ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XI-4-2026
XXV ISPRS Congress 2026 "From Imagery to Understanding", Commission IV, 4–11 July 2026, Toronto, Canada

IFC and QGIS integration for the Integrated Water Service Management

Michele Berlato[1], Giorgio Agugiaro[2], Ken Arroyo Ohori[2], Carlo Zanchetta[3]

[1] DTG – Department of Management Engineering, University of Padova, Italy – michele.berlato@phd.unipd.it
[2] 3D Geoinformation group, Department of Urbanism, Faculty of Architecture and Built Environment, Delft University of Technology, Delft, The Netherlands – {g.agugiaro, k.ohori}@tudelft.nl
[3] DICEA – Department of Civil, Building and Environmental Engineering, University of Padova, Italy – carlo.zanchetta@unipd.it

Keywords: Industry Foundation Classes; ifcSQL; QGIS; Integrated Water Service; Infrastructure Management

Abstract

Integrated Water Service (IWS), which combines water supply and wastewater treatment, requires complex geometric and semantic management. Building Information Modelling (BIM) and Geographic Information Systems (GIS) are the two main geospatial technologies involved in this field. In very simple terms, BIM allows to have 3D models with detailed geometric and semantic information, and GIS permits to geolocate and manage the models in the territory. To facilitate the integration of these two systems, we propose to manage the BIM models through a standardised relational database. In the BIM world, relational databases are not yet widely used, but the technology is already available. For example, ifcSQL is an encoding of the Industry Foundation Classes (IFC) data model for a relational database. This article proposes an extension of the ifcSQL database with the added possibility to store the georeferenced explicit geometries of the IFC models. Additionally, we present a prototype to make such IFC-based data available via QGIS. In this way, a user can interact with BIM data using open GIS technologies. As a result, it is possible to visualise the models in 2D and 3D, and to perform queries on their attributes. A set of real-world case studies has served as testing ground to develop the functionalities that allow for the interaction with the BIM models via QGIS. Such test cases originate from interviews with a company that manages IWS in Northeast Italy.

1. Introduction and background

1.1. Integrated Water Service management: Scenario and Motivations

In Italy, water supply and wastewater treatment are both managed by the same company for each territory recognised as an ATO (i.e. Ambito Territoriale Ottimale, in English: Optimal Territorial Area), with reference to a watershed (Regione del Veneto, 2025). The integration of these two services is called Integrated Water Service (IWS), and it includes numerous hydraulic infrastructures due to the diversity and complementarity of these two systems. For many years, companies involved in IWS management have been using GIS systems to manage hydraulic infrastructure in their area of competence. In fact, GIS systems, which are traditionally oriented at territorial scale, allow for advanced spatial analyses to be performed on the georeferenced elements available in a territory. However, with the growth in the availability of BIM models, also due to new regulatory requirements, in recent years more possibilities have opened up for better and more detailed management of IWS hydraulic infrastructure. BIM models which are traditionally created at the building scale, allow for detailed geometric and semantic management of every element of a facility. This has led to a need for IWS companies to integrate BIM and GIS to take advantage of both available technologies. Joining these two systems in the IWS management can improve data interaction and visualisation and allow for querying 3D models of hydraulic infrastructures in the territory.

1.2. BIM and GIS Integration for Integrated Water Service Management: Background and Gaps

The need for BIM and GIS integration is not solely required by IWS. In fact, several authors address this topic for a variety of purposes depending on the specific research field. In literature, the integration of BIM and GIS is generally known as GeoBIM, which tackles the integration of 3D geoinformation (usually 3D city models) and 3D building information models (BIM). The exchange of information between the two fields allows for mutual enrichment, bringing numerous advantages, such as a more detailed model of the city and the representation of the BIM model in its geographical context (Noardo et al., 2020). Over the years, several issues and challenges have been highlighted in the GeoBIM integration process. However, such challenges depend not only on the different application scale (buildings, cities, infrastructure, etc.), but also on the different phases of a project's life cycle (design, execution, operation and maintenance, etc.). For example, BIM and GIS integration for existing infrastructure has specific operation and maintenance purposes (Cepa et al., 2024). Therefore, each case must be analysed individually to find the appropriate solution. Furthermore, it is important to distinguish between the different types of infrastructure (gas, water, urban, etc.), although the significance of BIM and GIS integration for the management of all these infrastructure systems has been widely recognised (Demir and Yomralioglu, 2024). IWS management is a clear example of the importance of managing, visualising, and querying different BIM models of water infrastructure in a larger territory (Zhao et al., 2019). However, to do this, it is necessary to understand how to obtain an integrated management of BIM models and GIS data in a GIS environment. In the case of IWS, due to the spatial nature of both BIM and GIS data and the overall territorial scale of management, it is reasonable to think that this integration should take place in a GIS environment. This is also because it is necessary to simultaneously view and manage all the hydraulic infrastructure in a territory as a whole. For these reasons, a good starting point is to first analyse the BIM domain to identify potential integration incompatibilities before moving toward the GIS domain.

Since its first publication in 1996, IFC has been the open reference data model in the BIM world. Maintained and updated by the non-profit organisation buildingSMART International (bSI), Industry Foundation Classes (IFC) is the ISO standard for the interoperability of BIM models (ISO 16739-1, 2024). IFC, as a data model developed for the BIM world, has some inherent differences compared to the GIS world. These issues have been widely discussed in the literature, and analysing all the differences in detail is beyond the scope of this paper. However, we will mention here some peculiar issues related to integration within IWS, which will be discussed in the next paragraphs under the following categories: (1) the representation of geometry, (2) the coordinate reference system, and (3) the data encoding.

