

Article

A Unified Database Solution to Process BIM and GIS Data

Michał Wyszomirski * and Dariusz Gotlib *

Department of Cartography, Faculty of Geodesy and Cartography, Warsaw University of Technology,
Plac Politechniki 1/329, 00-661 Warszawa, Poland

* Correspondence: michal.wyszomirski@pw.edu.pl (M.W.); dariusz.gotlib@pw.edu.pl (D.G.)

Received: 6 November 2020; Accepted: 26 November 2020; Published: 28 November 2020



Abstract: For many years, the objective of spatial databases created using Geographic Information System (GIS) technology was to provide information about large spaces and areas outside of buildings. Building Information Modeling (BIM) technology focused mainly on small spaces, indoor and outdoor, targeted at other users and slightly different applications, was developed simultaneously for several dozen years. The significant development of GIS technology and new tools for quick data acquisition (e.g., laser scanning) and growing user needs resulted in the geoinformation modeling of the space inside buildings as well. BIM, on the other hand, began to be used for increasingly larger spaces outside buildings. Technology developers, users, and scientists started to notice that works turned out to be repetitive and that combining two different technologies is necessary; however, it is not simple. The research presented in the article is another attempt at connecting the world of BIM and GIS. The proposed integrated database environment of BIM/GIS spatial data makes it possible to store GIS and BIM data, enabling the use of the same data by both types of systems simultaneously and in a consistent manner. This allows BIM systems to obtain simultaneous access to BIM and GIS data, which may be needed in, for example, the process of analyzing a building and its immediate surroundings. At the same time, GIS can obtain up-to-date building data necessary for spatial analyses, building management, or route mapping in navigation applications. The concept proposed in this article assumes a pragmatic approach, which is based on sharing Industry Foundation Classes (IFC) and CityGML schemas from a single database for BIM and GIS applications in their practically original form using an additional integrated BIM-GIS schema, called BIGI-S. The research joins some other works in this field, complementing them and adding a new perspective. This paper describes the concept of this solution, including specific data structures, data conversion algorithms, and a prototype solution. The tests carried out by the authors prove the robustness of the adopted concept and its technical feasibility.

Keywords: GIS; BIM; CityGML; IFC; indoor modeling; spatial modeling; building information

1. Introduction

Intensive scientific research on the integration of databases created within GIS (Geographic Information System) and BIM (Building Information Modeling) systems has only been conducted over the last 10 years. This is a relatively short period, considering that GIS systems date back to the 1960s and BIM systems to the beginning of the 1980s. However, in the case of BIM, the terms “building information model” and “building information modeling” (including the BIM acronym) were started to be widely used in the 1990s [1]. In 2002, Autodesk Inc. published a document titled “Building Information Modeling” [2], which popularized the idea of BIM.

For many years, the objective of spatial databases created using GIS technology was to provide information about the space outside of buildings. Significant development of such technology

at the beginning of the 21st century, new tools of quick data acquisition (e.g., laser scanning), and growing user needs resulted in the modeling of the space inside buildings as well. Elaborated standards, such as the Building Interior Space Data Model (BISDM) and, later, CityGML, facilitated and encouraged such procedures. BIM technology, targeted at other users and slightly different applications, was developing simultaneously. Technology developers, users, and scientists started to notice, however, that works turned out to be repetitive and that combining two different technologies was not simple. Many applications require continuous (in a spatial sense) access to information inside and outside of buildings (e.g., navigation applications). The significance of this need is evident, among others, in the agreement between the key companies of both branches (i.e., ESRI and Autodesk announcement on 15 November 2017 [3]) and establishing a workgroup on the ISO 19166 standard of BIM–GIS conceptual mapping (B2GM) [4].

From many years of practice in BIM and GIS, the authors conclude that there is a need to display and analyze data from BIM resources in GIS systems without the need to import such data repeatedly after each update of the BIM dataset. The authors' motivation was to find a solution that makes it easier to handle BIM and GIS data in a uniform environment. Developing a common data model and one format is difficult and will take some time, so this solution is dedicated to software developers, mainly for an interim period until a common standard for BIM–GIS has been developed and accepted. However, even then, there will still be a need to use previously created data resources in their original form. The existing tools and proposals presented in the literature did not solve our problem. Looking for alternative solutions, the authors developed a concept and then conducted a study that showed the possibility to implement it. This article shows the results of research in the area of connecting the world of BIM and GIS. It presents the authors' concept of using a single database for BIM and GIS systems while preserving the IFC (Industry Foundation Classes) and CityGML source data models typical for them. The research joins some other works in this field, complementing them and adding a new perspective. To prove the concept, practical tests were performed that confirmed the correctness of the assumptions and the viability of the procedure. This research was, among others, a base for a doctoral thesis [5], a project commissioned by the industry, and it was used for the development of an information system on real estate for one of the big universities.

2. Materials and Methods

The basis of the described research was an in-depth analysis of the literature, an analysis of the available BIM and GIS technologies in the field of building modeling, an analysis of the BIM/GIS market, and an organized discussion on the authors' experiences related to implementation works carried out for many years. The new conceptual approach was then formulated, and technological experiments and practical tests were performed. As a result, conclusions were drawn, and final solutions were proposed.

Since 2007, a growing number of published research activities on data integration and exchange between BIM and GIS have been observed. Examples from 2007 include the authors of [6–8]. In the beginning, most of the proposed solutions were based on data migration using IFC and CityGML standards (the first official version in 2008). In the following years, proposals and research results were presented (i.e., in [9–14]). Attempts to unify data types in both systems were made [15] in an attempt to facilitate their cooperation, since the structure of data referring to the same class of objects in both models is incompatible. Meanwhile, there were efforts to use GIS 4D technology for construction design [16]. GIS models (e.g., CityGML) were also used to import BIM data from the models of buildings available in KML (Keyhole Markup Language) format [17]. An interesting classification of the method of integration of BIM and GIS into five types was presented in [18], the first three of which refer to the integration at the data level:

1. Conversion of BIM data into GIS based on the conversion of the IFC model into the CityGML model. As a result, BIM and GIS data are stored in the CityGML model and managed from the level of GIS software. This solution is often used by authors and users of GIS systems who

require detailed data concerning a building. This issue has been discussed in numerous papers (e.g., [10,14,19–25]).

2. Conversion of GIS data into BIM based on the conversion of the CityGML model into the IFC model. As a result, BIM and GIS data are stored according to the IFC model and managed from the level of BIM software. This solution is mostly used in cases when architects or people who manage a building using BIM software require data concerning the building's surroundings [17,26].
3. The unified data model. Solutions in this category concentrate on developing a separate unified model to combine BIM and GIS data for selected analyses and particular applications [15,27,28]. Moreover, the Unified Building Model was proposed to provide two-directional conversion of BIM–GIS data [29–32].

The next two approaches to integrate BIM and GIS data are at the application level:

4. Integration of BIM and GIS at the level of the application server. Solutions in this category offer a new concept of approaching the integration of BIM and GIS by introducing new IT (Information Technology) tools and extending existing applications so that it is possible to jointly operate BIM and GIS data. This is usually realized through the GIS application server [11,33–38].
5. Integration at the level of the client application. Solutions in this category depend on transformations of BIM and GIS data downloaded by independent client applications such as GIS or FM (Facility Management) [6,7,39].

This type of classification was analyzed from a slightly different, although generally similar, perspective in [40]. The authors proposed a division into three integration modes: “BIM leads and GIS supports”, “GIS leads and BIM supports”, and “BIM and GIS are equally involved”. In recent years, more and more research has indicated interest in the use of combined BIM and GIS data (e.g., [41–44]).

An analysis of the literature shows that the issues of BIM and GIS integration are in the area of scientific interest for many research centers, and thus far, there is no single unified approach for solving the problem. The search for more optimal solutions is still needed. Data exchange between GIS and BIM systems is becoming more advanced and easier to use, but the level of actual integration of both solutions can be considered as low. Solving the problem by data conversion may be sufficient only for some applications. Therefore, the authors undertook research to find solutions that would allow a greater integration of both systems. Among the many potential options, the possibility of BIM and GIS applications in a joint database was analyzed to an extent not previously present in the literature. However, it should be noted that the idea of using the database environment as an intermediary layer between BIM and specialized information systems appears in various contexts and applications, e.g., in [45]).

