TerrainTwin - Microservice Architecture for the Integration of Geodata (GIS) and Building Models (BIM) using Link Models

Sebastian SCHILLING and Christian CLEMEN, Germany

Key words: BIM-GIS Integration, Link Model, Semantic Web, DTM, Digital Twin

SUMMARY

BIM-GIS integration is a long-standing research topic because a complete mapping of information between the domains is not possible. Since the domains partly describe the same objects, e.g. buildings or related infrastructure like underground utilities and construction work, an integration of both domains is a desirable goal. This necessity also arises from the fact that the BIM and GIS worlds are increasingly converging in reality. However, the different view on real world objects causes conflicts, which have to be solved by integration.
Due to the emergence of new technologies, new possibilities are ready to use for the integration of BIM and GIS. One of these new technology areas is the Semantic Web. Although its basic technologies have existed for more than 20 years, they get more and more importance with the interconnection of the world, and this is also true for BIM-GIS integration.
Most approaches in BIM-GIS integration are based on converting data from one data format into a data format from the corresponding other domain. This always results in the loss of information due to the lack of interoperability. Instead of a pure data integration, we also want to include an integration on the application level with our approach in order to minimise information loss due to conversion.
The presented research discusses the idea of a modularised system architecture for the integration of geospatial data and building models, which are linked together using Semantic Web technologies. Several modules of different domain-specific applications are packed as Docker containers and provided in a microservice architecture. For the communication between the applications Application Programming Interfaces (APIs) are designed. The geometries of the BIM and GIS data are stored in domain-specific databases, e.g. PostGIS and BIMserver. The integration of the heterogeneous data is accomplished by a link schema and link instances. All links are collected in a graph database, e.g. GraphDB, with further information. Eventually, the endpoint provides cross-domain queries with the query language SPARQL.
The concrete outcome of the presented research are software components of the microservice architecture "TerrainTwin", however the methods used, design decisions made and functionality achieved are shown with examples in this paper.

TerrainTwin - Microservice Architecture for the Integration of Geodata (GIS) and Building Models (BIM) using Link Models

Sebastian SCHILLING and Christian CLEMEN, Germany

1. INTRODUCTION

BIM-GIS integration is a long-standing research topic because a complete mapping of information between the domains is not possible. The necessity for integration also arises from the fact that the Architecture, Engineering and Construction (AEC) and Geospatial domains are increasingly converging in professional practice. However, the different view on real world objects causes conflicts, which have to be solved by integration.
Due to the emergence of new technologies, new possibilities arise to perform integration between BIM and GIS data. One of these new technology areas is the Semantic Web. Although its basic technologies have existed for more than 20 years, they get more and more importance with the interconnection of the world, and this is also true for BIM-GIS integration.
There are already various methods for BIM-GIS integration, but none of them has yet been able to establish itself. The integration target can be either simple integration of data (BIM → GIS or GIS → BIM) from one domain into the other, or bidirectional integration (BIM ↔ GIS), where data from both domains can be integrated into the other domain. Alternatively, the data can be transferred to an external, neutral domain where all data can exist in a common model.
Although both domains describe our environment, they rely on a different granularity of data due to their field of application. Noardo et al. (2020) compare 3D city models and BIM as representatives of both domains. GIS data are used to describe large-scale phenomena and their interrelationships, such as the built-up area of a city or the network of roads. In the data, objects are described by geometries and enriched with information. Normally GIS data is not showing the structure of the objects in detail. This is in contrast to BIM models describing the detailed structure of individual objects and parameterizing each component. The extended environment of the described object plays only a minor role in BIM. These differences are reflected in the representation of geometries. GIS data is mostly represented as two-dimensional surfaces or three-dimensional boundary representations, whereas in BIM objects are mostly solids, modelled with parametric models, Constructive Solid Geometry (CSG) and Swept Solids. The geometric extent of the data also plays a role. BIM data usually describe an area with an extension of less than 1000 metres. All geometries are described in local coordinate systems with a scale of 1:1. GIS data, on the other hand, often extend over many kilometres, which is why the curvature of the earth must be considered when describing the geometry. The geometries are described in a global, projected coordinate reference system, e.g. the UTM coordinate system, in which all distances must be corrected using a scale.

