3DCITYDB4BIM: A SYSTEM ARCHITECTURE FOR LINKING BIM SERVER AND 3D CITYDB FOR BIM-GIS-INTEGRATION

I. H. Hijazi [1,2]*, T. Krauth[1], A. Donaubauer[1], T. Kolbe [1]

[1] Chair of Geoinformatics, Technical University of Munich, Arcisstr. 21, Munich, Germany - (ihab.hijazi, andreas.donaubauer, thomas.kolbe)@tum.de
[2] Department of Urban Planning Engineering, An-Najah National University, Nablus, Palestine – (eehab@najah.edu)

Commission IV, WG 10

KEY WORDS: 3D City Models, BIM, IFC, BIM Server, 3DCityDB, CityGML, glTF, CesiumJS, GeoBIM

ABSTRACT:

Nowadays, the three-dimensional visualization of cities is becoming feasible and popular. The applications serve key sectors of urban development including architecture and urban planning. The implementation of BIM in the geospatial environment will help in evaluation of the design proposals by answering questions such as “What windows have views of a particular spot?”, “How will a design proposal affect views and shadows in an urban scene?” from multiple stakeholder perspectives.

This paper presents an approach that facilitates the integration of BIM and geo visualization. 3DCityDB4BIM is a concept for combining existing open source software and open specifications for efficient data management and visualization of Building Information within its broader context. 3DCityDB4BIM core components are the 3DCityDB, the BIMServer, CesiumJS viewer, and the existing standards (glTF, CityGML and IFC). The paper presents the procedure used to enable the development of this new expert system as applied to the context of 3D Web GIS. A thorough analysis of the strengths and weaknesses of these different components has been undertaken to employ their strengths and overcome their weaknesses. The system architecture is providing an extension to the 3DCityDB linking to BIMServer without violating indoor information privacy issues. The 3D visualization format glTF works as the integration medium to visualize BIM and CityGML data in web browsers without plugins or other additional software on the client side. The developed software is open source.

1. INTRODUCTION

Earth’s population is growing, cities are getting bigger and with-it construction industries, facility management and the geospatial domain are developing new methods to tackle upcoming challenges, like developing more efficient methods to aid construction projects and managing big amounts of data (Song et al. 2017). One of these new methods is BIM-GIS integration. Building Information Modelling (BIM) provides “a digital representation of physical and functional characteristics of a single building. As such, it serves as a shared knowledge resource for information about a building forming a reliable basis for decisions, during its lifecycle from inception onwards” (wbdg.org, 2018).

Two most well-known open standardized information models in BIM and 3D GIS domains are: IFC for BIM (buildingsmart, 2019) and CityGML for GIS (Gröger et al., 2012). IFC serves as an exchange format between different BIM platforms and its building models preserve all details that are built into it while modelling. CityGML models on the other hand can include entire cities in a format that is relatively simple and easy to exchange. It is mainly used for spatial analyses like estimations of solar potential and energy consumption. Integration of BIM within GIS for visualization-only use cases such as communication of urban information and virtual reality, which do not necessarily rely on spatial operations, but where BIM is visualized within 3D city models have been found as an important application for 3D city models (Arroyo Ohori et al, 2017). It aims to provide: 1) Visualization of the urban environment 2) Visualization of design proposals 3) Online communication of users 4) Gathering user feedback 5) Analyzing the social and environmental impact of proposals.

In the recent years many studies showed how to integrate BIM data into GIS and although many of these efforts have been successful, most of them did not show the exact needs for such an implementation. Examples for this would be (Isikdag, 2006) by achieving information transfer from IFC to Shapefiles and Geodatabases and (Lapierre and Cote, 2008) who completed tests on the integration of CityGML and IFC models in OGC OGC Test bed No 4 (OWS-4). More examples like these can be found in the following review on integration methods (Hijazi and Donubauer, 2017). Opportunities for construction sites and urban management are mainly not part of integration studies (Iskidag and Zlatanova, 2009). “The applications are still limited in the use of integrated BIM-GIS, and the strengths of both BIM and GIS have not been fully integrated and utilized” (Song et al., 2017). Fosu, et al., 2015 reviewed 40 recent papers on different BIM-GIS integration approaches. 33 of them focused either on data conversion for one specific use case or tried to implement data by using new tools, extensions, frameworks, and/or ontologies. Only four papers tried to achieve a seamless integration by fully combining BIM and GIS features and capabilities in one Unified building model (UBM) (El-Mekawy et al., 2012). This shows a big gap in research focusing on integration. In addition, Fosu et al., 2015 mentions that future research must focus on “how much access and restrictions various stakeholders will have to the BIM/GIS assets database, and where this data will be cached” as well as “how to resolve clashes and inconsistent records entered in real time.” They point out that the reason for this lies in the upcoming huge amount of data and the possible loss

* Corresponding author

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-4-2020-195-2020 | © Authors 2020. CC BY 4.0 License.

of control. There is a need to provide a linking mechanism that allows to integrate BIM models within 3D city models without violating privacy issues, information overload and considering that changes in building information take place on facility management scale.

In this paper we present a Web based 3D GIS solution that allows linking Building information models stored in BIM server (BIMServer, 2019) within 3D city models. It is extending the 3DCityDB (open source database system for semantic 3D city and landscape models) (Yao et al., 2018) with a link to BIM Server. Using the proposed solution it is possible to perform queries in the BIM models and provide direct visualization using 3DCityDB geospatial browser. The system presents a new concept to process, analyze and visualize BIM models within their geographic context. Emphasis is laid on the integration of BIM information without loss of privacy by using a restricted access to the BIM data sources. 3CityDB4BIM (Visualizing BIM in 3D City Models) is entirely built in an Open Source environment, which can be downloaded from the Internet for free.