After analyzing the literature and performing a series of analyses of the available BIM and GIS models and experiments involving data combinations, the concept of BIM and GIS integration was proposed, involving the use of one integrated spatial database but without the need to create a new data schema and only through the appropriate use of existing IFC and CityGML schemas. The proposed integration is released primarily at the technological level. The authors focused on finding a solution that facilitates the operation of BIM and GIS data in a uniform environment. Integration in this context means providing easier than before procedures for accessing BIM data from the GIS software level and vice versa. Moreover, guidelines on how to transform a selected data element from IFC to the corresponding elements in CityGML were developed. The guidelines cover a procedure for geometrically transforming IFC data into a GIS-readable form, including a procedure for changing the local coordinate systems used in BIM–GIS. A full description of these issues, however, is too voluminous and goes beyond the possibility of presentation in one article.

To verify the proposed concept, the following procedure was adopted:

1. Elaboration of a relational spatial database project with a structure based on the IFC and CityGML data models (two schemas implemented in a single database).

2. Creation and testing of automated procedures for importing IFC data into the relational database.
3. Experimental transfer of six IFC models into the created relational database.
4. Evaluation of the adopted procedure.
5. Improvement of the procedure.
6. Checking access to the created database from BIM and GIS software.

In the experimental works, BIM models were used, derived from the management systems of the building technical documentation used by the institutions of public administrations and commercial companies in Sweden. The models were selected in such a way as to represent different types of objects and the different levels of complexity of the model.

3. Results

3.1. *The Concept of an Integrated BIM/GIS Database Environment*

As discussed in Section 2, many attempts were made to exchange data between BIM and GIS, as well as to create a common database based on a unified data model. The use of the new common data model by BIM and GIS requires the introduction of quite radical changes in the applications available on the market; therefore, the implementation of this solution will likely take many years. The concept proposed in this article assumes a pragmatic approach, which minimizes this shortcoming by using IFC and CityGML schemas in their practically original forms in a single database, later referred to as the Integrated BIM–GIS Database (BIGI-D). The integrated database does not mean in this case a single consistent database in the sense of a conceptual model but a coherent technological environment that allows, depending on the needs, the use of data from both resources by various applications and the improvement of the methods of full information exchange. Thus, the proposed method does not solve the problem of describing the same objects in the BIM and GIS by classes with different structures. On the contrary, it assumes that the conceptual (structured) discrepancies will still exist for some time and that solutions should be sought that accept this situation and improve the coexistence of BIM and GIS until fundamental changes are introduced. At the stage of implementation, BIM data saved in accordance with the IFC conceptual model are processed using tables, perspectives, and stored procedures in the IFC schema. The term schema is used here in two ways—as a data structure and as a place to store tables and procedures in the database. In the proposed technological approach, the use of stored procedures is essential. A stored procedure is a set of SQL (Structured Query Language) statements with its name, which are stored in a database management system as a block. A stored procedure is a subroutine available to applications that access a relational database management system. GIS data recorded according to the CityGML conceptual model are processed using tables, perspectives, and stored procedures in the CityGML schema. In addition, it is proposed to add the BIM–GIS Schema (BIGI-S) for joint storage of the data from IFC and CityGML to facilitate conversion between BIM and GIS and data sharing. All three schemes are implemented in one database.

A model based on dynamic data conversion is an alternative implementation solution. Within this approach, the data are only stored in the BIGI-S. Schemas dedicated to individual BIM and GIS applications (as well as others) are implemented as database views and stored procedures. The schema dedicated for the BIM application is generated on the fly, based on stored procedures that convert data from the BIGI-S. The case with the CityGML schema is similar. The main advantage of this solution is the elimination of data redundancy. Each object is saved in the database once but can be seen using CityGML and IFC views. From the point of view of BIM and GIS applications, the common BIGI-S is, in this case, completely invisible. Each of these applications read data using the appropriate IFC or CityGML model. The weakness of the above solution is the need to create complicated database perspectives that realize the translation of selected data between schemas, in the case of spatial data conversion to IFC, while requiring very fast access to these data. Despite the necessity of creating complex perspectives and stored procedures, the authors claim that avoiding redundancy has significant value; therefore, this solution was selected for the prototype. However, such a solution

may adversely affect the efficiency of processing. Therefore, preliminary studies on the effectiveness of using stored procedures for model conversion were performed.

The architecture of the proposed database includes the possibility of creating a number of additional schemas (e.g., a schema for facility management systems) that use both BIM and GIS data. Each of these schemas would allow access to integrated BIM/GIS data from the level of dedicated applications. This approach is easy to implement and allows to minimize the changes needed in the software currently available on the market. Figure 1 presents a diagram of the BIGI-D idea in a configuration that allows consistent use of data by GIS and BIM systems. In the presented concept, it was assumed that data are stored in BIGI-D using the types of geometric data that are standard in GIS systems. Later on in the text, the idea presented in Figure 1 is discussed in detail.

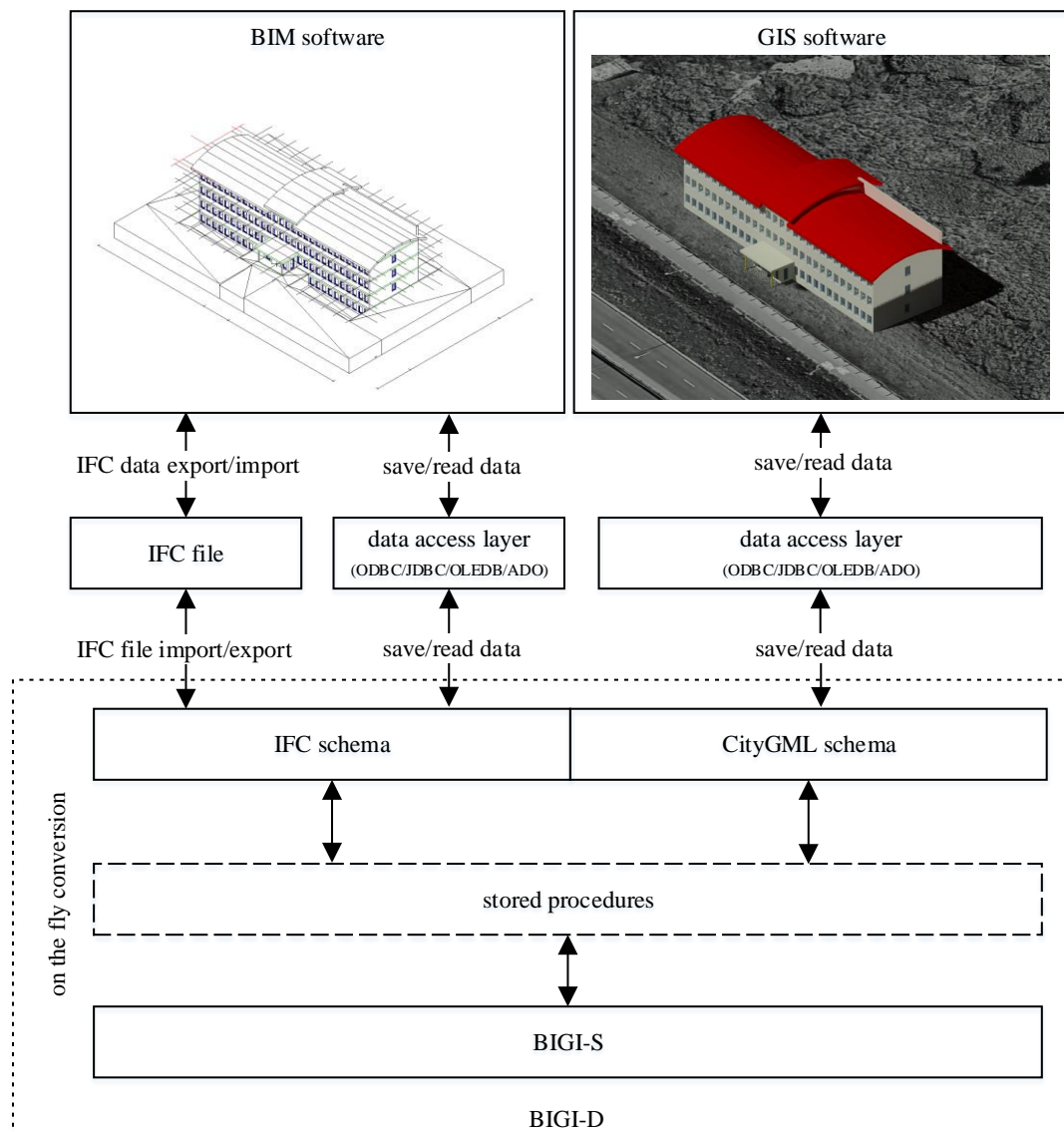


Figure 1. The process of simultaneous use by Building Information Modeling (BIM) and Geographic Information System (GIS) applications of a shared database (i.e., the BIM–GIS Database (BIGI-D)) through the BIM–GIS Schema (BIGI-S), allowing the conversion of source BIM and GIS data on the fly thanks to stored procedures. IFC: Industry Foundation Classes.