The integration can be performed at data / instance level or at application level. Both levels of integration are discussed in Beck et al. (2021). At the data level, information is transferred from one domain to the other partly resulting in information loss since the received data cannot be fully interpreted in the target domain.

"Application integration enables applications and systems that were built separately to work together, resulting in new capabilities and efficiencies that cut costs, uncover insights, and much more" (IBM Cloud Education, 2020). At this level, the data from BIM and GIS remain in domain-specific software. Data is accessed via interfaces so that no data is lost. An example of such interfaces here is the creation of APIs (Application Programming Interfaces). In our project, we will perform BIM-GIS integration mainly at the application level, but also combine it with data integrations. In addition, we will use Semantic Web technologies to create interfaces and links.

The result of our pragmatic research is a modular system architecture for the integration and usage of data from the BIM and GIS domains. The modularized design results from the provision of all applications and services as Docker (Docker Inc., 2023) containers, which can communicate with each other and are arranged in a microservice architecture. The advantage of this modularized design is that applications can be added, replaced or removed at will, making the architecture's functions adaptable to new requirements. By providing the interfaces as independent modules, they can be developed without having to change the applications themselves, which also makes them easy to update without having to reinstall the entire infrastructure.
As part of the "TerrainTwin" project, a test architecture was developed to integrate, prepare and provide data from the BIM and GIS domains. The overall aim of the project is to make terrain models, building models and geodata interactively usable for landscape planning in a common model. Users should be able to access and modify all information in an AR/VR environment. The specific prototype, described in this paper, forms the backend of the application, which will be used to import, provide, and technically modify BIM and GIS data.
Based on our test architecture, this paper will discuss the methodologies and design decisions, and demonstrate their functionality and benefit. Our paper aims to answer the following research questions:

- Which system components are needed for a microservice architecture for BIM-GIS integration? How can these system components be classified?
- Which database types are suitable for specific model types and tasks?
- How can heterogeneous data sources and model elements be linked within the system architecture?
- Which ontologies can be used to describe geometries, object information, metadata, and processes?
- How can different digital terrain models (DTM) be used and converted?
- How can the (now) integrated data be used? What is the benefit?

In the next section, existing research is presented before we explain our microservice approach in detail in Section 3. Finally, we discuss the results and provide an outlook.

2. RELATED WORK

Integration is a term that can be interpreted in many ways and assumes a variety of definitions in the context of software development and data exchange (Gulledge, 2006). In our research, we use data integration to convert BIM and GIS data, together with linking approaches for the integration on application level. Data integration is about converting data of one format into the schema of another format. Application integration is about creating interfaces so that one software can access the data of another software and interpret it in its domain. In the following, various integration approaches from research in the context of BIM-GIS integration are presented.

In the literature there are many versatile approaches for the integration of BIM and GIS data. On the one hand, data is converted unidirectionally from one domain to the other (BIM → GIS, GIS → BIM). Examples of this can be found in Laat and van Berlo (2011), Zhu et al. (2021) and Ding et al. (2020). Secondly, there are approaches for converting the data in both directions (BIM ↔ GIS). A third approach converts the data from both domains into a domain-unspecific format and merges them into a unified model (El-Mekawy et al., 2012).

However, due to the different areas of operation, complete interoperability cannot be achieved with any approach, i.e. there is always a loss of data during conversion. The integration of the data is mainly done by mapping on the data or schema level. In these approaches, the data formats are defined by the integration conditions and can only be integrated from one to exactly one other format. A comprehensive overview on current standardization efforts to overcome interoperability domains is given in the ISO/TR 23262:2021 GIS (geospatial) / BIM interoperability (Clemen, 2022).