The rest of the paper is organized as follows: Section 2 provides a description of the information system and its components (3DCityDB, BIMServer and GUI for visualization and interaction). Section 3 describes the system design and developments using specific OSS technologies; in section 4 we draw conclusions and give suggestions for future research and implementation.

2. THE 3DCITYDB4BIM SYSTEM ARCHITECTURE

Conceptually, the approach to BIM-GIS integration presented in this paper is based on the integration of spatial 3D objects from distributed, heterogeneous systems on the visualization layer instead of the data layer, as suggested by Altmaier and Kolbe (2003). A transformation of BIM data into GIS data structures is not required. Instead, the approach allows for visualizing BIM and 3D GIS data in a common 3D geovisualization client, however with a direct connection to the native BIM model. The design strategy considers maintaining the privacy of the BIM contents by keeping BIM data in a separate BIM server, and providing access to the GIS users to integrate the required BIM data based on specified queries which can be created at runtime through a Web interface. The software components of the system were selected according to two major criteria, i.e. mature Open Source software projects, and projects with wide development communities. The system proposed in the paper is highly relevant to both AEC and the geospatial industry. The system simplifies the process and facilitates the integration by creating a vehicle to visualize BIM data (IFC) into a user accessible 3D navigation and viewing environment, which also provides some analytical operations, such as semantic selection of objects from the BIM model and the 3D city / landscape model.

BIM and GIS integration at application level is usually for specific use cases. Thereby, both the source data and the object data are not changed and also no service or ontology is developed. Instead, for instance, the analysis might be performed separately and the results work as an exchange medium (Deng et al., 2016a). Another possibility, as seen in Irizarry et al. (2013), is to just extract the necessary information from BIM in a GIS compatible format to perform further analysis therein. A third example is the conversion of IFC into another schema for analysis purposes like in Niu et al. (2015). In this case, the result of the analysis was converted into KML and COLLADA in order to visualize it in a GIS. Thus, a multipurpose 3D visualization format works as an integration medium. The approach followed in 3DCityDB4BIM is similar to this approach, but extends it by providing access to the native BIM model and the native 3D city / landscape model. For this purpose, a server with a BIM database and a 3D GIS database serve as a backbone infrastructure and geometry and appearance data from both data sources are converted to the visualization format glTF, whereas the attributes from BIM and 3D GIS are directly accessed through the BIM server API and a Web Feature Service respectively. In contrast to current approaches mentioned above, the data flow from BIM is strictly regulated by access privileges and the detailed BIM models is not shown when the user does not have proper access rights. Figure (1) shows the schematic system architecture. The prototype named 3DCityDB4BIM is composed of three components.

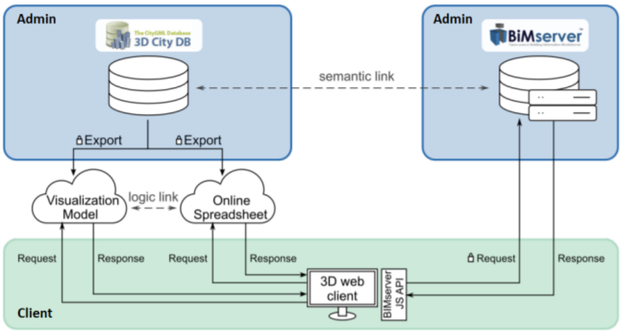


Figure 1. Architecture of the prototype- 3DCityDB4BIM, the lock icons mark processes where authentication is necessary

2.1 BIM server

BIMserver is an open source model server which is based on the open standard IFC and enables storage, maintenance and query of IFC based building information models (Beetz et al. 2010). Instead of being an end-user ready product, BIMserver serves as a basis for building BIM software tools on BIMserver (2019). For this it has many open interfaces and network protocols like SOAP, JSON and Protocol Buffers through which just the information that is needed can be retrieved. This makes it easier to create these tools because many tasks are already solved by the BIMserver. Within the BIMserver, data are stored using the key-value database BerkeleyDB which interprets and treats the data as objects. This allows to query, merge and filter the BIM data. In contrast to conventional, workstation based CAD and BIM software, BIMserver has a centralized model. Therefore, all participants of a project are able to work on the same model without having to worry about interoperability issues and having the latest version of a file. Additionally, the server can automatically respond to outside changes like the pricing and availability of materials (Harrison, 2009). BIMserver is written in Java and either runs as a stand-alone program using the Jetty library or within a Java web application server like Tomcat. The IFC models are structured by the user in projects and subprojects for which access rights can be assigned to other users. Adding an IFC model to a project is called "check in". By doing this, for each entity instance in the model a unique key is created, consisting of the object id, the project id and the revision number (Beetz et al., 2010). To receive an IFC model a download can be initiated with the revision id of a project and a query for filtering the model. For encoding the download, different serialization can be used like, for example, Ifc2x3tc1, IfcXML 2x3tc1, JSON, or

Binary glTF 2. The latter was the most important for this work, since it is possible to add glTF to the 3D web client of 3DCityDB. The query language used by the BIMserver is JSON based. A typical query that is also used in the prototype of this work can be seen in Figure 2. It returns all walls.

```
  {
        "type": {
                  "name": "IfcWall",
                      "includeAllSubTypes": true
                  }
     }
```

Figure 2: An example BIMserver query which returns all Walls of an IFC model

2.2 3DCityDB

3DCityDB is a free geo database on top of spatially extended relational database management systems like PostgreSQL with the PostGIS extension and Oracle Spatial. Its purpose is to store, represent, and manage 3D city and landscape models represented according to the CityGML standard.

