 등 클라우드 기반 시스템을 활용하고 있다. COBie (Construction Operations Building Information Exchange),

CityGML, Collada(Collaborative design activity) 등 다양한 파일 포맷을 지원하고 플러그인 형태로 기능을 확장할 수 있다. 이런 플러그인은 일종의 BIM MVD (Model View Definition)로써 동작된다.

이 오픈소스는 BIM 모델 관리를 위해 버클리DB를 사용하는 데 클라우드 시스템에서 많이 사용되는 Mongo DB와 같은 파일기반으로 처리 속도가 매우 빠르다. 버클리DB는 파일기반 DB로써 Key-Value 형식으로 데이터 레코드를 관리하기 때문에, Key에 대한 인덱스를 생성하여 빠른 검색 속도를 보장한다. 또한, 파일기반 DB이기 때문에 데이터 레코드 삽입 속도도 매우 빠르다.

bimServer는 IFC기반 BIM서버로써 동작되기 위해 Figure 2와 같은 컴포넌트 프레임워크 구조를 가지고 있다. bimServer는 크게 3가지 종류로 컴포넌트를 구분할 수 있는데 BIM모델 및 협업 관리 등을 지원하는 bimServer 플랫폼, BIM서버 기능 확장을 위한 플러그인 도구, 사용자에게 웹서버를 통한 인터페이스를 제공해 주기 위한 웹 서버이다. 플러그인 되는 모듈들은 IFC파일을 파싱하는 IfcEngine, CityGML파일을 저장하는 플러그인 등 다양하다.

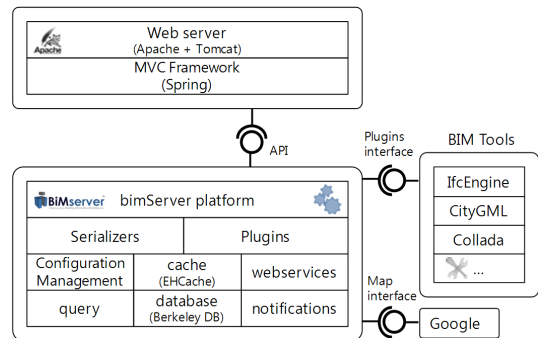


Figure 2. bimServer Component Framework Architecture

bimServer 아키텍처를 구성하는 컴포넌트는 Table 2와 같은 기능을 지원하고 있다.

bimServer는 플랫폼 개념으로 개발되어 있어 확장성이 좋다. 다른 이기종 시스템과 연계도 가능하며 이는 웹서비스나 IFC, CityGML과 같은 표준파일 포맷 입출력 기능을 이용해 처리할 수 있다. 표준파일 포맷을 이용해 데이터를 교환할 경우 IFC와 같은 BIM 파일은 포맷 변환기를 거쳐 모델서버에 저장된다. COBie와 같은 파일포맷 저장은 모델 뷰 변환을 통해 처리된다. 이러한 파일변환 처리 및 BIM객체 접근을

위해 SOAP 형식으로 웹서비스 호출을 지원하고 있다. 이와 함께 REST를 지원해 모델 변경 사항이 발생하였을 때 등록된 사용자에게 자동으로 변경 통보하는 기능을 지원한다. 이러한 아키텍처로 인해 GIS와 효과적인 데이터 상호운용이 가능하다. Figure 3은 bimServer에서 지원하는 GIS와 BIM간 데이터 상호운용 방식을 보여준다.

Table 2. bimServer Component

Component	Role Description
Database	Managing BIM Object Information by using File DB, Berkeley DB.
Cache	Cache to improve the BIM server operation performance as storing the used objects recently
Query	BIM information query support to extract the needed information from BIM DB.
Notification	Managing the notification message between the users and objects to support the collaboration and the configuration management.
Configuration Management	Configuration management support to collaborate between the objects including the users. This module has the various functions such as the model check-in, check-out, revision history.
Serializer	Data File Stream Input/Output Interface support to persist the BIM Objects including the Plugin interface mechanism to extend the Input/Output Data File Format.
Webservices	External Heterogeneous System Linkage support services by using SOAP(Simple Object Access Protocol)-based API(Application Program Interface).