1) In the BIM world the geometry can be represented in four ways: Boundary Representation (B-Rep), Constructive Solid Geometry (CSG), Swept Solid and Tessellated Geometry. CSG creates a complex object by using Boolean operators to combine simpler objects, while a “Swept Solid” creates an object by sweeping a 2D profile along a path. Instead, B-Rep creates a geometry using an aggregation of boundary surfaces which enclose the object (Zhu and Wu, 2022). In addition, starting from IFC4 it is possible to represent the geometry by means of a tessellation (IfcTessellatedFaceSet), which is made of triangular or polygonal faces (buildingSMART International, 2025). In IFC, the geometry is typically stored in an implicit or parametric way (CSG and Swept solid). The standard also allows geometry to be encoded explicitly (Tessellated Geometry), but this option is rarely used because modelling software generally does not allow it to be applied effectively. Conversely, the GIS world, which is oriented towards spatial analysis, represents the elements only in an explicit way by means of simple vector elements (points, lines, polygons). Additionally, in the GIS world, it is very common to store feature attributes and geometries in a database, because managing a lot of data in a territory is easier with the support of a database. When a GIS geometry is stored in a database, it is preferable to store it under the guidelines of the Simple Features Access (SFA) ISO standard (ISO 19125-1, 2004). This standard, maintained and updated by the Open Geospatial Consortium (OGC), describes the common architecture for a simple feature geometry (SFG). In the SFA standard, geometries (SFG) can be serialised into a string representation known as Well-Known Text (WKT). WKT is a standardised text syntax to represent vector geometric objects on a map with a reference coordinate system. WKT is just one way of representing geometry in the GIS world, but its simplicity, standardisation, and compatibility with databases also make it one of the best and most widely used encodings to transfer geometries from one system to another. Finally, geospatial databases do not support implicit geometries as defined in IFC, and therefore it is necessary to convert them into explicit geometries to ensure compatibility with GIS environments. This means that IFC geometries must be first converted from implicit to explicit (ideally encoded in WKT) before being integrated into a GIS system.

2) BIM models use a local coordinate system, taking a point as origin of the coordinate reference system. The local system is ideal for a BIM model of a building, but not for long infrastructures, like roads or pipelines, where the curvature of the Earth must be taken into consideration. In the GIS world, a coordinate system can be Geographic or Projected. Typically, it is preferable to use a projected coordinate system in GIS software because it is more suitable for mapping on 2D devices such as a computer monitor or a printer. However, to georeference a BIM model, there are some limitations which have already been extensively discussed in the literature, e.g. in (Azari et al., 2025). The important point is that, from IFC4, an IFC model can be georeferenced in a projected coordinate system using two classes, i.e. *IfcProjectedCRS* and *IfcMapConversion*. According to buildingSMART International (bSI) in IFC4.3: *IfcProjectedCRS* is used to define the coordinate reference system of the map to which the map translation of the local engineering coordinate system of the construction or facility engineering project relates; *IfcMapConversion* deals with transforming the local engineering coordinate system, often called the world coordinate system, into the coordinate reference system of the underlying map. It should be noted that the transition from the local coordinate system to a projected coordinate system of an IFC model is not a strict geodetic transformation. In fact, in the transformation from the local to the projected coordinate system, the model is simply translated, rotated, and scaled. In this way, the reference point of the BIM model gets the projected coordinates, which are then stored in *IfcMapConversion*, while the reference coordinate system is defined and then stored in *IfcProjectedCRS*. The BIM model is therefore georeferenced as a single block on a plane surface, as is typical of the BIM approach. This is generally not a problem if the spatial extent of the model is limited, since the Earth's surface can be approximated to a plane in small areas. However, for large projects, it is necessary to divide the project into multiple, smaller BIM models to mitigate the error caused by georeferencing that is not completely strict. Nevertheless, as of today, this is still the most advanced method available in the IFC standard (Clemen and Görne, 2019). In conclusion, it is possible to correctly georeference an IFC model using the aforementioned classes but keeping in mind that, in order to move from a local to a projected coordinate system, some simplifications are still required.

3) The IFC data model is characterised by structured and granular semantic. Every element is not only a geometric object but instead an entity with its own identity. Each entity, which is an IFC class, has a set of predefined attributes, properties and relations with the other entities. These structures are represented through the EXPRESS data model. IFC can be encoded in various file formats, each of which has advantages and trade-offs in terms of software support, scalability and readability. The STEP Physical File (SPF) is the most used IFC file format in practice and the most compact of the existing ones. However, an IFC-SPF file is designed to contain a single BIM model at a time. Therefore, this encoding is not suitable for the management of several BIM models at the same time, which is exactly what characterises IWS, which deals with many BIM models in a territory. For this reason, a scalable solution is needed, from a single pump to a province. Conversely, in the GIS world the elements are typically stored in a database with which it is possible to manage and query a lot of entities at the same time. The most commonly used type of database in the GIS world is the relational database, as GIS software is typically designed to support it and display the geometry of the stored elements. In addition, the use of a relational database allows for advanced spatial query functionalities, and gives the possibility to implement the OGC standards, such as SFA. Therefore, a relational database encoding for the BIM world is more suitable for interacting with the GIS world. Firstly, because BIM data is then directly linked to GIS software, and secondly, because BIM geometries could be written in standard GIS formats. However, despite its popularity, IFC does not have an official SQL encoding yet, and today the use of IFC in a relational database is still limited. The situation is different for non-relational databases (NoSQL), which have experienced considerable success in the BIM world in recent years thanks to their flexibility and scalability. The problem is that NoSQL databases are not comparable to SQL databases in terms of GIS functions, as they have less capacity for complex spatial queries and topological

integrity (Ellul et al., 2024). However, ifcSQL has been proposed as a relational database schema to store IFC data according to the official IFC data model (Bock and Eder, 2025a). With this encoding, it is possible to store many BIM models in one database, which is a good starting point for BIM and GIS integration for the reasons highlighted.