It is an Open Source tool. 3DCityDB consists of three main components: Firstly, the database schema, which represents spatial properties and semantic data of 3D city models, supports CityGML 1.0-2.0, LoD0-4, appearance data, digital terrain models (DTMs), and more. Secondly, the Importer/Exporter tool used for up- and downloading models and tile based import and export. It is also capable of creating KML, COLLADA and glTF formats for visualization purposes in different digital globe browsers. Some other features include XML validation of CityGML documents, coordinate transformation, spreadsheet export, and query capabilities. An OGC WFS implementation for 3DCityDB allows to query, retrieve and update the native CityGML data stored in the database over the Internet. Thirdly, the 3DCityDB-Web-Map-Client serves as an extension of the CesiumJS open source digital globe. It is capable of dynamically loading and unloading of large 3D city models, can manage different layers and models and can be linked to online spreadsheets like Google spreadsheets. Models are visualized interactively by retrieving data on mouse clicks. Several interactions are integrated e.g. toggle shadow casting (Yao et al., 2018).

The process for installing the 3DCityDB, the import and export of buildings, the creation of online spreadsheets as well as the adding of exported buildings to the 3D web client is described in detail in Yao et al. (2018). However, when creating the online spreadsheet with the 3DCityDB Importer Exporter, a column with bounding box of the city object – building has to be added, the bounding box information is required by AttributeProcessor described in section 3.2.

3. SYSTEM IMPLEMENTATION

The BIMserver extension for the 3D web client is completely written in JavaScript and HTML. Different requirements need to be considered to jointly visualize and query BIM and 3D GIS data in the 3D web client. These are as follows:

- IFC models and CityGML buildings need to be linked. This is achieved via a link table (section 3.1).
- As BIM models quite often do not contain complete and correct georeferencing information, the system needs to provide a simple and efficient method to georeference the BIM models based on the geometry of the corresponding CityGML buildings. Both BIM and 3D GIS data need to be transformed to a common coordinate reference system. In our implementation WGS84 is used (section 3.2).
- BIM models need to be visualized in a 3D virtual globe. This is achieved by transforming the georeferenced models into the glTF format (section 3.3).
- The user needs to be able to query BIM models via the 3D Web Client. This is achieved by storing predefined queries in JavaScript code (queryMap.js).

The remainder of this paper documents the development of the application logic as well as the development of a graphical user interface for the system.

The sequence diagram shown in figure 3 illustrates the interactions between the different components of the system. When the user is logged into the BIMserver, selects a CitGML building and clicks the "InsertBIM-Model" Button (figure5c), the script.js filters the bimCityTable in order to check if a corresponding IFC model exists (section 3.3). If so, the name of the BIMserver project, as well as all the GMLIDs of the corresponding CityGML buildings are returned. Next, the geodetic coordinates of the centroid of all the corresponding CityGML buildings is calculated by the AttributeProcessor (section 3.2).

Subsequently, with the help of the BimserverGltfRequestor, the download of the glTF representation of the IFC model is initiated through the main class of the BIMserver JavaScriptClient, BimServerClient. When the model is prepared, a download URL for the glTF model is returned and added to the 3D web client.

3.1 Linking of CityGML buildings with IFC models - bimCityTable

A way had to be found how to create links between the GMLID of CityGML buildings and the corresponding IFC models. Using this information, it is possible to replace a CityGML building by the appropriate IFC model, without having to choose it manually. This link could be defined on IFC entity level, on BIMserver project level or even spatially. In the prototype we developed, the link was created on project level. Other possibilities are discussed in section 4.1.

The links were established using a JavaScript array of multiple objects. Figure 4 shows an example of this array. Each object represents a link between an IFC model and all the CityGML buildings. That lie within the extent of this model. The object property "model" has the name of the subproject of a certain IFC model as value. The property "related" is an array of all GMLIDs of all CityGML buildings corresponding to that IFC model. Whereas the subproject name can be easily looked up in the BIMserver by the user, finding out the corresponding GMLIDs is a bit more complicated. As an initial step, the user can select the GMLID of a building that is most likely within the area of the IFC model. This can be done by clicking on a CityGML building in the 3D web client. This GMLID then has to be added to the corresponding model in the bimCityTable. By importing the model as described in section 3.3 the centroid of the IFC model will be in the centre of the previously selected CityGML building. By moving the model, the model can be placed correctly. This makes it possible to click on the other CityGML buildings that intersect with the model and write their GMLIDs in the array bimCityTable. The table was written into a separate file to facilitate user changes. Thus, the user can add new links without having to change the actual code.

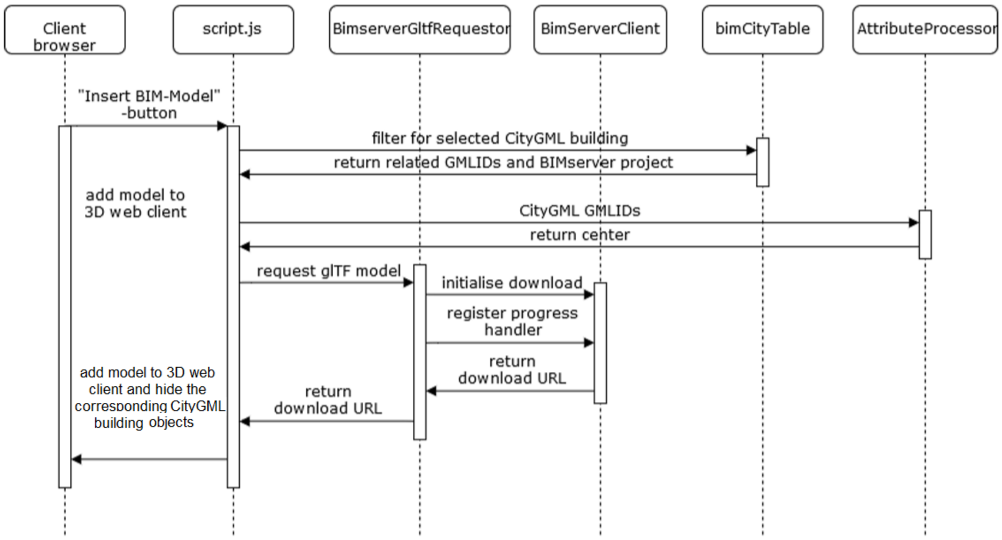
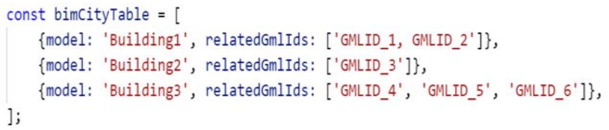