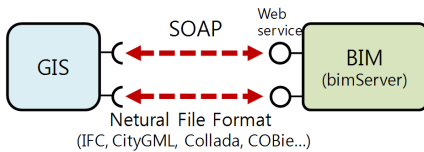


Figure 3. bimServer interoperability method between BIM and GIS

3.2 bimServer 웹서비스 구조 분석

bimServer는 다른 시스템과 연동을 위해 웹서비스를 지원하고 있으며 IFC의 BIM객체에 대한 CRUD(Create, Read, Update, Delete) 함수 뿐 아니라 bimServer

기능 자체의 확장성을 위해 시설물 관리에서 활용되는 COBie파일 포맷과 같은 IFC의 MVD(Model View Definition) 기반 파일변환을 할 수 있는 플러그인 모듈을 사용자가 구현할 수 있어 BIM서버의 기능을 효과적으로 확장할 수 있다. 이와 유사하게 IFC 파싱 처리 속도나 기능이 우수한 플러그인을 사용자가 추가로 개발해 확장할 수 있는 웹서비스 API를 지원한다. 이러한 서버 구조로 인해 사용자가 BIM서버 시스템 유지보수 비용을 감소시키고 서버의 처리 성능을 개선시킬 수 있는 효과를 얻을 수 있다.

```
<xs:element name="addObjectIDM" type="tns:addObjectIDM"/>
<xs:element name="addObjectIDMResponse" type="tns:addObjectIDMResponse"/>
<xs:element name="addProject" type="tns:addProject"/>
<xs:element name="addProjectAsSubProject" type="tns:addProjectAsSubProject"/>
<xs:element name="addProjectAsSubProjectResponse" type="tns:addProjectAsSubProjectResponse"/>
<wsdl:input name="addObjectIDMResponse" type="tns:addObjectIDMResponse" use="normal"/>
<wsdl:output name="addProject" type="tns:addProject" use="normal"/>
<wsdl:input name="addProjectAsSubProject" type="tns:addProjectAsSubProject" use="normal"/>
<wsdl:output name="addProjectAsSubProjectResponse" type="tns:addProjectAsSubProjectResponse" use="normal"/>
<wsdl:operation name="addObjectIDM" type="tns:addObjectIDM"/>
<wsdl:operation name="addProject" type="tns:addProject"/>
<wsdl:operation name="addProjectAsSubProject" type="tns:addProjectAsSubProject"/>
```

Figure 4. bimServer Web-service

Figure 4는 bimServer에서 제공하는 웹서비스의 일부를 보여준다. 웹서비스는 WSDL(Web-Service Description Language) 방식으로 표현되며, WSDL을 지원하는 Java, C#등 다양한 운영체제 플랫폼의 개발언어를 지원한다. 이러한 웹서비스 API를 지원하는 것은 BIM과 GIS간 데이터 상호운용성을 구현하기 위한 필수적인 요건이다.

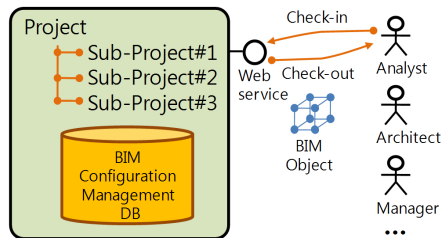


Figure 5. Project Configuration Management Method of bimServer

3.3 bimServer 프로젝트 관리 구조 분석

bimServer에서 BIM 프로젝트 하나는 하위에 여러 개의 프로젝트로 구성된다. 예를 들어 프로젝트는 건축, 구조, 토목으로 구성될 수 있다. 마찬가지로 각 하위항목도 프로젝트로 취급되어 그 밑에 세부 프로젝트 항목으로 구성될 수 있다. bimServer는 이런 프로젝트 계층 구조를 지원한다. 각 프로젝트마다 별도의 BIM모델을 관리하며 프로젝트에 등록된 여러 사용자 간 협업을 위하여 IFC파일을 기반으로 한 Checkin/Checkout과 같은 형상관리를 지원한다. 형상관리는 bimServer에 추가된 프로젝트 단위로 이루어지는 데

프로젝트는 하위 프로젝트(Sub-Project)를 가질 수 있어 복잡한 BIM 모델을 효과적으로 관리할 수 있다. Figure 5은 bimServer의 형상관리 방식을 보여준다.