In BIM and GIS integration as presented in the scientific literature, it is possible to find only a few articles that use a relational database, and the proposed solutions are not useful for the specific challenges of IWS management. In terms of proprietary solutions, ArcGIS GeoBIM allows the integration of BIM models in a GIS map. The problem is that, since it is not an open-source tool, the underlying database structure is hidden, not documented and cannot be extended or customised. This means that users are bound to a predefined structure that may not meet their management requirements. However, to the best of our knowledge, no one in the literature has tried to integrate BIM and GIS using ifcSQL as a database to store the IFC models. This is a research opportunity because ifcSQL, as a relational database encoding of IFC, can already be connected to GIS software through its existing database integration functionality. The problem is that the geometries stored in ifcSQL are not in an explicit format and therefore cannot be immediately accessed and published. Despite this limitation, if it were possible to insert the explicit geometry into the ifcSQL database after correctly georeferencing the IFC model, integration would be easily achievable. About the integration of 3D models in a GIS system starting from a database, an interesting work has been presented in the literature, which created a QGIS plug-in for the visualisation and management of the CityGML models stored into the 3DCityDB (Agugiaro et al., 2024). In a similar way, the IFC models stored in the ifcSQL database could be ideally accessed from QGIS. For this reason, QGIS seems to be a good starting point in terms of an open-source GIS platform for the integration of georeferenced BIM models with typical GIS data.

The aim of this work is developing a methodology based on the relational database encoding of IFC (ifcSQL) to integrate and manage BIM models within an open-source GIS software (e.g. QGIS). In the next sections the methodology will be explained and tested based on some real case studies provided by an IWS company from Northeast Italy.

2. Methodology

2.1. Tools

Before explaining the proposed workflow, a presentation of every tool used to fulfil the aim is provided.

2.1.1. ifcSQL, developed and maintained by (Bock and Eder, 2025a), is an encoding of the IFC data model for a relational database that implements and follows the EXPRESS data model. With ifcSQL it is possible to upload, download and delete multiple IFC-SPF files from the same database instance. When an IFC file is uploaded, the information contained in it is stored over multiple tables. The most important groups of tables are IfcProject, IfcInstance and IfcSchema:

- Inside the group IfcProject, the main information of every project (i.e. IFC file) is stored, as well as the correspondence of each IFC entity with the project to which it belongs. For example, the information of a project stored inside the table ifcProject.Project is: ProjectId, ProjectName, ProjectDescription, SpecificationId (which is the IFC schema), Author, etc.
- Inside the group IfcInstance, all the attributes and properties of each entity are stored, divided into separated tables based on the data type of each attribute (binary, float, integer, etc.). For example, the information stored inside the table ifcInstance.EntityAttributeOfString is: GlobalEntityId (of the IFC entity), OrdinalPosition (of the attribute), TypeId (of the attribute) and Value (of the attribute, in string form).
- The group of tables IfcSchema contains the information related to the official IFC schemas (IFC2X3; IFC4, IFC4.3). These tables are not modified except to add a new release of the IFC schema.

IFC data can be exported from the database to an IFC-SPF file, allowing for a lossless roundtrip. More information about the development process of ifcSQL can be found in (Toldo et al., 2025), while detailed information to install and set up the database is provided in the official ifcSQL GitHub pages (Bock and Eder, 2025a, 2025b, 2025c). In short, ifcSQL consists of a set of libraries, some of which are developed in C#, that can be used to create and populate the initial ifcSQL database with IFC schema information. With these libraries it is possible to do some operations like upload, download and delete the IFC-SPF files as mentioned before. When it comes to BIM and GIS integration, ifcSQL has one main limitation: it is a database schema currently developed only for Microsoft SQL Server (2019 or higher), which has some known limitations in terms of geospatial data support.

2.1.2. Microsoft SQL Server (MSSQL) is a proprietary relational database developed and maintained by Microsoft. It can be used free of charge in its “express” version, which is limited to 10 GB storage size. However, MSSQL does not support 3D geometry, and it offers just simple support for 2D geometries. In addition, the set of spatial functions implemented is rather limited when compared to the rich set of functions available, for example, with PostGIS for PostgreSQL. Finally, MSSQL is generally bundled with SQL Server Management Studio (SSMS), which is a useful front-end to work with and access/display data in the database.

2.1.3. PostgreSQL is the most used open-source relational database in the GIS world. PostgreSQL, thanks to its extension PostGIS, offers a lot of spatial functionality and supports 3D geometry. In addition, PostgreSQL has the possibility to directly access the data of another database through the foreign-data wrapper (fdw) extension. For the connection to MSSQL from PostgreSQL, the tds_fdw_16 library has been used, which is a custom tds_fdw for PostgreSQL 16 for Windows (Dawkins, 2024). A convenient front-end to administer PostgreSQL and work with its data is pgAdmin.

2.1.4. QGIS is an open-source GIS software, developed from the very beginning with the goal to support PostGIS geometries stored in PostgreSQL database instances. Although QGIS has evolved over the last 20 years to be a very powerful tool to visualise, organise, analyse and manage GIS data, it still lacks native support for any BIM-related data formats.

2.1.5. IfcOpenShell is an open-source library to read, write, and modify IFC files, and it is compatible with Python. Among its functionalities there is IfcConvert, which is a command-line application to convert IFC geometry into another type of geometry, such as OBJ and others.

https://doi.org/10.5194/isprs-annals-XI-4-2026-13-2026 | © Author(s) 2026. CC BY 4.0 License.

2.2. Workflow

This section presents the workflow adopted to integrate BIM and GIS to improve IWS management by overcoming the existing challenges. From the analysis carried out in the previous paragraphs, it is clear that integrating BIM models into a GIS system via the relational database encoding of IFC (ifcSQL) is the most convenient solution. In order for IFC data to be published correctly in GIS software, the geometry must be in a supported format. So, one of the important points of this work is to convert the implicit geometry of IFC into an equivalent explicit WKT text syntax. The decision to use WKT comes from the fact that it allows geometry to be represented in a simple and standardised way. It is also compatible with relational databases for input/output and can be interpreted by GIS software, such as QGIS. In addition, it should not be forgotten that geometries must also be correctly georeferenced. This is not a major problem because, as previously described, it is possible to correctly georeference an IFC model using the two classes *IfcProjectedCRS* and *IfcMapConversion*.