In practice, for a selected building, there may exist three main cases in which the source building model is developed in GIS only, in BIM only, or in both GIS and BIM. In the discussed concept, it was assumed that building models should be converted to an integrated database from the CityGML

schema only if there is no source BIM model available for them. Due to the high detail of the BIM model and the fact that the BIM process built around this model provides continuous data updates, it was assumed that the BIM model is the most desirable and complete source of building data.

The following section of the paper describes the experiments that attempted to evaluate the possibilities of implementing the presented concept. They are mostly based on implementing the IFC model into a relational database in a way consistent with the CityGML model and, in the next step, on performing test data imports.

3.2. The Case Study

The first stage of working with the BIGI-D database is feeding it with data. In the case of data from BIM applications, this process involves exporting the data as an IFC file and saving them in the database schema dedicated to the IFC data. This database schema is called IFC, and it contains tables and stored procedures that allow operating on each of the tables. The procedures perform write, read, update, and delete data. After saving the entire model, the data conversion process from IFC to BIGI-S is executed. Data is converted by stored procedures to tables in BIGI-S. When the process is complete, the data in the IFC schema is deleted. This avoids data redundancy.

In the next stage, BIM and GIS applications can work with building models saved in BIGI-D. The BIM application triggers the process of transferring the data for the selected building model from BIGI-S to the IFC database schema. This allows the BIM application to work on this model. Upon completion of the work, the model is converted back to BIGI-S and removed from the IFC schema. At the same time, the GIS system can use data of the same building. In this situation, the data is converted from BIGI-S to the CityGML database schema and is available for the GIS system. If the BIM application works on the same model at the same time, the data stored in the IFC database schema is converted to BIGI-S, the trigger in the database starts the process of updating the data in the CityGML database schema, and the GIS system receives the current data about the object. Similarly, building data saved by the GIS system is automatically available for the BIM application. This is where the data conflict problem arises when both BIM applications and GIS systems attempt to save building data. If the data on the interior of the building is obtained only from the IFC model, the data for the entire model is saved in BIGI-S. Similarly, in the case of obtaining data about the building only from CityGML. If the building data is obtained simultaneously from BIM and GIS, the geometric and descriptive data are taken from IFC and supplemented with the descriptive data from CityGML. This is because the IFC model is much more detailed than CityGML and is a better source of data about the facility. CityGML, on the other hand, can provide descriptive data not available in the IFC model. It was assumed that each building acquired into the database would have a globally unique identifier (GUID). The same building in the IFC and CityGML models will have the same GUID. This will allow BIGI-D to identify buildings in the process of obtaining data. The authors did not conduct further research on this problem.

In this operating mode, data redundancy occurs. The entire building model is saved in BIGI-S; if the BIM application works on it, it is also converted to the IFC database schema. If it is additionally used by a GIS system, the same model is converted to the CityGML database schema. Stored procedures and triggers ensure data consistency, but redundancy is threefold in this situation. This corresponds to the functionality of the version management system—the model is downloaded to the IFC schema to perform editing work (check-out), and after completion, it is loaded into the main repository—BIGI-S (check-in). Similarly, when working with a GIS model. The data is copied (with conversion) to the CityGML schema and edited there. After completing work on the model, the data is transferred to BIGI-S as a new version of the object. The subject of versioning object models in BIGI-D is beyond the scope of this article.

The solution to the problem of redundancy and the need to maintain the consistency of three copies of data is on-the-fly operation. In this case, the IFC and CityGML database schemas do not contain tables but only stored procedures. The process of writing data in the IFC database schema

consists of directly converting IFC data to BIGI-S. Reading the data also involves taking the source data from BIGI-S and converting it to IFC format without using intermediate tables in the IFC database schema. The process of working with data by GIS applications looks similar. Data is converted from and to BIGI-S by stored procedures in the CityGML database schema.

3.3. Implementation of the IFC Model in a Relational Database

The first step of the experiment was to create a database structure corresponding to the IFC model. The database model corresponds exactly to the IFC model, which makes it possible to avoid numerous complications in the processes of data writing and reading by application software. It should be remembered that the procedure of processing the data stored in IFC into a relational model is quite complex. The IFC model is not relational and has a number of quite specific features (e.g., multiple local coordinate systems and complex internal references between objects). In order to enable editing operations on the BIM data saved in a relational database, the mechanism of stored procedures was applied. For each class of IFC objects, four stored procedures were prepared for write, read, update, and delete operations. For the purpose of the experiments, the stored procedures enabling writing, reading, modification, and deletion of the data of IFC objects were prepared for more than 100 classes of IFC objects. The procedures were implemented in SQL and PL/pgSQL (Procedural Language/PostgreSQL) languages on the PostgreSQL platform. For comparison, stored procedures for several selected classes of IFC objects in PL/SQL (Procedural Language for SQL) in the Oracle database and T-SQL (Transact SQL) in the Microsoft SQL Server database were also prepared. The purpose of this part of the experiment was to check whether it was possible to create stored procedures that could be triggered in a similar way regardless of which database platform was used. As a result of the experiment, it was confirmed that the triggering of the prepared stored procedures operating on tables in PostgreSQL, Oracle, and SQL Server databases was very similar. Thus, the proposed solution can be used with any of the above relational database management systems.

3.4. Automated Process to Import IFC Data into a Relational Database

The general schema of the data import operation is presented in Figure 2. The process of rewriting data from an IFC file to an IFC schema in the database consisted of three steps: reading data from the IFC file, saving the data in a form readable for the database management system (creating SQL scripts), and loading the data into the database by executing scripts in the database. The data transfer operation is, therefore, a multistep task consisting of the export of data from the BIM program into the IFC format, followed by the import of data from the IFC file to the BIGI-S in the integrated BIM/GIS spatial database. The course of the experiment is shown in Figure 2.



Figure 2. General scheme of importing IFC data into the database. SQL: Structured Query Language.

3.5. Implementation of the Common Database Environment for IFC and CityGML Data

The architecture of the proposed solutions assumes the use of several schemas in a single database. Separate IFC and CityGML schemas are dedicated to storing BIM and GIS data, respectively. GIS applications only use data via the CityGML schema, while BIM applications can access data via IFC schema. IFC and CityGML schemas are treated as temporary datasets. The data stored in the IFC schema is converted into a BIGI-S, which allows for lossless saving of the data from the IFC model. Similarly, the data saved by GIS applications in the CityGML schema are also converted into the BIGI-S. It should be emphasized here that the BIGI-S does not mean, in this case, a completely new conceptual data model for the existing BIM and GIS data but, above all, refers to appropriately related IFC and

CityGML source schemas. The integrated data model is materialized in the form of a set of tables and stored procedures that are located in the BIGI-S. The differences in the GIS and BIM data conceptual models were taken into account when building the integrated data model. Table 1 shows selected examples, as well as differences between the methods of storing building data in the CityGML and IFC models. It requires a number of data transformations to convert data between these models via an intermediate model but is, with some simplifications, possible. The stored procedures implemented in the BIGI-S allow to save and read data between the CityGML and IFC models and the integrated model. The presented concept assumes that the data are stored in the BIGI-S in common tables with the use of the geometric data types typically used in GIS systems.