Most practical studies use the most representative formats for their respective domains, IFC and CityGML. Between their schemas, a mapping of entities describing the same object is performed (e.g. in Deng et al. (2016), Laat and van Berlo (2011), Noardo et al. (2020) and Vilgertshofer et al. (2017)). But other GIS formats such as shapefiles are also used (Zhu et al., 2021). In most cases, the data of one format is converted into a format of the respective other domain, so that a new conversion is necessary if the source data is changed.

The majority of the current studies use Semantic Web technologies for the conversion. Mapping can be done, for example, via the ontologies of the converted data schemas. Relationships are established between entities that describe the same real-world object. The relationships are created either manually (Vilgertshofer et al., 2017) or via semantic text analysis tools (Ding et al., 2020).

However, this approach is usually not completely automatable and leads to data loss because the domains have a different scope, different reference systems and different terminologies (Ding et al., 2020). No 1:1 mapping of the information is possible.

Roxin and Hbeich (2019) suggest two approaches that can be used to create a uniform, common model. Based on these, matches between BIM/GIS concepts and models should first be found together with BIM and GIS experts.

The approaches presented so far have mainly prioritised the conversion of information from one schema to another, which is called data integration. In application integration, however, the focus is on communication between domain-specific software that implements the

corresponding domain models. An interface is created, that allows software clients to access the non-domain-specific information in another software and interpret it in its own domain.
In our project we use both, data and application integration approaches to develop a software architecture that meets the requirements of the AEC and geospatial domain.

Semantic Web methods also enable the conversion of BIM and GIS data and schemas into the domain-independent Resource Description Framework (RDF) (RDF Working Group, 2014), mainly in Turtle (W3C, 2014) and XML/RDF syntax. This approach is used for integration in Vilgertshofer et al. (2017) and Karan et al. (2016). If all information is available in RDF, it can be combined into a uniform model.
The RDF models are stored e.g. in a graph database where the information can be queried. Graph databases are non-relational databases in which the information is stored as triples that together form a graph (Oracle, 2022). Unlike object-relational databases, the data are not stored in a fixed schema. This means that semantically much more complex facts can be mapped and queried. Implicit information that is not modelled in this way can also be obtained from the data.
SQL-like query languages have been developed for querying information in graph databases. SPARQL (W3C, 2013) and Cypher (Neo4j Inc., 2023a) are query languages that allow the user to make queries of any complexity without having to know the schema of the complete database.

Semantic Web technologies are also used to convert BIM data into RDF. The vendor neutral schema of the Industry Foundation Classes (IFC) provides the most used exchange format in the BIM domain. For usage in the Semantic Web the ifcOWL ontology has been standardised, Based on this ontology, the open source converter IFCtoRDF by Pauwels (2021) and Pauwels et al. (2016) was developed and is used in our microservice architecture.

The approach of using external link models does not require the conversion of data. The multi-model concept (proposed by Fuchs (2015)), brings together the data of the different domain models involved in the building process.
"Multi-models bundle subject models of different domains and allow the connection of their elements in external link models. The aim is to bring together what were originally separate information spaces and make them interchangeable between project participants" (Fuchs, 2015).
Similar to this, the Information Container for Linked Document Delivery (ICDD) was developed and standardised as BIM Standard in ISO 21597-1 (2018). This standard describes a container for the exchange of documents, in which documents from different disciplines can collected and linked with Semantic Web methods.

Essentially, two components are needed to implement an integration on application level. First, a software for the subject-specific data from each domain and, secondly, software that enables one software to access the data of the other. This is also called middleware. It can either be part of the specialised software or stand alone and, for example by providing APIs for communication.

In our investigation, we want to connect more than two pieces of software and develop more middleware. To manage this amount of software, we are using a so called microservice architecture (Google Cloud, 2022). Microservices are small, independent pieces of software that can be linked together to achieve common goals. This modular architecture has the advantage that microservices can be easily added, updated, tested and deleted. Communication between microservices is done via http queries and can be configured so that only specific services can make requests. The services can be used multiple times as containers, either to run processes in parallel instead of sequentially, or to ensure high availability of the application by redundancy.

Schilling und Clemen (2022a) explain how geodatabases and triplestores can be used together for BIM-GIS integration. We use the developed open source microservices as a starting point to develop the microservice architecture described below.