Client browser
script.js
BimserverGltfRequestor
BimServerClient
bimCityTable
AttributeProcessor
"Insert BIM-Model" -button
filter for selected CityGML building
return related GMLIDs and BIMserver project
add model to 3D web client
CityGML GMLIDs
return center
request glTF model
initialise download
register progress handler
return download URL
add model to 3D web client and hide the corresponding CityGML building objects
return download URL

Figure 3: Sequence diagram representing the interactions between different software components

The links were established using a JavaScript array of multiple objects. Figure 4 shows an example of this array. Each object represents a link between an IFC model and all the CityGML buildings that lie within the extent of this model. The object property "model" has the name of the subproject of a certain IFC model as value. The property "related" is an array of all GMLIDs of all CityGML buildings corresponding to that IFC model. Whereas the subproject name can be easily looked up in the BIMserver by the user, finding out the corresponding GMLIDs is a bit more complicated. As an initial step, the user can select the GMLID of a building that is most likely within the area of the IFC model. This can be done by clicking on a CityGML building in the 3D web client. This GMLID then has to be added to the corresponding model in the bimCityTable. By importing the model as described in section 3.3 the centroid of the IFC model will be in the centre of the previously selected CityGML building. By moving the model, the model can be placed correctly. This makes it possible to click on the other CityGML buildings that intersect with the model and write their GMLIDs in the array bimCityTable. The table was written into a separate file to facilitate user changes. Thus, the user can add new links without having to change the actual code.



```
const bimCityTable = [
    {model: 'Building1', relatedGmlIds: ['GMLID_1, GMLID_2']},
    {model: 'Building2', relatedGmlIds: ['GMLID_3']},
    {model: 'Building3', relatedGmlIds: ['GMLID_4', 'GMLID_5', 'GMLID_6']},
];
```

Figure 4: An example of the bimCityTable

3.2 Positioning of the model - AttributeProcessor

After the link between the IFC models and the CityGML buildings (section 3.1) was created the subsequent step is to automatically position the IFC model in the correct spot. Most of the IFC models used in this project were incorrectly georeferenced, some not at all. Since this is often the case (Clemen and Hendrik, 2019), this work aimed to find a simple and efficient way to position IFC models which do not contain georeferencing information as good as possible.

To add a glTF model to the 3D web client, a point with x, y and z coordinates and a rotation has to be specified. The model is then placed in the map having the origin of its local coordinate system at exactly this point. The Binary glTF 2 serializer of the BIMserver sets the origin of the local coordinate system in the centre of the model independently of the actual point of origin of the IFC model. This can help to estimate the approximate position.

When CityGML buildings are added to the 3DCityDB, 3D bounding boxes are automatically calculated. These can be added to the online spreadsheet, which gives access to them within the 3D web client. This makes it possible to calculate the centre of these bounding boxes and thus, the approximate centroid of the buildings. If multiple CityGML buildings are related to one IFC model, an aggregated bounding box can be calculated by aggregating the bounding boxes of the individual buildings. This is illustrated in figure 5. Since these are 3D bounding boxes, not only the latitude and longitude of the centroid can be calculated, but also its height.

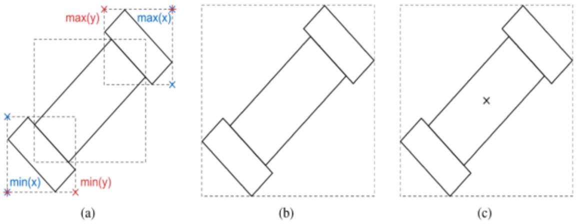


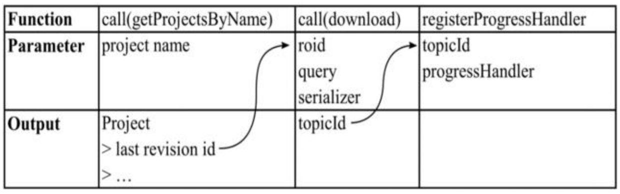
Figure 5: Simplified 2D example showing the steps for calculating the centroid of the bounding boxes for multiple related buildings. Dashed lines represent the bounding boxes, solid lines the buildings. First the minimum and maximum x and y values are chosen (a), next the joint bounding box is created (b), finally the center is calculated (c).

These calculations are done by the JavaScript class AttributeProcessor. This class has two main functions. One is needed to query the bounding boxes of the CityGML buildings from the online spreadsheet. The related GMLIDs from the bimCityTable (section 3.1) are used as input. The other function calculates the centre of the previously queried bounding boxes as described above. Additionally, the JavaScript library proj4.js is used to transform the calculated centroid from the original coordinate reference system of the CityGML objects (in our example EPSG: 31468, Gauß Krüger

Zone 4) into WGS84, which is needed to add the glTF model to the 3D web client.