3.4 bimServer 파일 호환성 분석

모델을 Checkout하거나 다운로드하는 것은 bim-Server에 등록된 Addin을 이용해 CityGML, Collada, KMZ등 다양한 표준 파일 포맷으로 변환해 저장할 수 있다. 이는 bimServer에서 추가적으로 개발할 수 있는 플러그 인이 MVD방식으로 처리될 수 있기 때문이다. MVD는 buildingSMART에서 제안한 표준으로 IFC의 서브셋(Sub-set)을 생성할 수 있는 표준 기술이다. 그러므로, MVD는 IFC파일이 포함하고 있는 속성들을 일부 형식으로 저장하는 데 활용할 수 있다. bimServer는 MVD방식을 자체적으로 구현하고 있다.

3.5 bimServer 좌표계 설정 방법 분석

bimServer는 Geolocate 객체를 이용해 IFC파일 내 포함된 건물의 단위, 원점과 방향등의 파라미터를 설정할 수 있는 기능을 지원한다. 이는 건물이 GIS좌표계에서 위치를 설정하는 데 필수적인 기능이다.

Geolocate 객체는 이러한 파라미터를 웹서비스나 사용자 GUI(Graphic User Interface)를 통해 설정할 수 있도록 지원하고 있다. GUI 방식에서는 Google map을 메쉬업(Meshup)하여 건물의 원점과 방향을 설정할 수 있다.

3.6 bimServer 정보 질의 방법 분석

bimServer의 정보 질의 방법은 서버 자체적으로 BIM객체를 버클리 DB에 등록할 때 생성되는 객체 ID를 통해 질의(Query)가 가능하고, IFC 객체의 GUID인 Globally Unique ID로도 질의가 가능하다. 또한, IfcSlab, IfcWall, IfcDoor와 같은 IFC 객체 종류 별로 IFC 객체들을 질의할 수 있다.

bimServer는 Figure 6과 같이 이런 질의를 웹서비스 API로 제공한다. 웹서비스는 XML기반의 SOAP형식으로 웹서버와 클라이언트 간 정보 통신을 지원한다. 이를 이용하면 원하는 BIM객체를 주어진 파라미터 값과 부합하는 객체를 리턴 받아 다양한 응용 도구를 개발할 수 있다.

SOAP의 형식은 웹서비스 API 이름과 API를 호출하는 파라미터로 구성되어 있으며 이를 호출 받은 측에서 SOAP메시지를 해석하여 필요한 정보 처리를 한 후에 SOAP 형식으로 리턴한다.

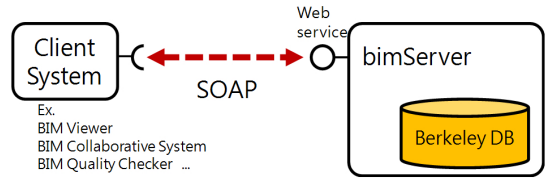


Figure 6. bimServer Object Information Query Method

4. bimServer 활용 GIS기반 BIM OVS 아키텍처 디자인

본 장에서는 분석된 bimServer의 아키텍처 구조를 활용하여 GIS기반에 BIM객체를 가시화하기 위한 시스템 아키텍처를 디자인한다.

시스템 아키텍처는 UML(Unified Modeling Language)를 사용하여 활용 관점을 정의하기 위한 유스케이스(Use-case) 디자인, 처리 순서를 정의하기 위한 시퀀스(Sequence) 디자인, 시스템 구성요소를 정의하기 위한 컴포넌트(Component) 디자인 및 컴포넌트 내 객체와 관계를 정의하기 위한 클래스(Class) 디자인을 통해 시스템의 동적, 정적 아키텍처를 정의하도록 한다.

4.1 유스케이스 디자인

GIS기반 BIM OVS 개발을 위해 도시 시설물 관리 자란 액터(Actor)를 정의한다. 액터는 BIM객체들을 bimServer에 업로드한 후 각 객체들이 포함하는 건물들의 좌표계를 정의하기 위해 Geolocate 파라미터를 설정한다.

BIM 객체들을 GIS상에 가시화하기 위해 제일 문제가 되는 부분은 BIM 객체의 형상정보를 파싱해 가시화 형태의 메쉬(Mesh)로 변환하는 시간이 너무 오래 걸린다는 것이다. 사용자가 시설물 관리의 목적으로 GIS클라이언트 상에서 BIM객체를 가시화하고 조작할 때 매번 각 객체의 형상을 메쉬 형태로 변환한다면 시스템 응답속도가 너무 떨어진다. 응답속도를 높이기 위해 BIM객체들을 최적화한다. 최적화를 위해 BIM객체의 형상 정보를 추출해 경량화된 메쉬 파일(BLF)로 저장한다. 이 BLF는 BIM객체를 쿼리하여 형상을 가시화할 때 사용한다. BIM객체의 속성 정보는 bimServer의 SOAP방식을 통해 얻는다.