3. THE TERRAINTWIN ARCHITECTURE

In this section we want to present the developed microservice architecture, which is called TerrainTwin architecture. Obviously digital terrain models are very important for the intended use case "interactive planning of wind turbines in a VR/AR environment” and other landscape planning tasks resulting in the name TerrainTwin. The software components of the microservice architecture are published on GitHub (Schilling & Clemen, 2022b).

3.1 Methodology

As shown in the workflow diagram (Figure 1) the resulting microservice architecture was developed use case driven and iterative. The use case was specified and discussed continuously with the project partners. It was determined which software components are required, which geospatial and building models are imported into the system, which conversion processes have to be triggered automatically, how the heterogeneous data is linked (schema and instance level), and how the microservices are orchestrated.

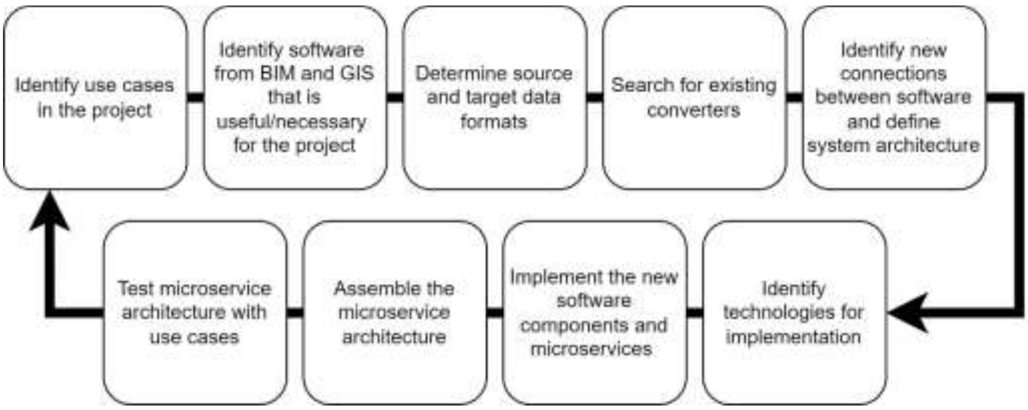

Figure 1: Workflow of the development of the TerrainTwin Microservice Architecture