3.3 Integration of Building models in 3DCityDB web client BIMserverJSAPI - BimserverglTFRequestor

In order to access and extract the required building models from the BIMserver, the BIMserver API JavaScriptClient was used.



Function	call(getProjectsByName)	call(download)	registerProgressHandler
Parameter	project name	roid query serializer	topicId progressHandler
Output	Project > last revision id > …	topicId	

Figure 6: The functions of the JavaScriptClient used by the BimserverGltfRequestor to initiate a download and register a progress handler. The arrows indicate which output is used as an input for the subsequent function.

Figure 6 shows the functions of the BimServerClient that are called by the BimserverglTFRequestor in order to initiate the download and register a progress handler. First, the "getProjectsByName" method is called. As a parameter it only requires the project name. This is taken from the bimCityTable (section 3.1). It returns information about the project. The last revision id of this project is needed as an input for the download method call. Next to the revision id, also a query and the serializer id are required. The query is taken from the user interface or from predefined queries the user can choose from (section 3.4.1). The serializer is the "Binary glTF Serializer 2". The download call returns a topic Id which is an identifier for a specific building and is needed to register a progress handler for this specific download. In addition to the topic Id a progress handler is needed as input for the registerProgressHandler function. The code of the progress handler of the BimserverGltfRequestor is taken mainly from the source code of BIMview with a few adjustments. It checks the state of the download in small time steps. In case of an error, the download is aborted.

When the state of the download is "Done preparing" the progress handler executes the BimServerClient function generateRevisionDownloadUrl to get the download URL. The URL is then used to insert the glTF model into the 3D Web Client. Subsequently the state of the download turns to "FINISHED" which causes the progress handler to execute the BimServerClient function unregisterProgressHandler and call the cleanupLongAction interface method. This causes the progress handler to stop and the download to be finished.

Besides the methods mentioned above, the BimserverGltfRequestor also makes use of the login and logout functions of the BimServerClient.

3.4 Extensions of the 3DCityDB Web-Map-Client - script.js

The most important addition to the 3DCityDB Web-Map-Client is the function that connects the 3DCityDB to the BIMServer. Figure 10 illustrates the sequence of conditions and tasks that are executed after pressing the "Insert BIM-model" button, shown in section 3.3. The function checks if there are any active layers within the 3D web client. If so, the next condition is that there is exactly one CityGML building selected. If none or many are selected the function is aborted and an error message is displayed in the console. If the condition is met, the function checks what query type for the IFC model is selected. Predefined queries are stored in and can be selected in the Graphical User Interface. In case of a custom query, the function obtains the query directly from the HTML input element. Subsequently it is verified if there is an IFC model for the selected building in the bimCityTable. If so, the centroid of all related CityGML buildings is calculated (section 3.2), the corresponding glTF model is downloaded from the BIMserver and finally the model is added to the 3D web client at the calculated centroid location.

Another important extension is the function that enables to move and rotate the glTF model after adding it to the 3D web map. This is realized with a JavaScript DOM EventListener that reacts to arrow key events. For moving and rotating a glTF and changing the model height. The function is also helpful to find out which CityGML buildings belong to an IFC model (3.1).

3.4.1 Graphical User Interface

The graphical user interface of the BIMserver extension was created in such a way that it matches the design of the 3D web client. The collapsed BIMserver panel is shown in Figure 7. It was inserted below the Terrain Layer panel. Figure 8 shows different stages of the expanded BIMserver panel. If the user is not logged in yet, four input fields and a login button are displayed (figure 4a). The first field is for the URL address of the running BIMserver. The second and third fields are for the user name and password, respectively, required to log in to the BIMserver to gain access to the projects. The last field is for the address of the location of the bimCityTable, which is needed as a connection between the CityGML buildings and the IFC models. If it is not possible to log in to the BIMserver with the specified credentials, an error message is displayed (Figure 8b). If the login is successful, the input fields disappear and are replaced by a select box and several radio buttons. The login button is also replaced by the "Insert BIM-Model" button. This can be seen in figure 8c. Via the select box it can be chosen whether to select from predefined queries or whether to write a custom query (figure 8d). If a query is defined and a CityGML building is selected, a model can be inserted by clicking the "Insert BIM-Model". In case of an error it will be displayed in the console (section 3.3).