Table 1. An example of the process of obtaining data for the CityGML model from the Industry Foundation Classes (IFC) model.

CityGML <i>AbstractBuildingType</i> Class	Multiple IFC Attributes and Classes
<i>class</i>	The class of an object can be determined from the data contained in the attribute sets assigned to the object.
<i>function</i>	The function of an object can be determined from the data contained in the attribute sets assigned to the object.
<i>usage</i>	<i>OccupancyType</i> attribute from the <i>Pset_BuildingCommon</i> property set attached to the building.
<i>yearOfConstruction</i>	<i>YearOfConstruction</i> attribute from the <i>Pset_BuildingCommon</i> property set attached to the building.
<i>yearOfDemolition</i>	The date of decommissioning the building is determined by analyzing the data contained in the property sets assigned to the object if such information is present in the building model.
<i>roofType</i>	<i>ObjectType</i> attribute of the <i>IfcRoof</i> object class assigned to the <i>IfcBuilding</i> object.
<i>measuredHeight</i>	Determining the height of the building takes place through the analysis of geometric objects indicated by the attributes <i>Representation</i> and <i>ObjectPlacement</i> of the <i>IfcBuilding</i> object.
<i>storeysAboveGround, storeysBelowGround</i>	<i>NumberOfStoreys</i> attribute from the <i>Pset_BuildingCommon</i> property set attached to the building or the number of <i>IfcBuildingStorey</i> objects attached to the building. The determination of the number of stories above and below the ground level is made by analyzing the attributes <i>ObjectPlacement</i> of each <i>IfcBuildingStorey</i> object attached to the building.
<i>storeyHeightsAboveGround, storeyHeightsBelowGround</i>	The determination of the height of the stories above and below the ground level is made by analyzing the <i>Representation</i> attributes of the <i>IfcBuildingStorey</i> objects.
<i>lod0FootPrint</i>	The outline of the building on the ground is determined by analyzing the geometric objects indicated by the <i>Representation</i> attribute of the <i>IfcBuilding</i> object.
<i>lod0RoofEdge</i>	The roof edge is determined by analyzing the geometric objects indicated by the <i>Representation</i> attribute of the <i>IfcRoof</i> object assigned to <i>IfcBuilding</i> .
<i>lod1Solid, lod1MultiSurface, lod1TerrainIntersection</i>	The geometry of the building body and its contact with the ground are determined by analyzing the geometric features of the objects indicated by the <i>Representation</i> and <i>ObjectPlacement</i> attributes of the building.
<i>lod2Solid, lod2MultiSurface, lod2MultiCurve, lod2TerrainIntersection</i>	The geometry of the building body and its contact with the ground are determined by analyzing the geometric features of the objects indicated by the <i>Representation</i> and <i>ObjectPlacement</i> attributes of the building.
<i>outerBuildingInstallation</i>	The list of objects related to the building outside of the building body can be obtained on the basis of the relationships that aggregate these objects with the building or <i>IfcSite</i> object.
<i>interiorBuildingInstallation</i>	The list of building-related objects inside the building body can be obtained on the basis of the <i>IfcRelContained</i> aggregating relationships.
<i>boundedBy</i>	The building outline can be obtained from the <i>IfcPolyLoop</i> object defining the building outline.
<i>lod3Solid, lod3MultiSurface, lod3MultiCurve, lod3TerrainIntersection</i>	The geometry of the building body and its contact with the ground are determined by analyzing the geometric features of the objects indicated by the <i>Representation</i> and <i>ObjectPlacement</i> attributes of the building.
<i>lod4Solid, lod4MultiSurface, lod4MultiCurve, lod4TerrainIntersection</i>	The geometry of the building body and its contact with the ground are determined by analyzing the geometric features of the objects indicated by the <i>Representation</i> and <i>ObjectPlacement</i> attributes of the building.
<i>interiorRoom</i>	The list of rooms can be obtained by <i>IfcRelAggregates</i> aggregation.
<i>consistsOfBuildingPart</i>	The list of objects inside the building can be obtained by the <i>IfcRelAggregates</i> and <i>IfcRelContained</i> aggregations.
<i>address</i>	The <i>IfcPostalAddress</i> object related to the <i>IfcBuilding</i> object.

Stored procedures in the BIGI-S allow on-the-fly conversion of data from the IFC schema to the BIGI-S and vice versa, as well as bidirectional data conversion between the CityGML schema and the BIGI-S according to the analysis of both data models briefly presented in Table 1. For objects of the classes inherited from *IfcProduct* class, the process of converting data from the IFC schema to the BIGI-S includes several steps: rewriting the basic attributes of the object, converting the geometry of the object to the global coordinate system, taking into account the rotation of the coordinate system according to the TrueNorth attribute, saving the individual attributes assigned to the object, and saving the sets of attributes assigned to the object (Figure 3).

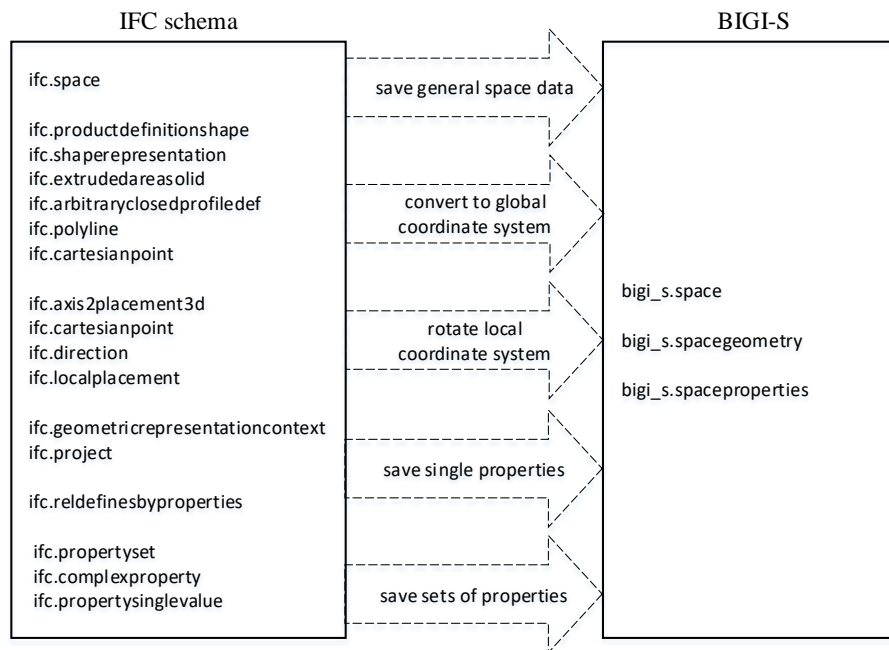


Figure 3. General scheme for transforming data from the IFC schema to the BIGI-S based on the example of an *IfcSpace* object.

The general algorithm of the transformation process of object data from the IFC schema to BIGI-S for each object class inherited from the *IfcObjectDefinition* class is the same. For example, for an *IfcSpace* object, this process is as follows:

1. Attributes *GlobalId*, *Name*, and *Description* are read from object's table.
2. These attributes are saved in the *bigi_s.space* table, along with the project GUID.
3. The data about author, date, and BIM application used to create the processed object in the IFC model is read from the *IfcOwnerHistory* class object indicated by the *OwnerHistory* attribute.
4. The data about the source of the object data is saved in the *bigi_s.space* table.
5. The data about relations between the processed object and other objects of the *IfcObjectDefinition* class are read from the source tables.
6. Data about these dependencies are saved in a table in the BIGI-S schema.
7. Based on the *IfcPropertySet*, *IfcComplexProperty*, and *IfcPropertySingleValue* objects assigned to an *IfcSpace* object through the *IfcRelDefinesByProperties* object, a list of single properties and property sets assigned to this object is determined.
8. Data about *IfcRelDefinesByProperties*, *IfcPropertySet*, *IfcComplexProperty*, and *IfcPropertySingleValue* class objects is stored in the *bigi_s.spaceproperties* table for reverse conversion to IFC.
9. Single properties retrieved from *IfcPropertySingleValue* objects assigned by *IfcRelDefinesByProperties* are saved in the *bigi_s.spaceproperties* table.