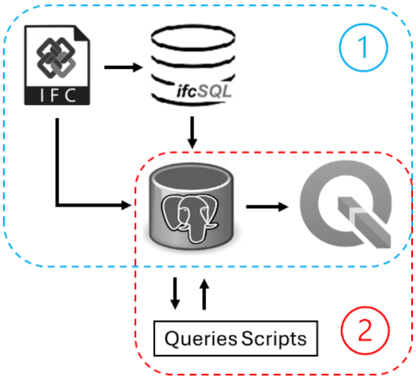

Figure 1. General schema of the Workflow with the two main parts together

The workflow has two main purposes. The first is to publish and visualise IFC models in QGIS through the use of ifcSQL. The second is to query the IFC data following some real case studies provided by a company that manages the IWS in Northeast Italy. For these reasons, the workflow, as can be seen in Figure 1, is divided into two main parts. The main challenge of the first part is that MSSQL does not allow a user to store the 3D explicit geometry and ifcSQL currently works only for MSSQL. As a solution to this problem, it is necessary to also use PostgreSQL as a database with the PostGIS extension. In this way, it is possible to manage the IFC semantic data and the IFC geometric data together and see the complete models in QGIS. For the second part, once we have all the data together, we can perform some SQL queries to meet the requirements of the case studies and publish a filtered view of the models in QGIS. To overcome the current limitations, in our workflow all the tools presented in the last section are used. More specifically, these are the versions used:

- Microsoft SQL Server 2022.
- SQL Server Management Studio 20.
- PostgreSQL 16 + PostGIS 3.5.
- pgAdmin 4 v. 9.4.
- QGIS 3.40.
- Python 3.12.
- IfcOpenShell 0.8.3.

2.2.1. First part: Publishing IFC models in QGIS. In the first part a unique environment is created by joining the semantic data and the explicit geometry of IFC. This part is fundamental because, at this moment, it is not possible to have an ifcSQL database schema in which semantic data and explicit geometry can be stored together. The following part explains how the workflow solves the problems mentioned above. Firstly, the IFC models that will be included in the workflow must comply with certain information specifications. They must be georeferenced correctly and exported following IFC4 or IFC4.3, since the methodology works for both the IFC schemas. Then, the first part of the workflow (Figure 2) starts from one side inserting the IFC data in the ifcSQL database (MSSQL) using the "IfcSharp Scripts" provided by Bock and Eder, and from the other side inserting the IFC explicit geometry into PostgreSQL using the "Geometry Script" that we have developed (the script is explained in the next Section 2.2.2). Afterwards, data stored in MSSQL and data stored in PostgreSQL are joined by means of the PostgreSQL fdw extension (tds_fdw_16). In this way, since spatial data consisting of the explicit 3D geometries of the IFC objects are stored as PostGIS geometries, it is possible to retrieve and visualise the IFC models in QGIS by simply dragging and dropping the PostgreSQL table containing the corresponding explicit and georeferenced geometries.

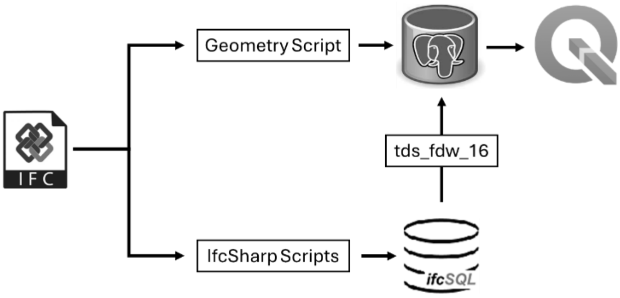

Figure 2. First part, with the purpose to merge semantic data and explicit geometry and publish them in QGIS

We have structured the PostgreSQL database to replicate the IfcInstance and IfcProject table groups coming from MSSQL, which contain the information of the projects and the entities in ifcSQL. In addition, it contains the schema IfcGeometry. Inside this last schema, there is the table named "entitygeometry" where the explicit geometries are stored. The table has five columns:

- GlobalId_MSSQL: contains the same numeric ID that identifies the entity inside the ifcSQL database.
- GlobalId_IfcFile: contains the same original GlobalId of the entity within the IFC file where it comes from.
- ProjectNumber_MSSQL: contains the same numeric ID that identifies the project of ifcSQL from which the entity derives.
- IfcClass: contains the IFC class of the entity.
- Geometry: contains the PostGIS geometry of the entity.

2.2.2. Python Geometry Script: we have developed a script to convert the implicit geometry of the IFC file to explicit geometry and then to insert the latter into the PostgreSQL database (Figure 3). This script is needed because ifcSQL does not generate explicit geometry at any point but instead stores the original implicit geometry in a number of tables following their original EXPRESS data model definition. The script starts with the selection of the IFC file. Once the file is read, the IFC classes to be converted are selected using the IfcOpenShell Python library. This step is done because not all IFC classes have geometry, and so there is no need to convert all of them. Then, the coordinate system information and the location of the IFC model are extracted from *IfcProjectedCRS* and *IfcMapConversion*. Afterwards, the implicit IFC geometries of the selected elements are converted to explicit geometries in OBJ format using IfcConvert. This step simplifies the following steps, as directly obtaining the WKT geometry from an IFC file is not easily achievable using existing libraries and requires additional custom implementation. Successively, the OBJ geometry is converted to WKT3D geometry, as converting explicit geometry such as OBJ to another such as WKT is simpler. In particular, the WKT type called “MULTIPOLYGON Z” is used. In this last part the coordinate system information and the location of the IFC model are inserted to correctly georeference every element. Finally, the geometry is inserted in the PostgreSQL table.