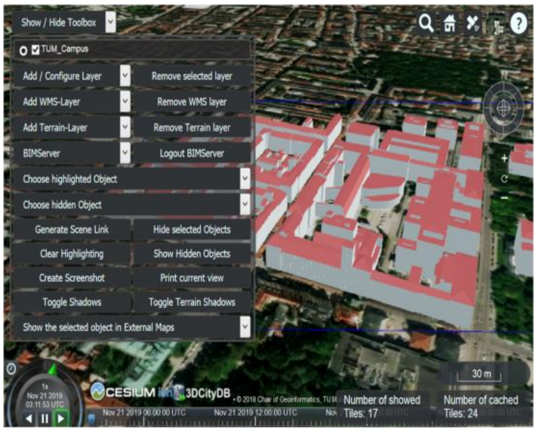
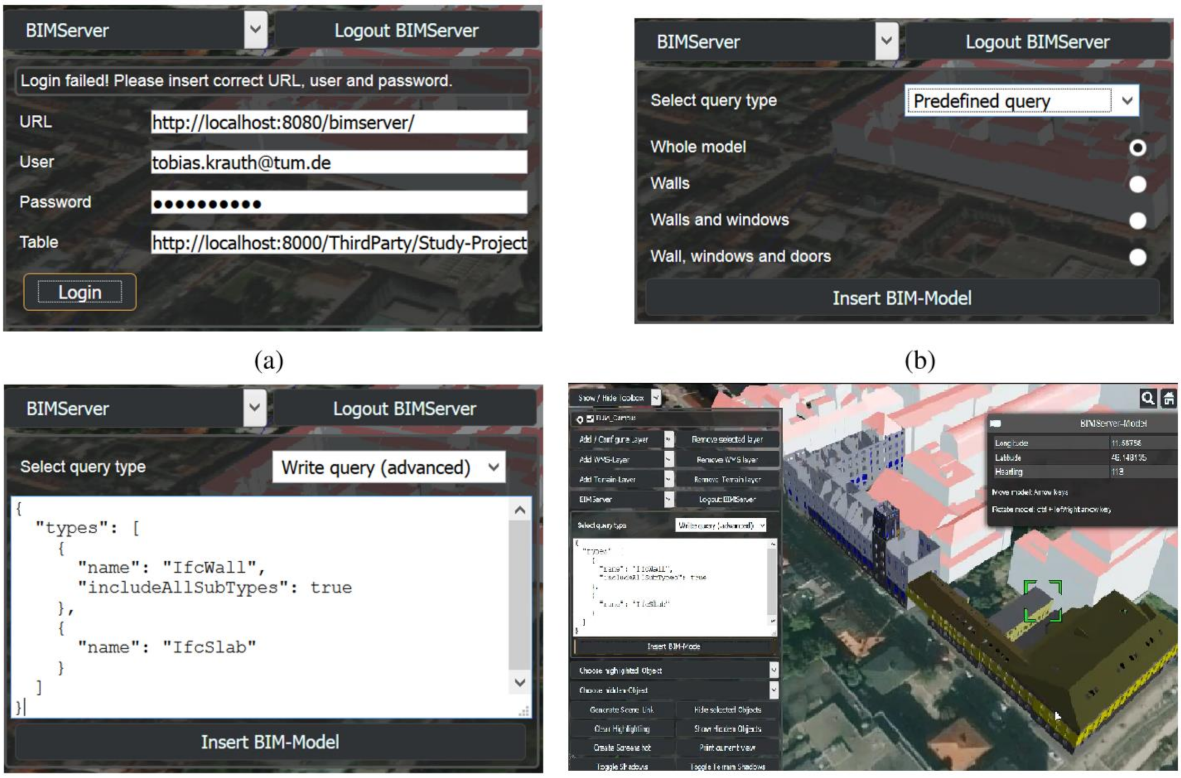


Figure 7: 3D Web client Toolbox with the BIMserver panel.

When clicking on an inserted glTF model, an attribute display is shown in the upper right corner of the screen in the 3D web client, similar to the one appearing when clicking on CityGML buildings (figure 9). The difference, however, is that no



Login failed! Please insert correct URL, user and password.

URL: http://localhost:8080/bimserver/
User: tobias.krauth@tum.de
Password: ●●●●●●●●●●
Table: http://localhost:8000/ThirdParty/Study-Project

(a)

Select query type: Predefined query

Whole model ○
Walls ●
Walls and windows ●
Wall, windows and doors ●

(b)

Select query type: Write query (advanced)

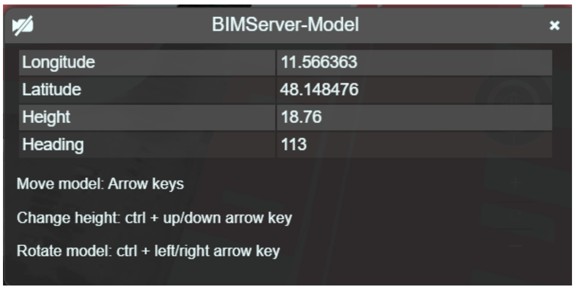
```
{
  "types": [
    {
      "name": "IfcWall",
      "includeAllSubTypes": true
    },
    {
      "name": "IfcSlab"
    }
  ]
}
```

(c)

(d)

Figure 8: Four different states of the BIMserver panel inside the 3D web client, when a login failed (a), after the login with "Predefined query" selected (b), after the login with custom query selected (c), BIM model successfully inserted into city model (d)

information about attributes is displayed, but instead the geodetic coordinate of the point of origin of the glTF model. If the model is moved using the arrow keys (section 3.2), the coordinates shown in the attribute display will also change. In addition, instructions for moving the model are also displayed.



BIMServer-Model

Longitude	11.566363
Latitude	48.148476
Height	18.76
Heading	113

Move model: Arrow keys

Change height: ctrl + up/down arrow key

Rotate model: ctrl + left/right arrow key

Figure 9: Attribute Display of the 3D web client with information about the glTF model position.

4. DISCUSSION AND CONCLUSION

The application described in this paper is a working prototype, which solves the intended task described in the introduction. It is a pragmatic approach to integrate BIM datasets into comprehensive 3D city models. The approach is highly relevant to both AEC and geospatial industry. The specific combination of Open Source tools creates a vehicle to integrate BIM data (IFC) into a user accessible 3D navigation and viewing environment. The prototype demonstrates the potential of using open source software to provide highly interactive 3D GIS applications on the web. CesiumJS and the glTF standard proved to be an easy way to use 3D visualization tools with powerful functionalities. They provide a medium to integrate BIM and GIS for visualization purposes. The source code is available under the github link (TUM-GIS, 2020). Our future research will concentrate on implementing analysis functionalities required by the 3D application. Nevertheless, the prototype 3DCityDB4BIM has some limitations and weaknesses which are discussed in the following. Furthermore, alternative approaches are presented.