10. The property sets retrieved from the `IfcPropertySet`, `IfcComplexProperty`, and `IfcPropertySingleValue` objects are saved in the `bigi_s.spaceproperties` table.
11. On the basis of the `IfcProductDefinitionShape`, `IfcShapeRepresentation`, `IfcExtrudedAreaSolid`, `IfcArbitraryClosedProfileDef`, `IfcPolyline`, `IfcCartesianPoint` class objects, the geometry of the object is determined.
12. The data of the `IfcProductDefinitionShape`, `IfcShapeRepresentation`, `IfcExtrudedAreaSolid`, `IfcArbitraryClosedProfileDef`, `IfcPolyline`, and `IfcCartesianPoint` class objects are stored in the `bigi_s.spacegeometry` table in BIGI-S to enable reverse conversion to IFC.
13. Based on the `IfcAxis2Placement3d`, `IfcCartesianPoint`, `IfcDirection`, and `IfcLocalPlacement` classes, the location of the local coordinate system of the room in the local coordinate system of the `IfcSite` class object is determined. When processing data on the location of the local coordinate system of an object, dependencies of local coordinate systems are taken into account. The geometric elements representing the room are expressed in the local coordinate system of the room, the position of which is determined in the local coordinate system of the floor and that in the local coordinate system of the building, which was expressed in the local coordinate system of the site.
14. Data of the `IfcAxis2Placement3d`, `IfcCartesianPoint`, `IfcDirection` and `IfcLocalPlacement` objects are stored in the table in the BIGI-S schema to enable the reverse conversion to IFC.
15. The location of the local coordinate system of an `IfcSite` object is specified in the global geodetic coordinate system through the `RefLatitude`, `RefLongitude`, and `RefElevation` attributes. The rotation angle of the local coordinate system relative to the geodetic system (TrueNorth) is determined by the `IfcGeometricRepresentationContext` object referenced from the `IfcProject` object.
16. The geometry of the object expressed in the global coordinate system is stored in the geometry attribute of the `bigi_s.spacegeometry` table.

The reverse conversion involves creating, on the basis of the data stored in the tables in the BIGI-S, sets of attributes assigned to the object, single attributes assigned to the object, and geometry in the local coordinate system of the object obtained from the geometry expressed in the global system, taking into account all local coordinate systems of the host objects. The last step is to create the object itself with its basic attributes (Figure 4).

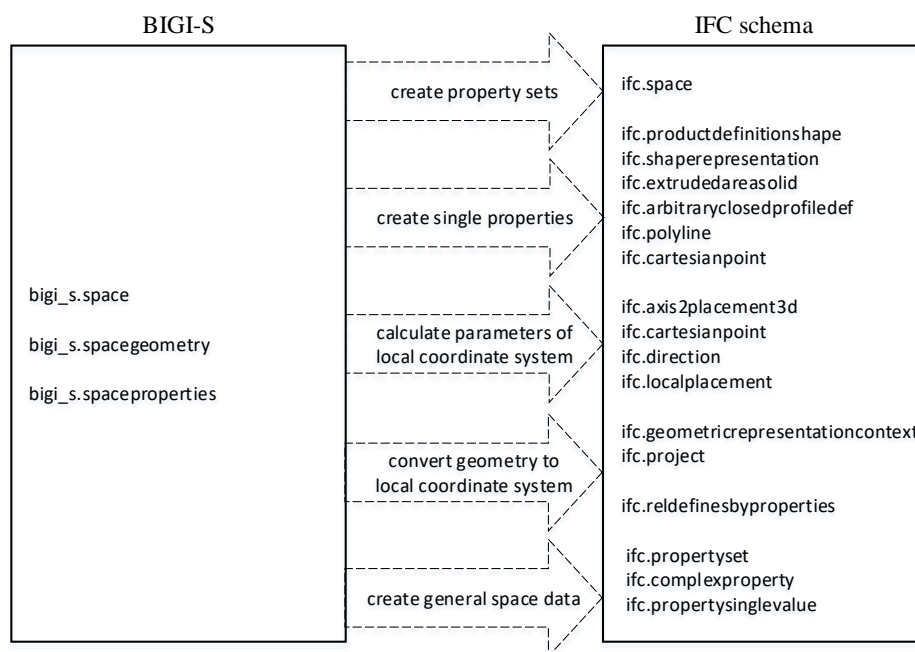


Figure 4. General scheme for transforming data from the BIGI-S to the IFC schema based on the example of an `IfcSpace` object.

The general algorithm of the transformation process of the “room” class from the BIGI-S to the IFC schema is as follows:

1. Attributes GlobalId, Name, and Description are read from the `bigi_s.space` table.
2. These attributes are saved in the `ifc.space` table in the IFC schema.
3. The data about author, date, and BIM application data used to create the processed object originally derived from the IFC model is read from the `bigi_s.space` table.
4. The data about the source of the object data is saved in the `ifc.ownerhistory` table.
5. The data about relations between the processed object and other objects derived from the `IfcObjectDefinition` class is read from the table in BIGI-S.
6. This data about these dependencies is saved in the `ifc.relaggregates` table.
7. Room properties are read from the `bigi_s.spaceproperties` table.
8. Simple properties are saved in the `ifc.propertyinglevalue` table.
9. Property sets data is saved in the `ifc.complexproperty` and `ifc.propertyset` tables.
10. Data of the `IfcRelDefinesByProperties` object aggregating the `IfcPropertySet` objects with the processed object is saved in the `ifc.reldefinesbyproperties` table.
11. The geometric representation of the room is read from the `bigi_s.spacegeometry` table.
12. Transformation of coordinates from the global geodetic coordinate system to the local coordinate system of the project is performed:
 - a. The angle between the project North direction and the True North direction for the selected project (`TrueNorth`) is read from the `bigi_s.project` table.
 - b. The coordinates of all points describing the geometry of the room are read from the `bigi_s.spacegeometry` table.
 - c. The transformation of coordinates of points by rotating the coordinate system by the angle that `TrueNorth` is performed.
 - d. The point with the smallest x and y coordinates is selected as the starting point of the local coordinate system of the object.
 - e. Transformation of the point coordinate system is performed by moving the coordinate system by the vector $w = (-x, -y)$, where x and y are the coordinates of the selected point as the starting point of the local coordinate system.
 - f. The x and y coordinates calculated in step 12.d are saved because they allow the position of the local coordinate system of the processed object to be determined in the local coordinate system of the parent object in the hierarchical structure of the building.
13. `IfcLocalPlacement`, `IfcDirection`, and `IfcCartesianPoint` class objects that define the location of the local coordinate system of the object in the local coordinate system of the parent object in the hierarchical structure of the building are saved in dedicated tables in BIGI-S.
14. `IfcAxis2Placement3d`, `IfcLocalPlacement`, `IfcCartesianPoint`, `IfcPolyline`, `IfcExtrudedAreaSolid`, `IfcShapeRepresentation`, `IfcShapeRepresentation`, and `IfcProductDefinitionShape` class objects that define the geometry of the object in its local coordinate system and the location of this object in the local coordinate system are saved in dedicated tables in BIGI-S.
15. Reference to the `IfcProductDefinitionShape` class object is saved in the `Representation` attribute of the room.

Conversion from the CityGML schema to the BIGI-S is much less complicated. The conversion process involves rewriting the basic data about the object, geometric data, and properties (Figure 5).