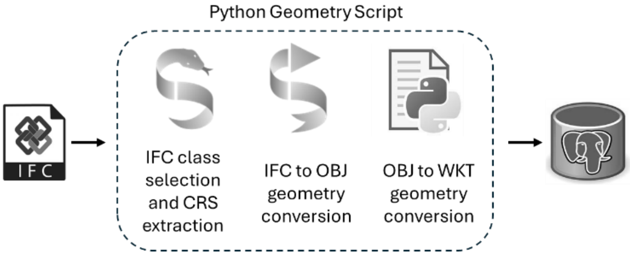

Figure 3. Python Geometry Script

2.2.3. Second part: Querying IFC data. The purpose of this second part is to query the data following some real case studies provided by an IWS company in Northeast Italy. Once the semantic data and the explicit geometry are collected in PostgreSQL, there is the possibility to query the data and visualise the results through QGIS (Figure 4). The PostgreSQL database is queried using some Python scripts that implement SQL queries. The queries generate database views consisting of the resulting geometries and attributes. At the moment, to facilitate the visualisation of the results, the views are also stored in the database as they can be imported into QGIS by a simple drag and drop operation.

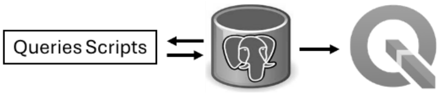

Figure 4. Second part of the workflow, with the purpose of querying the IFC data

To better understand which queries may be necessary in a real-world context, the previously mentioned company was interviewed. The requirements and suggestions resulting from the interviews were successively analysed and grouped to simplify their application and avoid creating different scripts for similar problems. In the end, we identified three main groups of queries:

- Selecting entities and their attributes inside an area (province, municipality, landside risk area, flood risk area, etc.).
- Selecting entities by means of an elevation filter (example: underground rooms, because they may be associated with a radon risk).
- Calculating area or volume of entities (example: total surface and volume of rooms with chemical agents within a municipality).

Some examples are provided in Section 3.2.

3. Results

3.1. Case studies and QGIS visualisation

The case studies used to evaluate the methodology for the BIM and GIS integration are provided by a company that manages the IWS in Northeast Italy, and they are typical facilities of the IWS. The first one is a Wastewater Lift Station (WWLS), which is a simple facility used to raise the wastewater to a higher level (Figure 5). The WWLS has been originally modelled in Revit, with the structural part and the MEP (Mechanical, Electrical, Plumbing) systems part modelled together in a single model. Finally, the WWLS is georeferenced using EPSG 7795 and exported in IFC4.3.

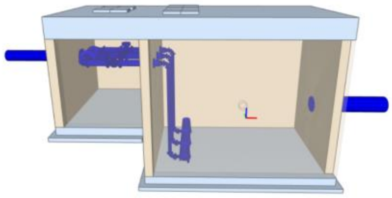

Figure 5. Wastewater Lift Station (WWLS)

The second BIM model is a Water Supply Plant (WSP), which is a complex facility used to chlorinate and then distribute the water coming from the wells to the supply areas (Figure 6). The WSP has also been originally modelled with Revit, but in this case in two separated models (structural and MEP systems), georeferenced in EPSG 7795 and exported in IFC4.

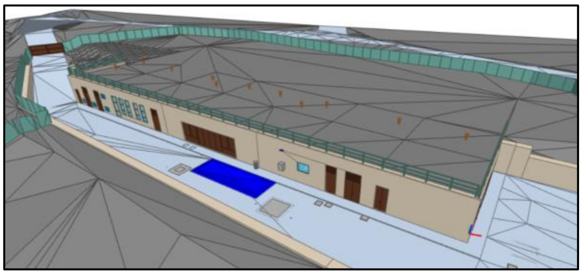

Figure 6. Water Supply Plant

Following the workflow presented before, the models are imported both in MSSQL and PostgreSQL and finally visualised in QGIS. The results of the publication of the models can be seen in the next figures. In particular, in Figure 7, it is possible to see both of the models in their geographical position using OpenStreetMap as base map. In the main QGIS window, the models are displayed in 2D.

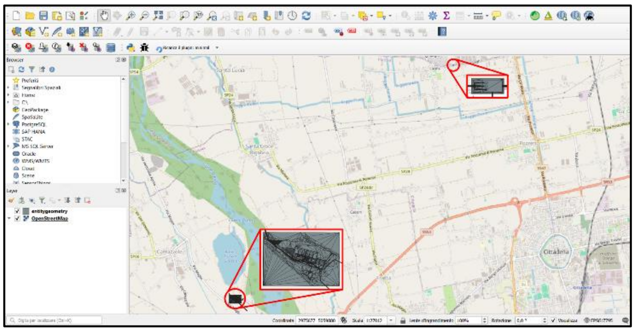

Figure 7. Visualisation of both of the models in QGIS

In Figure 8 and in Figure 9 it is possible to see the 3D geometry of the models directly in QGIS, using the standard 3D visualisation. With regard to the WSP (Figure 8), it is possible to view not only the building but also a portion of the terrain around it, as the BIM model delivered by the company was modelled in this way. The decision to also import the terrain was made in agreement with the company, as it could be useful for future considerations. When it comes to the WWLS (Figure 9), it can be viewed in two different 3D representations. In the top one, the terrain is not displayed so the complete 3D model can be seen, while in the lower one, it is possible to see that the 3D model is correctly underground using OpenStreetMap as reference.

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-XI-4-2026-13-2026 | © Author(s) 2026. CC BY 4.0 License.