Figure 7의 유스케이스 다이어그램은 기술한 유스케이스 시나리오를 정의한 것이다. 유스케이스의 액터(Actor)는 IFC파일을 이용해 파일내의 BIM객체를 bimServer에 등록한다. 그리고 BIM건물의 위치를

Geolocate서비스를 이용해 설정한다. bimServer에 BIM객체를 등록한 후, BIM객체 최적화 유스케이스를 수행하기 위해, 형상을 경량화된 메쉬 파일 형식으로 변환한다. 액터가 정보를 질의할하면 bimServer의 웹서비스 기능을 이용해 객체 정보를 검색한다. BIM객체의 시각화는 형상과 속성 정보 시각화 유스케이스로 나뉘며, 형상은 앞서 경량화된 형상 메쉬파일을 이용하고 속성 정보는 bimServer의 질의 웹서비스를 이용해 처리한다.

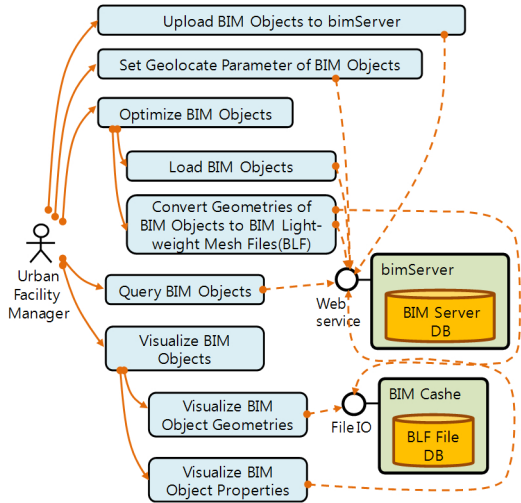


Figure 7. BIM OVS Use-cases Diagram

4.2 시퀀스 디자인

BIM OVS 유스케이스들 중 핵심적인 역할을 하는 BIM 객체 최적화하는 시퀀스는 크게 BIM객체들을 bimServer로부터 로딩하는 부분과 로딩 된 BIM객체들의 형상정보들을 BLF로 변환하는 부분으로 나뉘진다. BLF변환을 위해서 BIM 표준파일인 IFC에서 B-Rep(Boundary Representation)정보를 메쉬 정보로 변환할 수 있는 변환기가 필요하다. Figure 8은 이를 표현한 시퀀스 다이어그램이다. OptimizeBIMObject() 함수를 호출하면 내부적으로 현재 등록된 bimServer의 객체들을 열거하고 각 객체에 포함된 B-Rep형상정보를 메쉬변환기를 이용해 메쉬를 생성한다. 본 연구에서 메쉬변환기는 오픈소스인 OpenCASCADE를 사용하였다.

BIM객체 형상 및 속성정보를 쿼리하는 부분은 각각 BLF Cache File DB와 bimServer DB에서 제공하는 API를 이용한다. 각 BIM객체들에 대한 속성정보를 얻기 위해 다음과 같이 bimServerClient에 정보 쿼리

웹서비스 API를 쿼리 문과 함께 호출한다. bimServer에서 지원하는 쿼리는 SQL과 유사하며 BIMql이란 BIM객체 쿼리하는 엔진을 통해 서비스된다.

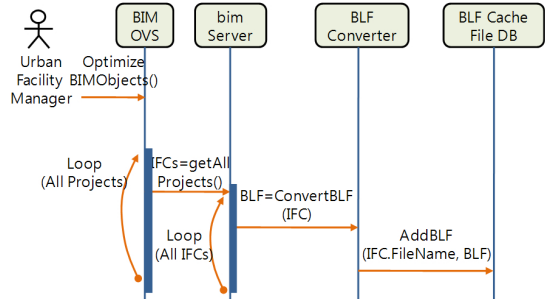


Figure 8. Optimize BIM Objects Use-case Sequence Diagram

```

List<Project> projects
= bimServerClient.getAllProjects();
for(Project project:projects)
{
    List<Revision> revisions
    =bimServerClient.getAllRevisionsOfProject(project);
    for(Revision revision:revisions)
    {
        File IFC
        = bimServerClient.downloadQuery
        (revision,
        "Select $Var1
        Where $Var1.EntityType=IfcSlab");
    }
}
    
```

bimServer 아키텍처 분석에서 기술한 것과 같이 프로젝트 객체들을 획득하고 각 객체들의 리비전 객체를 다시 얻은 후에 BIMql을 이용해 필요한 정보를 SQL형태로 질의할 수 있다.