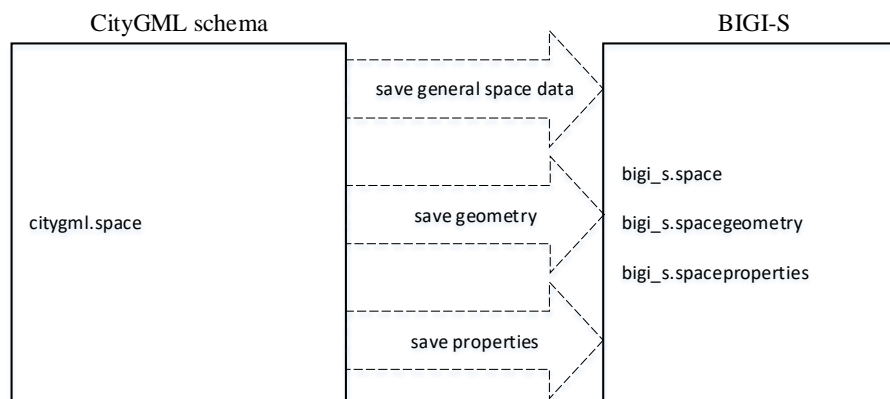


Figure 5. General scheme for transforming data from the CityGML schema to the BIGI-S based on the example of the Space object.

The general algorithm for the transformation of data related to the “room” class from the CityGML to BIGI-S is as follows:

1. All descriptive attributes (class, function, and usage) are read from the citygml.room table.
2. These attributes are saved in the bigi_s.propertyinglevalue table in the appropriate attributes of the property sets attached to the object.
3. The geometric elements of Solid and MultiSurface are read from the citygml.space table at the LoD4 detail level.
4. The geometric data for LoD4 is saved in the bigi_s.spacegeometry table.

The inverse conversion of data from the BIGI-S to the CityGML schema is analogous. It includes basic object data, object geometry, and properties (Figure 6).

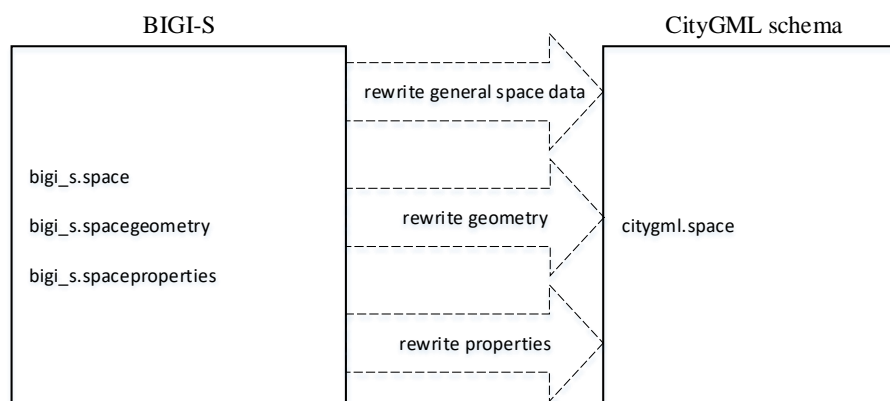


Figure 6. General scheme for transforming data from the BIGI-S to the CityGML schema based on the example of the Space object.

The general algorithm for transforming data related to the “room” class from the BIGI-S to the CityGML schema is as follows:

1. The appropriate attributes from the property sets attached to the room are read from the bigi_s.spacegeometry table.
2. The attributes are saved in the citygml.space table.
3. The geometric data for LoD4 for the room being processed are read from the bigi_s.spacegeometry table.
4. The geometric data are saved in the citygml.space table.

In conclusion, let us analyze the example of room data editing by BIM and GIS applications. The BIM application uses data for a single room. In this case, it calls stored procedures in the IFC database schema allowing to read data about object classes *IfcSpace* (basic data of the room), *IfcProductDefinitionShape*, *IfcShapeRepresentation*, *IfcExtrudedAreaSolid*, *IfcArbitraryClosedProfileDef*, *IfcPolyline*, *IfcCartesianPoint*, *IfcLocalPlacement*, *IfcAxis2Placement3D*, *IfcDirection* (data on the geometry of the object), *IfcRelDefinesByProperties*, *IfcPropertySet*, *IfcComplexProperty*, and *IfcPropertySingleValue* (object descriptive data). These stored procedures are *ifc.getspace*, *ifc.getproductdefinitionsshape*, *ifc.getshaperepresentation*, *ifc.getpropertyset*, etc. The routines retrieve relevant data from tables in BIGI-S, convert it to an IFC model, and make it available for BIM application. The modified data is saved by calling the stored procedures: *ifc.updateinspace*, *ifc.updateproductdefinitionsshape*, *ifc.updateshaperepresentation*, *ifc.updatepropertyset*, etc. At the same time, the GIS system uses the data of the same room using a stored procedure in the CityGML database schema: *citygml.getspace*, which retrieves data from tables in BIGI-S and converts it to CityGML format. Data is saved by the *citygml.updateinspace* stored procedure in the CityGML database schema, which converts data from CityGML and saves it in tables in BIGI-S. In this way, assuming the right performance is ensured, conversions can be performed almost invisibly to users of BIM and GIS applications.

3.6. Tests of Access to the Common Database from the Level of BIM and GIS Applications

As part of the research, six IFC models were transferred into an integrated database. The levels of complexity of these buildings, determined by the number of IFC objects in each of their models, are illustrated in Table 2.

Table 2. The number of IFC objects in the test models.

IFC Test Model	Objects	Floors	Rooms	Walls	Points
Church	98,915	5	52	172	27,432
Office building	100,058	5	0	219	24,154
School	312,705	4	83	427	66,786
Hospital	526,341	6	657	2320	100,991
University building	1,915,427	8	258	1740	331,291
College building	2,846,624	8	480	3005	751,262

The test IFC models were assigned globally unique identifiers (GUIDs), unique 128-bit numbers produced by applications to identify a particular component, application, file, database entry, and/or user. For each test model, a schema with a name identical to the GUID was created; then, the IFC model was imported into each schema. The next step was to convert the selected data from the IFC schemas to the BIGI-S. The high number of object classes in the IFC model (approximately 770) resulted in the need to limit the number of prepared stored procedures at the stage of research. Procedures were implemented only for those IFC classes that were used in the test IFC models. Such an approach was sufficient to assess the viability of the task. The most characteristic objects, such as the building, floors, rooms, walls, ceilings, doors, and windows, were selected for the test. The implementation of the procedures for the remaining classes representing the elements of a building was similar. Each of the above-mentioned test models was correctly imported into a relational database according to the adopted concept and procedures. Detailed tests showed good performance results. Saving a single room in the database, along with the conversion to BIGI-S, takes about 50 ms. Similar results were obtained when retrieving a single room from the database. It takes approximately 8 s to write or read a set of 500 objects. The tests were carried out on an environment consisting of a PostgreSQL 9.5 database server running on the CentOS 7 operating system (2 CPU, 4 GB RAM) and a Windows 8.1 workstation (6 CPU, 32 GB RAM). It was decided to use the Revit and ArchiCAD applications for testing, as these are currently widely used to create BIM models. According to the scheme of cooperation between BIM and GIS software with a common relational database presented in Figure 1, the test

import process of the IFC models was carried out with the export IFC file from the BIM software and imported to the BIGI-D. Technically, IFC data from the BIM application can be accessed through remote mechanisms, such as ODBC (ODBC—Open Database Connectivity)/JDBC (JDBC—Java Database Connectivity)/OLEDB (OLE DB—Object Linking and Embedding, Database)/ADO (ADO—ActiveX Data Objects). The proposed solution allows BIM applications to communicate with the BIGI-D through the import/export of IFC files and through the data access mechanism. In the latter case, the BIM application that can access the data stored in the database communicates with the database through standard data access protocols. This is a method analogous to the data manipulation performed by GIS applications.

BIM applications that are currently available on the market do not have mechanisms to work with the data stored in a database, so it was impossible to carry out a real, fully comprehensive test of access to the proposed BIGI-D. It was decided to simulate this process. The simulation involved reading and writing data using a special application developed by the authors. The application simulated a module of access to the BIM data recorded in a relational database. In the case of the final implementation of the discussed concept of this type, the module should be added to the existing BIM applications. It should be stressed that the construction of such a module is very simple and inexpensive. The simulation included saving and reading the entire model; reading the data of all objects of the selected class; reading the data of the object with a given identifier; and reading the data of the objects that defined the spatial structure of the BIM project, i.e., the building, floor, and rooms, as well as reading the geometry of the selected object, taking into account its location in the global geodetic coordinate system. In the next step, table contents were visualized in the BIGI-S. The final stage of testing was the reading of the data in the test building models from the CityGML schema. The reading process was seamless. The correctness of the data read in the CityGML model confirmed the correctness of the proposed method.