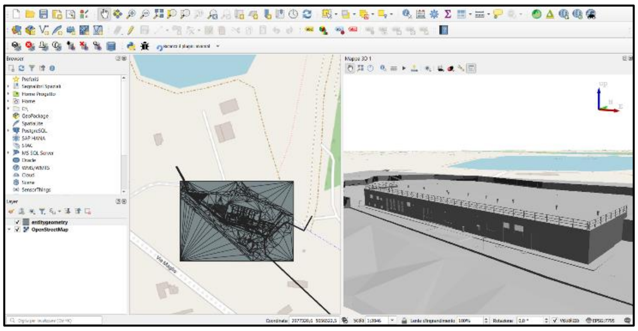

Figure 8. 2D and 3D visualisation of the WSP in QGIS

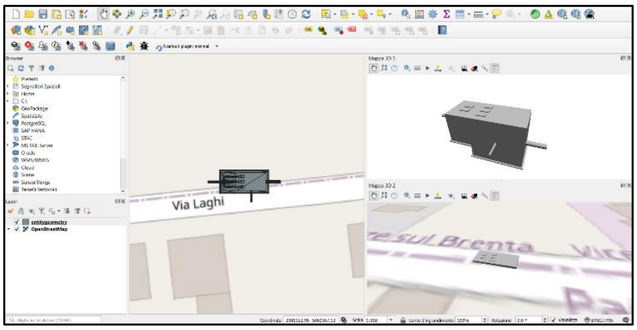

Figure 9. 2D and 3D visualisation of the WWLS in QGIS

3.2. Query examples: Select entities and attributes within a certain area

The main query typologies have been presented in Section 2.2.3. From there, three examples of the first type are tested and implemented, i.e.: Selecting entities and their attributes inside an area (province, municipality, landside risk area, flood risk area, etc.). The aim of this query is to visualise in QGIS the entities belonging to an IFC class and their connected attributes within a specific area. To perform this query, some areas are first needed. For this reason, the company that provided the case studies, also provided Shapefiles with polygons representing the areas of the provinces and the areas of the municipalities in their territory. The files have been imported into PostgreSQL by creating two additional tables in a dedicated database schema. In this way, it is possible to perform the query using all the information directly from the database. The script that we developed in Python is structured to carry out three main steps:

- Select whether to filter by province or municipality.
- Select which specific province or municipality to filter by.
- Select which IFC class to choose from those available in the models within that province or municipality.

The script is briefly explained below. When the area and the IFC class are selected, the script obtains the name of the properties present in the database for the selected class. Then, using the name of the property, it is possible to create a view to fill with the information of each property in the second part of the script. The results are then visible in QGIS with a simple drag and drop from the view. So, for example, if the municipality of "Carmignano di Brenta" and the IFC class "IfcWall" are selected by the user, it is possible to see all the walls in that municipality and the properties of a wall when the user clicks on it. Figure 10 shows the 3D view of the walls in the top, the 2D view at the bottom, and the properties of the wall selected in red on the right.

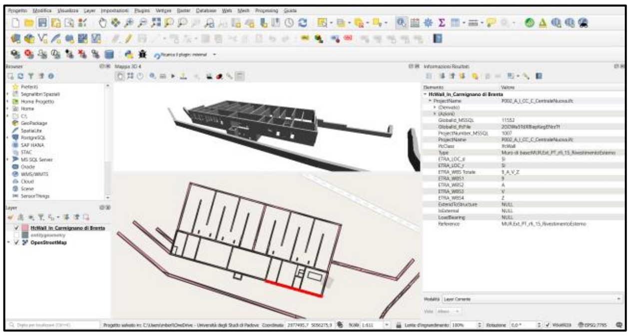

Figure 10. IfcWall in Carmignano di Brenta

Another example can be done with the class "IfcPump", but this time using the complete province of Padova. In this case both models are involved. In Figure 11 it is possible to see in 2D and in 3D the two sets of pumps highlighted for the two models, and also some of the attributes of the pump selected in red. In the two windows at the top, it is possible to see the pumps inside the WWLS, and in the two lower windows the pumps inside the WSP. On the right, the table with the attributes.

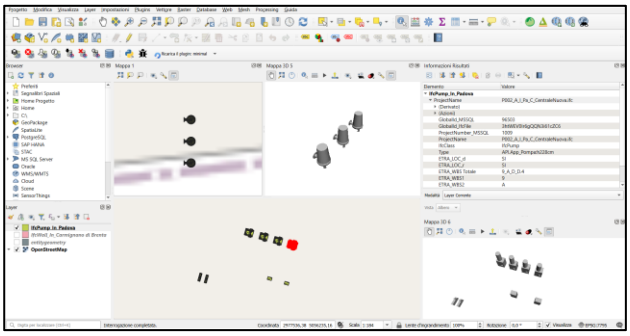

Figure 11. IfcPump in Padova

This type of query works for every BIM entity of a facility but not for a space (IfcSpace). This is because every building element is classified under the "IfcElement" class in the IFC standard. Instead, an "IfcSpace" is classified under the "IfcSpatialElement" class because it describes a space and not a real element. This means that the structure of the attributes of a space is slightly different from those of a wall or a pump. To see the attributes of an "IfcSpace", it is not possible to use the same structure of the table of a building element. So, the script is slightly different for an "IfcSpatialElement" because the queries are designed to retrieve a different structure of attributes. An example of the visualisation of "IfcSpace" in the municipality of "Carmignano di Brenta" is provided in Figure 12. It is possible to see the 3D models of the spaces at the top, the 2D at the bottom and the properties of the space selected in red on the right.

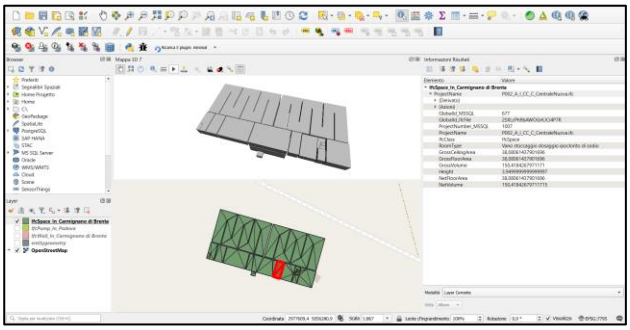

Figure 12. IfcSpace in Carmignano di Brenta

The three examples are made with the area of a municipality or with the area of a province, but if the area is changed the procedure does not conceptually change. So, it is possible also to query the entities in a flood risk area if there is one area available. In addition, there is the possibility to do some other typologies of queries as presented above.