3.2 Overall architecture and classes of system components

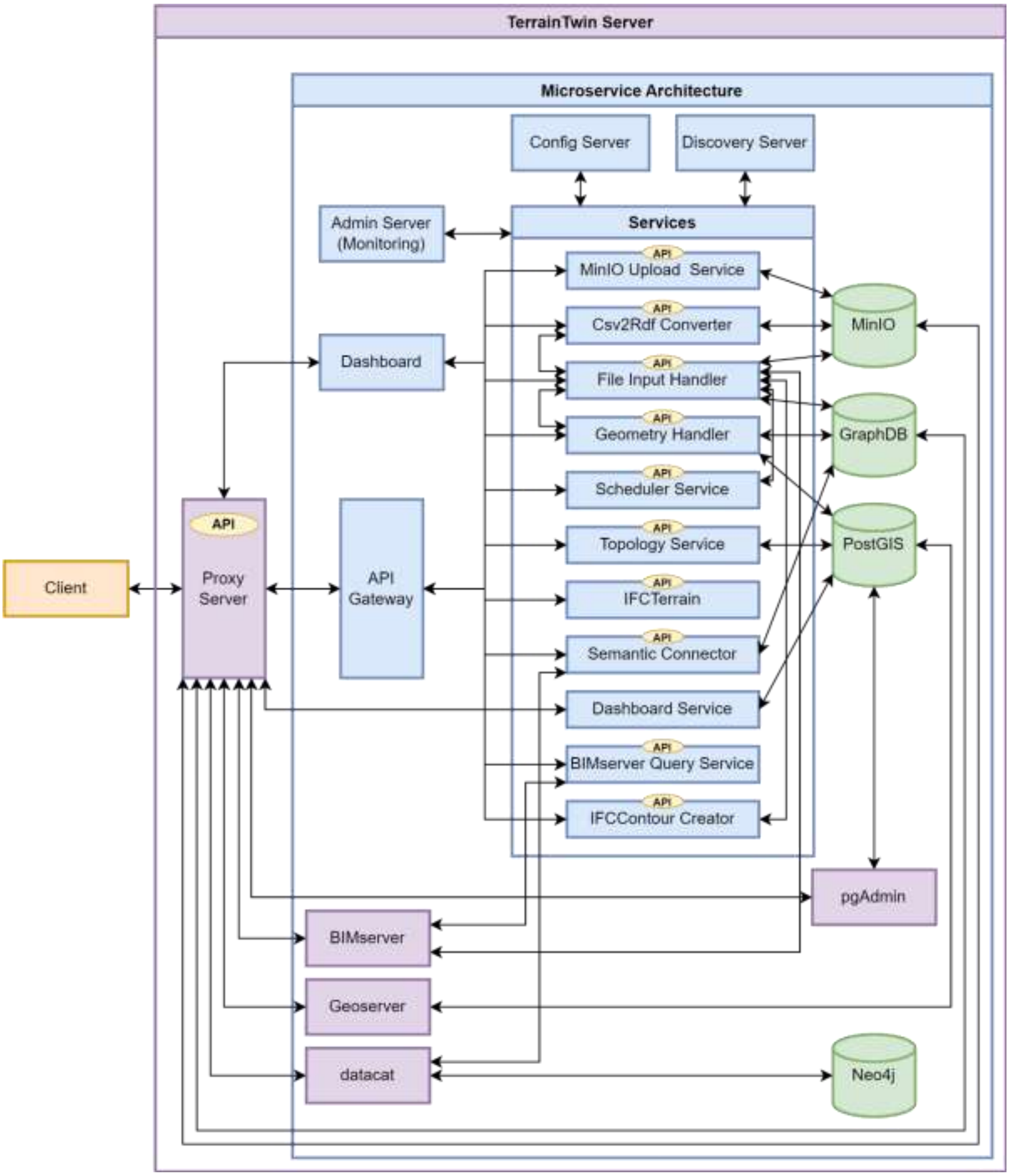

Figure 2: The TerrainTwin Microservice Architecture on a Server

Figure 2 shows the final architecture with its individual components as we installed it on a server. All blue components were developed during the project. The components with a yellow oval, provide a simple REST-API (Representational State Transfer API). Green components are databases and purple components are domain-specific software components. The components can also be divided into the following categories according to their functionality:

- Applications for permanent storage of domain-specific geodata and file storage (BIMserver (The open source BIM collective, 2022), PostGIS (POSTGIS, 2022), MinIO (MinIO Inc., 2023)).
- Applications as interfaces for the input and delivery of information
- Applications for generating, editing, processing, converting data
- Applications for external storage of permanent link sets (GraphDB (Ontotext Inc., 2022), Neo4j (Neo4j Inc., 2023b))
- Applications for monitoring system components and data (Admin Server, Dashboard, Dashboard Service)

3.3 Design decisions and methods

The programming of new system components was mainly done with Java's Spring Boot Framework or with C#. The final applications were packaged in Linux Docker containers. For the processing of geometries and in particular geometries of terrain models, various software libraries were examined and tested. The Java Topology Suite (JTS) (Locationtech, 2022) is only partially suitable for use in the project because it was not originally intended for use with 3D geometries. However, there are a large number of functions in it for processing 2.5D and 3D geometry in recent releases. The NetTopologySuite (NTS) (NetTopologySuite, 2022) is very well suited for C# development, whereby the JTS also serves as a basis here.

One of the core components of our architecture are application programming interfaces (APIs). We need them for the effective and secure communication between our applications. The OpenAPI Specification (OAS) (OpenAPI Initiative, 2021) is used for the technical implementation of REST-APIs.
OGC's API Feature Standard (Open Geospatial Consortium, 2019) can be used for the standard-compliant description and use of geometry elements. Among other things, the standard describes how an API should be structured in order to make geometries available. Since the intended structure of frontend and backend requires the transfer of geometries to each other, such an API is a good solution.