3.7. Evaluation of the Adopted Procedure

The results confirmed the validity of the initial assumption: that it was possible to implement the IFC model in a relational database in a way that allows the use of the IFC data saved in the database by BIM applications, without needing to significantly change the data operation methods that these applications currently perform. At the same time, such an approach allows GIS applications to operate on the same data. The experiment proved that the IFC model can be implemented in a relational database in a way that makes it possible to perform data writing and reading, which is currently done by other methods using BIM applications.

During the research, the advantages and disadvantages of the two methods of converting data within BIGI-D were analyzed: copying data with schema changes and dynamic data conversion to another schema “on the fly”. The advantages of the solution based on data copying include the ease of access to the spatial data saved in the IFC schema by BIM applications (direct reading of IFC data, only saved in a different format). The main disadvantages of this approach are the data redundancy, the time needed for copying, and the required additional computer memory resources. The solution using dynamic data conversion ensures that the data are stored only once in the BIGI-S. CityGML and IFC schemas are created only through perspectives and stored procedures (and, thus, virtually). The main advantage of this solution is, therefore, the elimination of data redundancy. The disadvantage, however, is the necessity to create rather complicated perspectives that translate selected data between schemas in the case of spatial data conversion into the IFC, while requiring very fast access to these data. It was decided that the advantages of the second approach were greater; therefore, this method was used to build the prototype.

4. Discussion

The originality of the proposed solution lies, among other things, in that it allows BIM and GIS applications to “see” their source data as they did before integration, while gaining access to other data

consistently. The BIM application sees GIS data and vice versa. This limits the necessary modifications of the BIM and GIS software, because it is sufficient to modify only the data writing format and not the data model. The storage of BIM and GIS data in a uniform database environment is another advantage of this solution. Conversions can be performed very efficiently and even imperceptibly to users.

The authors' solution involves the use of the spatial data-processing mechanisms available in popular environments of relational database management. Therefore, BIM data can be stored in a relational database management system (e.g., in PostgreSQL with the PostGIS module, Oracle Spatial, and Graph or in the SQL server). For the prototype and testing, it was decided to use the PostgreSQL relational database management system and the PostGIS extension supporting spatial data, spatial indexes, and spatial functions. The advantage of PostgreSQL is the ability to create a hierarchy of dependent tables, which makes it easier to represent the IFC model. Therefore, the IFC class inheritance hierarchy can be restored in the form of inherited tables in the database.

The scenario involving the use of dynamic data conversion requires the implementation of stored procedures that are able to return the data collected from the tables of an integrated spatial database for the given GUID building, transforming them "on the fly" into the IFC model using the perspective mechanism and stored procedures. The advantage of this solution is its continuous access to the current state of data through BIM and GIS applications.

5. Conclusions

This article described research attempting to find an effective, practical, and useful method to integrate BIM and GIS data environments. After a detailed analysis of the various publications and the available BIM software, a new method of BIM and GIS data integration was proposed that used an integrated spatial database, which consisted of three different schemas (in terms of both the structure and the separate places in the database). However, it is not an integrated database in the full meaning of the term based on a uniform conceptual model but, rather, a solution based on the process of on-the-fly integration of data collected in several harmonized application schemes. The research included experimental works, as well as preparation of the solution prototype.

Using a database as storage for IFC data allows GIS systems to use current BIM data without having to import IFC files into their formats/models, which opens up completely new functional possibilities. The proposed integrated database of BIM/GIS spatial data makes it possible to store GIS and BIM data, enabling the use of the same data by both types of systems simultaneously and in a consistent manner. This allows BIM systems to obtain simultaneous access to BIM and GIS data, which may be needed in, for example, the process of analyzing a building and its immediate surroundings. At the same time, GIS can obtain up-to-date building data necessary for spatial analyses, building management, or route mapping in navigation applications. In addition, users of BIM systems gain new technological solutions as follows:

- Providing opportunities for group work offered by the management systems of relational databases with the possibility of using various BIM software while working on a single project.
- Providing standard mechanisms for controlling the access rights to the data offered by relational database management systems.
- Providing mechanisms for the transactional processing of the data offered by relational database management systems.
- Providing backup mechanisms offered by relational database management systems.

The condition for implementing such a solution is to add a database access mechanism to the BIM tools. Such a mechanism could replace the existing BIM application functionality of working on its own data files in binary formats. This is probably one of the simplest solutions, because it requires only a technical modification without the need for integration into the conceptual model. In terms of GIS, virtually no significant modification is necessary, as long as the software supports the CityGML schema. Of course, there is nothing to prevent the proposed concept from being applied to

data schemas alternative to CityGML. The proposed concept also makes it possible to create additional thematic schemas that allow access to the integrated BIM/GIS data stored in the BIGI-S in a way readable for other applications using these data, such as applications that support the process of property management.

Author Contributions: Conceptualization, M.W. and D.G.; methodology, M.W. and D.G.; investigation, M.W.; software, M.W.; data curation, M.W.; visualization, M.W.; writing—original draft preparation, M.W. and D.G.; writing—review and editing, D.G.; and supervision, D.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was financed from the funds for statutory research of the Warsaw University of Technology and grant number 504/03787/1060/42.000100.

Acknowledgments: The authors express their gratitude to SWG Sweden for providing the data for the tests. Some tests were performed on the sample BIM model provided by the Institute for Automation and Applied Informatics (IAI)/Karlsruhe Institute of Technology (KIT).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ruffle, S. Architectural design exposed: From computer-aided drawing to computer-aided design. *Environ. Plan. B Plan. Des.* **1986**, *13*, 385–389. [CrossRef]
2. Autodesk, Inc. *Building Information Modeling, Autodesk Whitepaper*; Autodesk, Inc.: San Rafael, CA, USA, 2002.
3. Schrock, G. Esri-Autodesk Partnership, Esri. 21 January 2019. Available online: <https://www.esri.com/about/newsroom/arcnews/esri-autodesk-partnership/> (accessed on 17 February 2020).
4. ISO Technical Committee. *ISO/TC 211 Geographic Information/Geomatics. ISO/CD TS 19166*; ISO: Geneva, Switzerland, 2020.
5. Wyszomirski, M. The concept of an integrated spatial database for BIM and GIS systems. In *Koncepcja Zintegrowanej Bazy Danych Przestrzennych Dla Systemów BIM i GIS*; Politechnika Warszawska: Warszawa, Poland, 2019.
6. Döllner, J.; Hagedorn, B. *Integrating Urban GIS, CAD and BIM Data by Service-Based Virtual 3D City Models*; Taylor & Francis Group: London, UK, 2007.
7. Lapiere, A.; Côté, P.-M. Using Open Web Services for urban data management: A testbed resulting from an OGC initiative for offering standard CAD/GIS/BIM services. In *Urban and Regional Data Management*; Taylor & Francis Group: London, UK, 2007.
8. Işıkdağ, Ü.; Underwood, J.; Aouad, G.; Trodd, N. Investigating the Role of Building Information Models as a Part of an Integrated Data Layer: A Fire Response Management Case. *Arch. Eng. Des. Manag.* **2007**, *3*, 124–142. [CrossRef]
9. Isikdag, U.; Underwood, J.; Aouad, G. An investigation into the applicability of building information models in geospatial environment in support of site selection and fire response management processes. *Adv. Eng. Inform.* **2008**, *22*, 504–519. [CrossRef]
10. Isikdag, U.; Zlatanova, S. Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models. In *Lecture Notes in Geoinformation and Cartography*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 79–96. [CrossRef]
11. De Laat, R.; Van Berlo, L. Integration of BIM and GIS: The development of the CityGML GeoBIM extension. In *Advances in 3D Geo-Information Sciences*; Springer: Berlin/Heidelberg, Germany, 2011. [CrossRef]
12. Cheng, J.; Deng, Y.; Du, Q. Mapping Between BIM Models and 3D GIS City Models Of Different Levels of Detail. In Proceedings of the 13th international conference on construction applications of virtual reality, London, UK, 30–31 October 2013.
13. Deng, Y.; Cheng, J.C.; Anumba, C. Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. *Autom. Constr.* **2016**, *67*, 1–21. [CrossRef]
14. Rafiee, A.; Dias, E.; Fruijtier, S.; Scholten, H. From BIM to Geo-analysis: View Coverage and Shadow Analysis by BIM/GIS Integration. *Procedia Environ. Sci.* **2014**, *22*, 397–402. [CrossRef]
15. Isikdag, U.; Zlatanova, S. A SWOT analysis on the implementation of Building Information Models within the geospatial environment. In *Urban and Regional Data Management: UDMS 2009 Annual*; Taylor & Francis Group: London, UK, 2009. [CrossRef]