4.3 컴포넌트 및 클래스 디자인

BIM OVS의 컴포넌트 구조는 유스케이스를 고려해 Figure 9와 같이 디자인되었다. GIS클라이언트는 BIM서버에서 웹서비스 프로토콜인 SOAP를 사용해 객체 정보를 얻으며, 형상을 가시화하기 위해 메쉬변환기인 BLF Converter를 사용한다.

유스케이스와 시퀀스 시나리오를 구현하기 위한 핵심적인 컴포넌트의 클래스 다이어그램은 Figure 10과 같다.

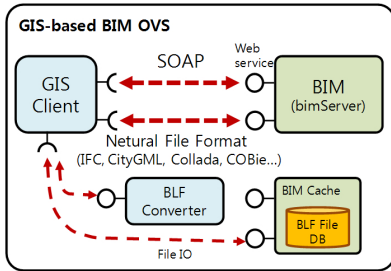


Figure 9. GIS-based BIM OVS Component Diagram

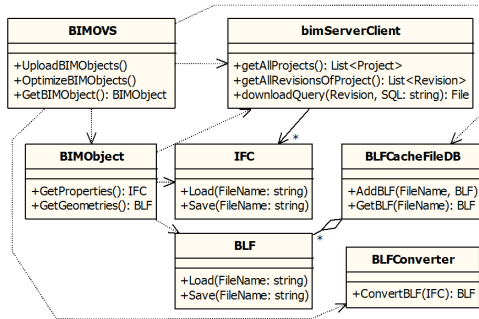


Figure 10. GIS-based BIM OVS Class Diagram

Table 3. GIS-based BIM OVS Classes Responsibility

No	Class	Responsibility
1	BIMOVs	Controlling the other classes such as bimServerClient, BLFConverter, BLFCacheFileDB to support the BIM Object Visualization services.
2	bimServer Client	Supporting the bimServer API such as getAllProjects, getAllRevisions OfProjects and the BIM information query function.
3	BIM CacheFileDB	BIM Cache DB to manage BLF File.
4	BLFConverter	Converter to transform IFC to BLF File.
5	BIMObject	BIM Object Class to virtualize the IFC and BLF information takeoff mechanism such as the algorithms.
6	IFC	IFC File which has BIM Object Properties.
7	BLF	BLF File which has BIM Object Shape Meshes converted from IFC's Geometries by using BLFConverter.

Figure 10의 각 클래스에 대한 책임은 Table 3와 같이 정의하였다. 이 클래스들 중 BIMOVs는 다른 클래스들의 객체들을 관리하면서 사용자에게 필요한 서비스를 제공해 주는 컨트롤을 책임진다.

5. bimServer 활용 GIS기반 BIM OVS 시스템 개발

디자인된 아키텍처를 바탕으로 GIS-based BIM OVS를 개발해 적용성을 시험해 보았다. GIS클라이언트는 WWJ(World Wind Java)를 활용하였으며 오픈소스 bimServer를 활용해 BIM객체들을 관리하도록 하였다. BLF변환기는 OpenCASCADE를 활용해 개발하였다. bimServer로부터 최적화된 BIM객체의 형상들을 관리하기 위한 BLF Cache File DB는 별도로 개발하였다.

4장의 Figure 7에 정의된 유스케이스 시나리오를 테스트하기 위해 건축 구조 및 MEP(Mechanical, Electronic, Plumbing) 객체 모델이 포함된 IFC파일을 bimServer에 업로드하였다. 업로드 된 IFC파일은 객체에서 형상 정보는 경량화된 파일로 변환되어 서버에 저장된 후, GIS클라이언트 상에 가시화된다.

객체의 속성정보는 BIMql을 이용해 질의하도록 하였으며, 질의방법은 Figure 4, Figure 6과 Figure 9에 기술된 SOAP기반 웹서비스 방식이다.