For the project, it was analysed in which data formats and structures digital terrain models are created and transferred. It was found that digital terrain models can exist in a variety of different data formats, most of them originate from the GIS domain. The data format is strongly related to the acquisition method of the terrain. Examples of acquisition methods include satellite-based radar imagery, terrestrial and airborne laser scanning, and tachymetric acquisition by surveyors. Examples of data formats are:

- Text Files containing semantic and geometric information: *.TXT/*.CSV/*.XYZ
- Industry Standards for Drawings (*.DXF) or DTMs (*.LandXML, *.REB, *.DA45, *.DA49, *.DA58)
- Information Models (*.CityGML, *.IFC, *.GeoJSON)
- Specific national CAD formats (*.OUT by Trimble/HHK)

For the identified formats for terrain models, the tool IfcTerrain (DD-BIM, 2022) was developed, with which the terrain models can be converted into IFC, georeferenced and enriched with metadata. Using this tool, the GIS to BIM data integration takes place via the classic conversion between domain-specific data formats. The core of IfcTerrain has already been presented in previous works (Clemen et al., 2021). Within the TerrainTwin project it is implemented as an additional microservice with an API to make terrain models available for BIM projects in parallel to their storage in the PostGIS database.
The storage of terrain models in PostGIS is not possible from all formats without further work. In order to be able to read in different file formats, an API was developed which converts the geometry presentations into the WKT string of the TIN type for importing this uniformly. To ensure that the geometries are available in a uniform coordinate system, a transformation into the defined PostGIS coordinate system takes place during the import with functions of the database, if necessary. In the API, the user can specify any coordinate system for the output of the geometries.

In the context of the Semantic Web, there are also several ontologies that can be used to describe geometries. For example, the NeoGeo Vocabulary Ontology, the WGS 84 Ontology, the Geonames Ontology and the GeoSPARQL Ontology (OGC, 2012) can be mentioned here. Most ontologies so far only support simple coordinate specifications in the WGS 84 coordinate system. The GeoSPARQL ontology is an exception here. It is based on the standard "GeoSPARQL - A Geographic Query Language for RDF Data" of the OGC (2012). With this ontology, geometries can not only be described, but also query results can be declared. GeoSPARQL provides simple function declarations for topological and spatial calculations that can be executed using queries.
From the BIM domain we use the ifcOWL ontology, which describes the conceptual model of IFC and can be used to convert IFC files to RDF. Here we use the IFC to RDF converter from Pauwels (2021).

The accompanying performance tests have shown that graph databases are unsuitable for complex geodata. Therefore, only the semantic information and links between data sets are stored in the graph database, while the geometries are stored in object-relational databases. In the graph database, a link is added to each geometric representation in the PostGIS database. IFC models are stored externally in a domain-specific database, the BIMserver, and are also made accessible via a link.
The Geoserver is only suitable to a limited extent for providing the geometries in the project, as it cannot process 3D geometries, and is therefore only used to a small extend in the presented microservice architecture.

3.4 Adding metadata

To ensure the quality of the terrain models and geodata, metadata according to the DIN SPEC 91391-2 and DIN 18740-6 standards can be added to each file to be used in the project via a user interface. DIN 18740-6 is specifically intended for the description of digital elevation models, while the second DIN defines general metadata for all files. The framework conditions for DIN SPEC 91391 are set by the international standard ISO 19650. All metadata for a file is stored uniformly in a file in JSON format, uploaded via an API and converted into RDF to be subsequently imported into the graph database. The additional input can be used, for example, to store the topicality, version, geometric accuracy and creator of the data. This information can be used in the further processing of the data or for filtering.

3.5 Creation and use of link models and ontologies

For the creation of a link model, the TerrainTwin ontology was developed for the project. Some of the contents are described below in the presentation of the linking.