16. Naik, G.; Aditya, M.; Naik, S.B. Integrated 4D Model Development for Planning and Scheduling of a Construction Project using Geographical Information System. In Proceedings of the 2nd International Conference on Construction and Project Management, Singapore, 16–18 September 2011.
17. Nagel, C.; Stadler, A.; Kolbe, T.H. Conceptual Requirements for the Automatic Reconstruction of Building Information Models from Uninterpreted 3D Models. In Proceedings of the Academic Track of the Geoweb 2009-3D Cityscapes Conference, Vancouver, BC, Canada, 27–31 July 2009.
18. Fosu, R.; Suprabhas, K.; Rathore, Z.; Cory, C. Integration of Building Information Modeling (BIM) and Geographic Information Systems (GIS)—A literature review and future needs. In Proceedings of the 32nd CIB W78 Conference, Eindhoven, The Netherlands, 27–29 October 2015.
19. Amirebrahimi, S.; Rajabifard, A.; Mendis, P.; Ngo, T. A Data Model for Integrating GIS and BIM for Assessment and 3D Visualisation of Flood Damage to Building. *Locate* **2015**, *15*, 12.
20. Amirebrahimi, S.; Rajabifard, A.; Mendis, P.; Ngo, T. A framework for a microscale flood damage assessment and visualization for a building using BIM–GIS integration. *Int. J. Digit. Earth* **2015**, *9*, 363–386. [[CrossRef](#)]
21. Hjelseth, E.; Thiis, T. Use of BIM and GIS to enable climatic adaptations of buildings. In *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2008*; Taylor & Francis Group: London, UK, 2009.
22. Stouffs, R. A Triple Graph Grammar Approach to Mapping Ifc Models into Citygml Building Models. In Proceedings of the 3rd CAADRIA Conference, Beijing, China, 17–19 May 2018.
23. Olsson, P.-O.; Axelsson, J.; Hooper, M.; Harrie, L. Automation of Building Permission by Integration of BIM and Geospatial Data. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 307. [[CrossRef](#)]
24. Wu, B.; Zhang, S. Integration of GIS And BIM for Indoor Geovisual Analytics. *ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2016**, 455–458. [[CrossRef](#)]
25. Jusuf, S.K.; Mousseau, B.; Godfroid, G.; Soh, J.H.V. Path to an Integrated Modelling between IFC and CityGML for Neighborhood Scale Modelling. *Urban Sci.* **2017**, *1*, 25. [[CrossRef](#)]
26. Chen, L.-C.; Wu, C.-H.; Shen, T.-S.; Chou, C.-C. The application of geometric network models and building information models in geospatial environments for fire-fighting simulations. *Comput. Environ. Urban Syst.* **2014**, *45*, 1–12. [[CrossRef](#)]
27. Hagedorn, B.; Trapp, M.; Glander, T.; Döllner, J. Towards an Indoor Level-of-Detail Model for Route Visualization. In Proceedings of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, Taipei, Taiwan, 18–20 May 2009; pp. 692–697. [[CrossRef](#)]
28. Hor, A.-H.; Sohn, G.; Claudio, P.; Jadidi, M.; Afnan, A. A Semantic Graph Database For BIM-GIS Integrated Information Model for an Intelligent Urban Mobility Web Application. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2018**, 89–96. [[CrossRef](#)]
29. El-Mekawy, M.; Östman, A. Semantic Mapping: An Ontology Engineering Method for Integrating Building Models in IFC and CityGML. In Proceedings of the 3rd ISDE Digital Earth Summit, Nessebar, Bulgaria, 12–14 June 2010.
30. El-Mekawy, M.; Östman, A.; Shahzad, K. Towards Interoperating CityGML and IFC Building Models: A Unified Model Based Approach. In *Advances in 3D Geo-Information Sciences*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 73–93. [[CrossRef](#)]
31. El-Mekawy, M.; Östman, A.; Hijazi, I. A Unified Building Model for 3D Urban GIS. *ISPRS Int. J. Geo-Inf.* **2012**, *1*, 120–145. [[CrossRef](#)]
32. El-Mekawy, M.; Östman, A. *Emerging Issues, Challenges, and Opportunities in Urban E-Planning*; A Unified Building Model for a Real 3D Cadastral System; IGI Global: Pennsylvania, PA, USA, 2015; pp. 252–279. [[CrossRef](#)]
33. Yuan, Z.; Shen, G. Using IFC standard to integrate BIM models and GIS. In Proceedings of the 2010 International Conference on Construction and Real Estate, Brisbane, Australia, 1–3 December 2010.
34. El Meouche, R.; Rezoug, M.; Hijazi, I. INTEGRATING AND MANAGING BIM IN GIS, SOFTWARE REVIEW. *ISPRS Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2013**, *2*, 31–34. [[CrossRef](#)]
35. Hwang, J.-R.; Hong, C.-H.; Choi, H.-S. Implementation of prototype for interoperability between BIM and GIS: Demonstration paper. In Proceedings of the IEEE 7th International Conference on Research Challenges in Information Science (RCIS), Paris, France, 29–31 May 2013; pp. 1–2. [[CrossRef](#)]

36. Wu, W.; Yang, X.; Fan, Q. GIS-BIM Based Virtual Facility Energy Assessment (VFEA)—Framework Development and Use Case of California State University, Fresno. In Proceedings of the 2014 International Conference on Computing in Civil and Building Engineering 2014, Orlando, FL, USA, 23–25 June 2014. [[CrossRef](#)]
37. Mignard, C.; Nicolle, C. Merging BIM and GIS using ontologies application to urban facility management in ACTIVE3D. *Comput. Ind.* **2014**, *65*, 1276–1290. [[CrossRef](#)]
38. Kang, T.W.; Hong, C.H. A study on software architecture for effective BIM/GIS-based facility management data integration. *Autom. Constr.* **2015**, *54*, 25–38. [[CrossRef](#)]
39. Karan, E.P.; Irizarry, J. Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services. *Autom. Constr.* **2015**, *53*, 1–12. [[CrossRef](#)]
40. Wang, H.; Pan, Y.; Luo, E. Integration of BIM and GIS in sustainable built environment: A review and bibliometric analysis. *Autom. Constr.* **2019**, *103*, 41–52. [[CrossRef](#)]
41. Tobiáš, P. BIM, GIS and semantic models of cultural heritage buildings. *GeoInform. FCE CTU* **2016**, *15*, 27–42. [[CrossRef](#)]
42. Zhao, L.; Liu, Z.; Mbachu, J. An Integrated BIM–GIS Method for Planning of Water Distribution System. *ISPRS Int. J. Geo Inf.* **2019**, *8*, 331. [[CrossRef](#)]
43. Ismail, M.H.; Ishak, S.S.M.; Osman, M. Role of BIM+GIS checker for improvement of technology deployment in infrastructure projects. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *512*, 012038. [[CrossRef](#)]
44. Rahman, S.A.F.S.A.; Maulud, K.N.A. Approaching BIM-GIS Integration for 3D Evacuation Planning Requirement Using Multipatch Geometry Data Format. *IOP Conf. Ser. Earth Environ. Sci.* **2019**, *385*, 012033. [[CrossRef](#)]
45. Rodrigues, H.; Teixeira, J.; Matos, R. Development of a Web Application for Historical Building Management through BIM Technology. *Adv. Civ. Eng.* **2019**, *2019*, 9872736. [[CrossRef](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).