4장에서 설계된 아키텍처를 바탕으로 개발된 BIM OVS의 메뉴를 통해 로드된 BIM건물 객체들을 가시화하고 속성정보를 질의해 보았다.

Figure 11, 12는 IfcSlab와 IfcFlowSegment에 대한 객체와 속성정보가 GIS클라이언트 위에서 통합적으로 가시화된 모습을 보여준다. 통합된 가시화를 위해 활용한 컴포넌트는 오픈소스를 활용한 것으로 별도의 컴포넌트 구입비용은 없었으며, 개발에 들어간 노력은 오픈소스를 분석하고 설계한 유스케이스를 구현하는데 필요한 사용자화(Customization) 비용뿐이었다.

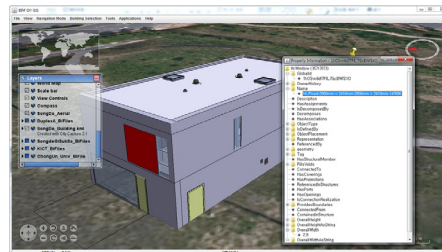


Figure 11. GIS-based BIM Objects Visualization

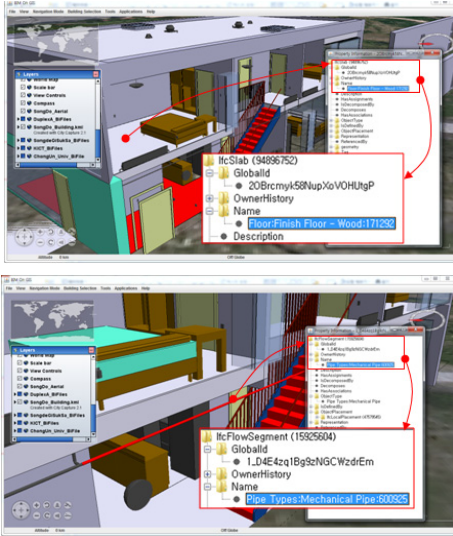


Figure 12. IfcSlab and IfcFlowSegment Object's Geometries and Properties Visualization

6. 오픈소스 기반 BIM서버 개발 방식의 효과 분석

오픈소스를 사용하지 않고 BIM서버를 구현할 때 노력과 오픈소스 기반 bimServer를 이용해 BIM서버를 구현할 때 비용 차이를 추정해 보기 위해 BIM서버 개발 경험이 있는 5년 이상의 중급 개발자를 대상으로 하여 5점 척도를 이용해 설문을 하였다. 다만, 현실적으로 BIM서버 개발 경험이 있는 개발자 표본수를 확보하기 어려워, 실제 설문에 참여한 개발자는 7명임을 밝힌다. 설문 항목은 설계된 유스케이스를 구현하는데 필수적인 기능인 7개 기능을 Table 4와 같이 정의하였다.

5점 척도는 BIM서버를 오픈소스가 없이 직접 개발하는 방식과 오픈소스를 기반으로 개발하는 두 가지 개발 방식의 노력을 다음과 같이 정의해 질문하였다.

- 1점 - 2주 이내 작업 노력
- 2점 - 4주 이내 작업 노력
- 3점 - 8주 이내 작업 노력
- 4점 - 16주 이내 작업 노력
- 5점 - 예측이 어려움

설문 분석 결과는 Table 5와 같다.

두 개발 방식의 차이는 평균 1.4 차이로 나타났으며, 각 기능 별 평균 개발 기간이 약 3주가 차이 나는 것을

의미한다. 이는 개발자가 BIM서버 개발 시 OD방식이 DD방식에 비해 상대적으로 개발 노력이 적다고 인식하는 것을 알 수 있다. 또한, OD방식이 IFC파서 개발 시 효과가 가장 크고, 사용자 응용 목적에 따라 다를 수 있는 GUI 기능에 해당하는 F7은 효과가 1.0으로 낮음을 알 수 있다.

다만, 각 기능별 차이가 1.0에서 2.0사이 정도로, 그리 크지 않은 이유는 OD방식을 사용하더라도 활용 목적에 따른 오픈소스의 사용자화가 필요하며, DD방식에 비해 분석 시간에 많은 노력이 소요되는 문제에 기인한 것으로 보인다.