4.1 Linking of CityGML buildings with IFC models

The main advantage of the approach for linking the IFC model with the corresponding CityGML buildings described in section 3.1 is that no time-consuming preprocessing of the files is necessary. The user can add new links in just a few steps without the need of an additional application such as the Feature Manipulation Engine (FME). Nevertheless, this only works well if the IFC model is larger than the CityGML buildings and if they have the same footprints. Figure 11 illustrates this problem. In case (a), one IFC model replaces several CityGML buildings. In this case the CityGML buildings are well replaced by the IFC model. In case (b) however, there is a 1:1 relation between IFC model and CityGML building, but the footprint of the CityGML building is larger than the IFC model. If the CityGML building is replaced, there is an unwanted gap between the IFC model and other CityGML buildings. The same happens in case (c), when there is a 1:N relation between the IFC model and the CityGML buildings, but the footprints do not match.
To avoid the problem mentioned above the data could also be linked on IFC entity level. This could be done by storing the GUID of the IFC entities in the external references of the corresponding CityGML buildings. For this, however, the external references would have to be included in the online spreadsheet in order to access the information inside the 3D

web client. This could lead to overload in the 3D web client attribute display due to the vast amount of IFC entities. As an alternative an IfcExternalReference could be created for each IFC entity, storing the GMLID of the corresponding CityGML building. With this it would be possible to use a BIMserver query to get back only those entities that have the GMLID of the selected CityGML building as an IfcExternalReference. The disadvantage of this approach is the complex preprocessing of the source files to find out which IFC entities belong to which CityGML building and to store the GMLIDs in the IFC files. A question that arises in this context is, for example, what to do with IFC entities that overlap with two CityGML buildings. Nevertheless, the linking on IFC entity level has the potential to solve the problem illustrated in figure 11 and therefore should be examined and tested in future works.

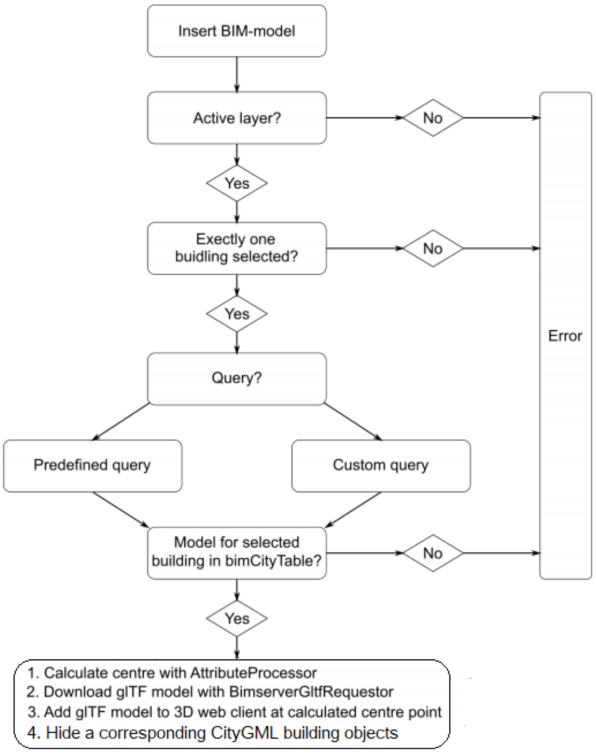

Insert BIM-model
Active layer?
No
Yes
Exectly one buidling selected?
No
Yes
Error
Query?
Predefined query
Custom query
Model for selected building in bimCityTable?
No
Yes
1. Calculate centre with AttributeProcessor
2. Download glTF model with BimserverGltfRequestor
3. Add glTF model to 3D web client at calculated centre point
4. Hide a corresponding CityGML building objects

Figure 10: Sequence of conditions and tasks that are executed by the main function that was added to the script.js. This function is the link between the script.js, the bimCityTable (section 3.1), the AttributeProcessor (section 3.2) and the BimserverGltfRequestor (section 3.3).

4.2 Positioning of the model

In contrast to the calculation of the height, the calculation of the longitude and latitude of the centre point of CityGML buildings does not work optimally. This is illustrated in figure 12. In figure 12a the point of origin of the glTF model is shown. 12b shows the bounding box and the centroid of several related CityGML buildings. The bounding boxes calculated by 3DCityDB are parallel to the axes of the coordinate reference system. This leads to the problem that depending on the rotation and especially the shape of the building, a different bounding box and thus, also a different centre is calculated than the one set as the origin of coordinates by the BIMserver glTF serializer. Therefore, when replacing CityGML building(s), the glTF model is often placed with a small offset. Additionally, it is unknown how much the glTF model has to be rotated to fit in correctly. Since no automated solution could be found for these problems, the user got the possibility to adjust the position and rotation, which is described in section 3.4.

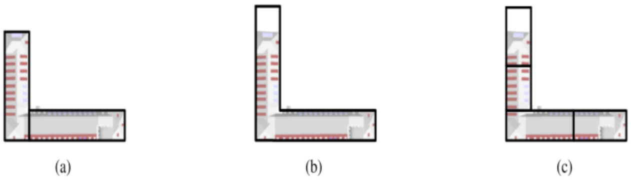


Figure 11: Three examples show CityGML buildings and the IFC model. The colored model in the background is the IFC model, the black boxes represent the CityGML building objects.

Within this work no fully automated positioning of the glTF model, derived from non-georeferenced IFC models, could be implemented. However, if georeferenced IFC models are considered, new possibilities arise. A recently introduced approach for georeferencing IFC models is the Level of Georeferencing (LoGeoRef) concept from Clemen and Hendrik (2019). They propose six levels, with a higher level providing more quality. To place the glTF model, produced by the BIMserver, correctly, LoGeoRef 30 would be necessary. Here, the real world position as well as the rotation of the building is stored in the IFC file. To access this information with the BIMserver JavaScript API the LowLevelInterface method "getDataObjectsByType" could be used. This method allows to get the attributes of a specific class like IfcSite and, thus information about the georeference.