The first task was to make the geometry data in the PostGIS database accessible from outside. For this purpose, an API was created based on the OGC API Features (2019) , which can be used to query geometries by their unique identifier. The identifier in the URL also makes it unique and can be stored in the graph database as a link to the geometry, which returns the geometry when executed (Figure 3).

https://geodatabase.org/collections/polygon_2d/items/ce8007d3-0798-4712-894a-8ce0c071300d


```
{
  "id": "ce8007d3-0798-4712-894a-8ce0c071300d",
  "geometry": "SRID=25832;POLYGON((685636.701199641 5646606.3536494,
              685686.011941892 5646602.44895838,685718.819502458 5646553.76775835,
              685713.569069366 5646496.58646396,685652.508571504 5646448.69792641,
              685605.234283641 5646452.58222739,685576.576094471 5646496.59665898,
              685596.186207196 5646575.49078808,685636.479635002 5646606.11546631,
              685636.701199641 5646606.3536494))"
}
```


Figure 3: Link to geometry in PostGIS DB and resulting geometry from execution over the API

In the graph database, the GeoSPARQL ontology was used to declare the geometry objects. For our needs, GeoSPARQL is extended, for example with a predicate "url" and a class "GeoLink", to link the geometry in the domain specific databases PostGIS and BIMserver.

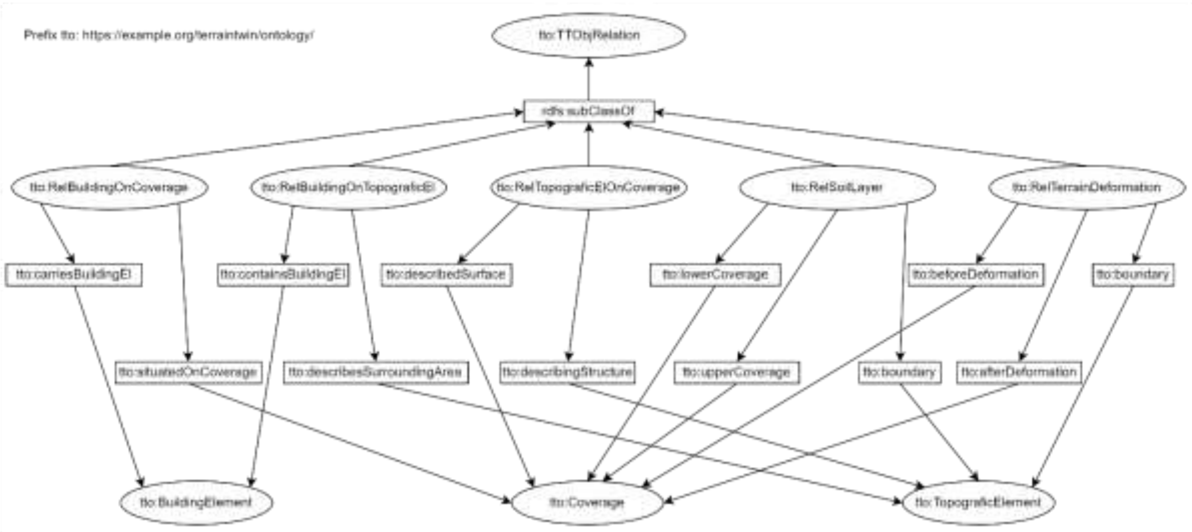

Figure 4: Part of the TerrainTwin Ontology (tto)

Once the information from the files has been sorted into the appropriate storages, it can be linked together. The information is roughly divided into three classes and links are created between them (Figure 4). Terrain models are defined as "Coverage", IFC models as "BuildingElement" and other geometries as "TopograficElement". Links can be defined between them, for example, that a BuildingElement is located on a Coverage. In general, links are mainly created between document – element and element – element in order to achieve the most precise, detailed linking of information. The classes and link structure was defined in the ontology.
First, the IFC models were completely converted and stored in the triplestore during import using the IFCtoRDF converter by Pauwels (2021) and the ifcOWL ontology. However, storing the IFC geometries in the ifcOWL ontology is not very efficient, which unnecessarily increases the number of triples in the Triplestore. The test on some converted IFC files has shown that up to 90 % of the triples are used for geometry description.
The complex description of the geometries also makes them difficult to query. Therefore, the final decision was made for a similar procedure as for the other geometries. IFC models are stored externally in a domain-specific database, the BIMserver. For linking with other information, the ontology is extended so that an IFC model is always linked to a corresponding 2D polygon of its footprint. In addition, metadata of the BIMserver project, e.g. "ifcProjectId" and "ifcRevisionId", are included in the ontology.