4. Discussion: strengths and limitations

Today IWS companies have the necessity to integrate BIM models of hydraulic infrastructure in a GIS system. This is because the recent growth of the available BIM data has opened up new possibilities to improve the management of hydraulic infrastructures. In fact, being able to see a detailed 3D representation of every entity of a facility in a map, as needed through certain queries, can significantly improve the quality of the management. This is sometimes possible using a proprietary solution, despite some potential limitations (e.g., hidden, undocumented, and non-customizable database structure). However, this task is currently not possible with open-source solutions. In fact, there are open-source solutions that allow one to view a 3D BIM model in a GIS environment, but currently none of them combine visualisation with management of each individual entity via a relational database. Starting from the necessity of BIM and GIS integration through open-source solutions a new methodology has been presented in this paper, and it is now necessary to analyse its current strengths and limitations.

The main strength of this work is that it is possible to integrate IFC models in a GIS map and query them using mostly open-source tools (except, at the moment, for the MSSQL database). As a result, the presented methodology can be checked and further improved in each one of its steps and, wherever necessary, customised.

In this methodology the ifcSQL database is used for the first time for BIM and GIS integration. This is because in the BIM world, the use of standardised relational databases is still limited. Conversely, the use of non-relational databases is quite common in BIM literature. This is easily proven by the amount of scientific research in the BIM world currently focusing on ontologies (e.g. ifcOWL), which serve to structure data in non-relational databases (e.g. graph databases). However, while NoSQL databases have been increasingly adopted in the BIM domain for their flexibility and scalability, their use in the GIS domain remains limited due to insufficient support for complex spatial queries and topological constraints. For these reasons, for a topic such as BIM and GIS integration, the BIM world must move away from a file-based approach toward a more efficient relational database approach.

IfcSQL has proven to be a very interesting technology to manage multiple BIM models at the same time. Nevertheless, ifcSQL can be improved. At the moment, it works only on MSSQL, which represents its major limitation, especially when it comes to 3D geometry support. This limitation currently affects our whole methodology, as it has forced us to work with two different, albeit linked, databases. This solution is acceptable due to the exploratory nature of this research work, but the aim is to manage all the hydraulic infrastructure of a territory with this system, so there is the need to improve it. Therefore, ifcSQL should be ported completely to PostgreSQL in a future work. In this way, there will be the possibility to store the explicit 3D geometry directly in the database but also to simplify overall database setup. Porting ifcSQL to PostgreSQL also requires adapting the accompanying libraries, currently developed in C#. Ideally, they could be ported to Python in order to facilitate interaction and integration with IfcOpenShell, as well as to ensure better compatibility with other tools and libraries that nowadays are mainly developed in and for Python. It is necessary also to highlight that PostgreSQL is completely open-source and MSSQL is proprietary software, so a conversion to PostgreSQL is welcome also for this reason, as it would enable us to have a completely open-source methodology. Furthermore, with PostgreSQL there are many more possibilities to implement functionalities to process geometries, thanks to PostGIS. Finally, having a unique and open-source database in the background to jointly store and manage spatial and non-spatial data greatly simplifies the overall system architecture.

At this moment, the table which contains the IFC geometry in PostgreSQL has only five columns. In a future work, the table could be improved with the addition of other columns. For example, a column containing the number of triangles of the BIM geometry, and another column containing a simplified geometrical representation of the same entity (e.g. if the number of triangles is too high and therefore the geometry too complex).

With the proposed workflow, it is currently possible to perform two main operations. The first is to visualise the 3D geometry of the IFC models in QGIS through the database, even if visualisation management is still limited due to QGIS's standard 3D viewer. For example, rotating the model is difficult and zooming is usually slow. However, (Lutra Consulting, 2025) is now working on a better 3D viewer for QGIS so this limitation might be overcome in the near future. The second is querying the models to obtain results that can simplify the management of IWS. The case study and different types of queries are provided from a real company that manages the IWS in its daily work. So, one of the next steps is to set up the system for the company to receive some additional feedback that can further improve the work.

At the moment, the methodology consists of a set of independent Python scripts that are functional but not yet usable by the final user, and this is a limitation. Consequently, it is necessary to improve the system by better integrating the different pieces of software, and to add a user-friendly Graphical User Interface (GUI). QGIS, for example, has the possibility to extend its functionalities by means of plug-ins, so a future work will consist in creating such a plug-in to interact with the databases directly from QGIS. Some of the envisioned functionalities encompass: A) the possibility to insert an IFC file in the database; B) the possibility to delete a project from the database; C) the possibility to download a project from the database to an IFC-SPF file; D) the possibility of running some predefined queries and seeing the results.

5. Conclusions

Integrated water service (IWS) requires the integration between BIM and GIS to improve the efficiency of the management of the hydraulic infrastructure. This paper has dealt with the need for a standardised and open-source integration, proposing a new workflow for the visualisation and querying of IFC models in a GIS system. The methodology overcomes some of the principal gaps between BIM and GIS, in particular about geometry and data structures, leveraging the ifcSQL relational database. The IFC data has been stored in MSSQL thanks to ifcSQL and the explicit geometry in PostgreSQL thanks to our work. The semantic data in MSSQL and the explicit geometry in PostgreSQL have been merged thanks to a foreign data wrapper, giving the possibility to have all the information accessible from

PostgreSQL, apparently as a unique database environment. The preliminary results have demonstrated the validity of the methodology, showing the 3D geometry of two real case studies in the territorial context on QGIS. In addition, it is possible to execute queries to select some specific entities inside defined administrative boundaries, responding to the specific needs of the IWS companies. Despite the validity of the methodology, there are still some limitations due to its prototypic nature. The main one is the dependence on MSSQL for ifcSQL, which is a proprietary software and does not support 3D geometry. For this reason, there is the necessity to use two separate databases. Then, the workflow is based on a set of Python scripts which are not yet easy to use for the end users. To overcome these two limitations, some future work solutions have already been identified and described in the previous section, i.e. porting of ifcSQL to PostgreSQL and developing of a QGIS plug-in. In addition, another planned improvement will be the improvement of existing queries and the development of new queries, again resulting from interaction and discussion with the end user. An example is the possibility to view the floor plan of a specific floor of the selected facility, which can then be printed in PDF format using QGIS to be used in certain company processes.