정성적 분석을 위해 OD개발 방식의 한계점이 무엇인지 별도로 질의를 하였으며, 다음과 같은 의견이 있었다.

첫 번째, 응용 목적에 완벽히 부합하는 오픈소스를 찾기 어려우므로 좀 더 상세한 사용자화 비용 분석 과정이 필요하다.

Table 4. BIM Server Core Functions

No	Functions
F1	IFC Parser
F2	IFC Object DB Creation and Management
F3	IFC Object Web-service Support (BIM Object CRUD-Create/Read/Update/Delete)
F4	BIM Project Management (Project Object CRUD)
F5	Multiple User DB Access and Control
F6	External Plugin Support to extend the Server Functions
F7	Web Interface for the BIM Server Management

Table 5. Difference Analysis between BIM Server Direct Development and Open source-based Development

No	Direct Development (DD)	Open source-based Development (OD)	Difference
F1	4.3	2.4	1.9
F2	3.9	2.3	1.6
F3	3.9	2.6	1.3
F4	3.3	2.0	1.3
F5	3.9	2.7	1.1
F6	3.3	1.9	1.4
F7	2.4	1.4	1.0
Average	3.6	2.2	1.4

두 번째, 오픈소스의 라이선스 정책을 고려한 개발이 필요하다.

세 번째, OD방식은 DD방식에 비해 별도의 학습 시간이 필요하다. 대부분의 오픈소스는 상용 SW라이브러리에 비해 사용자 매뉴얼 등 관련 문서가 매우 부족하다. 학습 시간은 해당 오픈소스의 아키텍처 분석부터 기능 활용 방법, 그리고 예제 작성 등을 포함해야 한다.

7. 결 론

본 연구는 비용효과적인 GIS기반 BIM객체 가시화 시스템을 개발하기 위해 오픈소스 BIM서버를 효과적으로 연계하는 방법과 아키텍처를 제안하였다. 이를 위해 최근 해외에서 활용되기 시작한 오픈소스 기반 BIM서버인 bimServer의 구조를 분석하여 이를 아키텍처 설계에 활용하였으며, 아키텍처의 적용성을 확인하기 위해 GIS기반에 BIM객체를 가시화하는 시스템을 개발하고, 이를 통해 BIM객체가 BIM서버를 통해 GIS와 효과적으로 가시화되고 통합될 수 있음을 보였다. 또한, 이를 바탕으로 BIM서버 개발자에 대한 설문 조사를 통해 BIM서버 개발 시 오픈 소스를 이용하지 않는 직접 개발 방식과 오픈 소스 기반 개발 방식과의 차이를 비교하여 시사점을 도출하였다.

향후 제안된 아키텍처를 바탕으로 GIS에서 수만 개 이상의 대용량 BIM객체를 BIM서버에서 질의할 때 발생할 수 있는 성능 저하 문제를 확인하고 개선점을 연구할 계획이다. 본 연구에서 제안한 효과를 이를 바탕으로 도시 차원의 시설물 객체들을 BIM객체로 변환해 GIS상에서 실시간으로 객체를 가시화할 수 있는 효과적인 방법에 대해 연구할 계획이다.

References

[1] 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, 3(2):10-19.

[2] 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, XL-2(W2):31-34.

[3] 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, 20(5):49-56.

[4] Jakob, B; Leon, V. B; Ruben, D. L; Pim, D. H, 2010, An open source IFC Model Server, Proceedings of the CIB, 27(78):1-8.

[5] Jeong, J. Y, 2011, A study of efficient BIM data exchange and sharing, Yonsei University.

[6] Kang, T. W; Hong, C. H; Hwang, J. R; Choi, H. S. 2012, The External BIM Reference Model Suggestion for Interoperability Between BIM and GIS, Korea Spatial Information Society, 20(5):91-98.

[7] Lee, G; Jeong, J. Y; Won, J. S; Cho, C. Y; You, S. J; Ham, S. I; Kang, H. S. 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, 28(2):210-222.

[8] Tomo C, 2011, A review and outlook for a Building Information Model(BIM): A multi-standpoint framework for technological development, Advanced Engineering Informatics, 25(2):224-244.

[9] Umit I, 2012, Design patterns for BIM-based service-oriented architectures, Automation in Construction, 25:59-71.

[10] 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, 33(4):1719-1730

논문접수 : 2014.01.04

수정일 : 2014.02.26

심사완료 : 2014.02.27