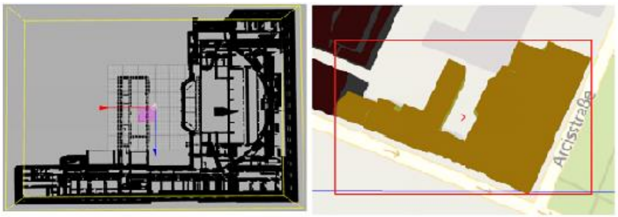


Figure 12: Comparison of the origin of coordinates of a glTF model created by BIMserver (a) and the calculated bounding box centre of the corresponding CityGML buildings within the 3D map client (b)

REFERENCES

Altmaier, A., Kolbe, T.H., 2003. Applications and solutions for interoperable 3D geo-visualization. *Proceedings of the Photogrammetric Week 2003 in Stuttgart*. Wichmann Verlag, 2003.

Beetz, J., van Berlo., L., de Laat., R., van den Helm, P., 2010. Bimserver.org–an open source ifc model server. *Proceedings of the CIB W78 2010: 27th International Conference –Cairo, Egypt, 16-18 November*. Cairo, 2010.

BIMserver 2019. BIMserver Homepage.
http://www.bimserver.org. (04 April 2020).

Buckler, C., 2018. Understandinges6modules.
https://www.sitepoint.com/understanding-es6-modules/ (04 May 2020)

BuildingSmart., 2020.http://www.buildingsmarttech.org/specifications/ifc-releases/summary (28 January 2020).

Clemen, C., Görne, H., 2019: Level of georeferencing (logeoref) using IFC for BIM. *Journal of Geodesy* (10), 15–20.

Deng, Y., Cheng J. C., Anumba C., 2016. Mapping between BIM and 3d GIS in different levels of detail using schema mediation and instance comparison. *Automation in Construction* 67(1), 1–21.

El-Mekawy, M., Östman, A., Hijazi, I., 2012. A Unified Building Model for 3D Urban GIS, *ISPRS International Journal of Geo-Information*, 1(2), 120-145.

Fosu, R., Suprabhas, K., Rathore, Z., Cory, C., 2015. Integration of Building Information Modeling (BIM) and Geographic Information Systems (GIS) - a literature review and future needs. *Proc. of the 32nd CIB W78 Conference*, Eindhoven, Netherlands.

Gröger., G., Kolbe, T.H., Nagel., C., Häfele., K.-H., 2012. OGC City Geography Markup Language (CityGML) Encoding Standard, OGC 12-019.

Harrison., D., 2009. BIMserver and the potential of serverside BIM. http://www.stress-free.co.nz/ bimserver_and_the_potential_of_serverside_bim. (06 November 2019).

Hijazi, I., Donaubauer, A., 2017. Integration of building and urban information modeling-opportunities and integration approaches. *Geoinformationssysteme 2017* – Beiträge zur 4. Münchner GI-Runde. Wichmann Verlag.

Irizarry, J., Karan., E. P., Jalaei., F., 2013. Integrating BIM and GIS to improve the visual monitoring of construction supply chain management. *Automation in Construction*, 31(1), 241–254.

Isikdag, U., 2006. Towards the Implementation of Building INformation Models in Geospatial Context, United Kingdom: University of Salford. Phd thesis.

Isikdag, U., and Zlatanova, S., 2009. A SWOT analysis on the implementation of Building Information Models within the Geospatial Environment. *Urban and Regional Data Management.* 27-42. CRC Press.

Lapierre, A., Cote., P., 2007. Using Open Web Services for urban data management: A testbed resulting from an OGC initiative for offering standard CAD / GIS / BIM services. *Urban and Regional Data Management*, London, Taylor & Francis, 381-393.

Niu, S., Pan, W., Zhao, Y., 2015. A BIM-GIS integrated web-based visualization system for low energy building design. *Procedia Engineering*, 121, 2184–2192.

Ohori, K.A., Biljecki, F., Diakité, A., Krijnen, T., Ledoux, H., Stoter, J., 2017. Towards an integration of GIS and BIM data: what are the geometric and topological issues?. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., IV-4/W5, 1–8, https://doi.org/10.5194/isprs-annals-IV-4-W5-1-2017, 2017.

Song, Y., Wang, X., Tan, Y., Wu, P., Sutrisna, M., Cheng, J. C. P., Hampson, K., 2017. Trends and Opportunities of BIM-GIS Integration in the Architecture, Engineering and Construction Industry: A Review from a Spatio-Temporal Statistical Perspective, *International Journal of Geo-Information*. 6(12), 397-429.

The Open Source Geospatial Foundation, https://www.osgeo.org/. (10 January. 2020)

TUM-GIS 2020, https://github.com/tum-gis/3DCityDB4BIM, (04 May 2020)

wbdg.org, 2020. https://www.wbdg.org/building-information-modeling-bim. (4 May 2020).

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., Kolbe, T.H., 2018. 3DCityDB - a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML, *Open geospatial data, software and standards*. 3(5), 1-26 https://doi.org/10.1186/s40965-018-0046-7

Yao, Z., Nguyen, S., Sindram, M., Chaturvedi, K., Kolbe, T.H., 2018. 3D City Database for CityGML - A Hands-on Tutorial for Beginners, version 4.0.0 edition. https://github.com/3dcitydb/tutorials/blob/master/3DCityDB_V4.0_Hands-On_Tutorial.pdf. (04 May 2020).