This paper contributes to demonstrating how BIM and GIS can be integrated through an open and standardised database implementation. Building on this, the practical contribution consists in a methodology based on the relational database encoding of IFC (ifcSQL) to integrate and access BIM data from an open-source GIS software (QGIS). The system is not only designed for model visualisation but also enables data querying, as demonstrated through real-world examples provided by an Integrated Water Service (IWS) company from Northeast Italy.

Acknowledgements

The authors would like to thank ETRA S.p.A. società benefit, partner of this research, for providing the BIM models and the practical requirements that enabled the validation of this workflow. This research is part of the Italian PhD program funded by Ministerial Decree (D.M.) 117 within the framework of the National Recovery and Resilience Plan (PNRR).

References

Agugiaro, G., Pantelios, K., León-Sánchez, C., Yao, Z., Nagel, C., 2024. Introducing the 3DCityDB-Tools Plug-In for QGIS. *Lecture Notes in Geoinformation and Cartography*. Springer, pp. 797–821. https://doi.org/10.1007/978-3-031-43699-4_48

Azari, P., Li, S., Shaker, A., Sattar, S., 2025. Georeferencing Building Information Models for BIM/GIS Integration: A Review of Methods and Tools. *ISPRS Int. J. Geo-Inf.*, 14. https://doi.org/10.3390/ijgi14050180

Bock, B.S., Eder, F., 2025a. ifcSQL. GitHub. https://github.com/IfcSharp/IfcSQL (10 February 2026).

Bock, B.S., Eder, F., 2025b. IfcSharpLibrary. GitHub. https://github.com/IfcSharp/IfcSharpLibrary (10 February 2026).

Bock, B.S., Eder, F., 2025c. IfcSharpApps. GitHub. https://github.com/IfcSharp/IfcSharpApps (10 February 2026).

buildingSMART International, 2025. IfcTessellatedFaceSet, IFC 4.3 ADD2. buildingSMART International. https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/lexical/IfcTessellatedFaceSet.htm (10 February 2026).

Cepa, J.J., Alberti, M.G., Pavón, R.M., Calvo, J.A., 2024. Integrating BIM and GIS for an Existing Infrastructure. *Applied Sciences,* 14. https://doi.org/10.3390/app142310962

Clemen, C., Görne, H., 2019. Level of Georeferencing (LoGeoRef) using IFC for BIM. *J. of Geodesy, Cartography and Cadastre*. 15-20. https://jgcc.geoprevi.ro/docs/2019/10/jgcc_2019_no10_3.pdf

Dawkins, M., 2024. tds_fdw_16. https://www.dropbox.com/scl/fi/uq40cb1op4c4hqhbqrisv/tds_fdw_16.zip?rlkey=ouytwm23ozsmenbreo0h5p8en&e=2&st=rxnviztg&dl=0 (10 February 2026).

Demir, S., Yomralioglu, T., 2024. Bridging Geo-Data and Natural Gas Pipeline Design Standards: A Systematic Review of BIM-GIS Integration for Natural Gas Pipeline Asset Management. *Energies*, 17. https://doi.org/10.3390/en17102306

Ellul, C., Loo, R., Floros, G., 2024. Exploring the Potential of NoSQL Databases for a BIM and 3D GIS enabled Location Framework for Construction Digital Twins and the Golden Thread. *ISPRS Ann. Phot. Remote Sens. Sp. Inf. Sci*., X-4-W5-2024, 137–145. https://doi.org/10.5194/isprs-annals-X-4-W5-2024-137-2024

ISO 16739-1, 2024. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries - Part 1: Data schema.

ISO 19125-1, 2004. Geographic information - Simple feature access.

Lutra Consulting, 2025. QGIS 3D for Open Source Digital Twins. https://www.lutraconsulting.co.uk/crowdfunding/qgis-3d-for-open-source-digital-twins (10 February 2026).

Noardo, F., Harrie, L., Ohori, K.A., Biljecki, F., Ellul, C., Krijnen, T., Eriksson, H., Guler, D., Hintz, D., Jadidi, M.A., Pla, M., Sanchez, S., Soini, V.P., Stouffs, R., Tekavec, J., Stoter, J., 2020. Tools for BIM-GIS integration (IFC georeferencing and conversions): Results from the GeoBIM benchmark 2019. *ISPRS Int. J. Geo-Inf.*, 9. https://doi.org/10.3390/ijgi9090502

Regione del Veneto, 2025. Il Servizio Idrico Integrato. Regione del Veneto. https://www.regione.veneto.it/web/ambiente-e-territorio/servizio-idrico-integrato (10 February 2026).

Toldo, B.M., Modolo, A., Zanchetta, C., Bock, B.S., 2025. SQL Relational Database Usage for Integration Between BIM Models and Quantity Take-Off Platforms. *New Frontiers of Construction Management*, 165–176. https://doi.org/10.1007/978-3-031-87224-2_

Zhao, L., Liu, Z., Mbachu, J., 2019. An Integrated BIM–GIS Method for Planning of Water Distribution System. *Int. J. Geo-Inf.*, 8, 331. https://doi.org/10.3390/ijgi8080331

Zhu, J., Wu, P., 2022. BIM/GIS data integration from the perspective of information flow. *Autom. Constr.*, 136. https://doi.org/10.1016/j.autcon.2022.104166