For quick access to the non-geometric data of the IFC files, e.g. the "PropertySets", a copy is created when importing an IFC file, in which all geometry elements and associated descriptions are removed. This file with the PropertySets is then converted to RDF and stored in the Triplestore. Via the BuildingElement instance of the IFC file, the PropertySets are further linked to the externally stored complete IFC file.

3.6 Spatial queries on spread data sources

With our TerrainTwin architecture, cross-domain queries can be answered using the query language SPARQL (SPARQL Protocol and RDF query language). Two possible questions from landscape planning are presented here to demonstrate the functionality of the architecture. The SPARQL query in Figure 5 can be used to ask the following question:

- Which kinds of land use are affected by constructing the building? Give the geometry of these land uses.

This query looks for all buildings whose footprint has a topological relationship to at least one geometry representing a land use type. The result of the query is a table (Figure 6) containing the buildings with the corresponding land use type and its link to the geometry in the PostGIS database.

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX tto: <https://terrain.dd-bim.org/terraintwin/ontology/>
PREFIX sd: <https://terrain.dd-bim.org/Sachdaten/>
select ?building ?url ?landUseType where {
    ?building tto:hasFootprint / geo:hasGeometry / geo:sfIntersects ?featureGeom .
    ?featureGeom ^geo:hasGeometry ?feature.
    ?featureGeom tto:url ?geomUrl .
    ?feature tto:hasSource / ^tto:hasSource ?landUse .
    ?landUse sd:id ?id .
    ?featureGeom tto:originId ?oId .
    ?landUse sd:Nutzungsart ?landUseType .
    Bind(str(?geomUrl) as ?url)
    Filter(?id = str(?oId))
}
```

Figure 5: SPARQL query - Which kinds of land use are affected by constructing the building? Give the geometry of these land uses.

	building	url	landUseType
1	bim:7181769e-ea89-4bbf-94c5-d089725b2079	"https://terrain.dd-bim.org/geometry/export/collections/polygon_2d/items/2143723d-c970-4524-a4d5-6792f857e842"	"Greenland"
2	bim:7181769e-ea89-4bbf-94c5-d089725b2079	"https://terrain.dd-bim.org/geometry/export/collections/polygon_2d/items/b78df06b-2814-4627-8750-858051a3f6a2"	"Forest"

Figure 6: Resulting table of the SPARQL query with building entity, link to land use geometry and land use type

The information to answer the question comes from both, BIM and GIS sources. The land use types are 2D polygons with associated technical information exported from a GIS. The building is a georeferenced BIM model exported as an IFC file, from which a 2D footprint geometry was calculated during import into the microservice architecture.

In another scenario, an excavation pit has already been created and a new digital terrain model was created including these changes. For example the following question might arise:

– How big is the excavation of the building pit and which building should be built there? Where do the building data come from and which IFC project number does the building have? (Figure 7)

In this case, the instance of the new digital terrain model is used as the starting point. When saving a new digital terrain model, the order and removal are determined as the mass difference to the previous digital terrain model and saved in GraphDB as a triple. The value of the removal is first queried here. Then the building is searched for which has a spatial relationship with the digital terrain model of the excavation. The link to the source file in the MinIO Object Storage and the ifcProjectId of the project in the BIMserver are output from this (Figure 8). This query shows that calculation results can be queried, source files can be searched for and references to other platforms for visualising the information can be output.

```
PREFIX tto: <https://terrain.dd-bim.org/terraintwin/ontology/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX postgres: <https://terrain.dd-bim.org/postgres/>
select ?excavation ?bimSource ?projectId where {
    VALUES ?dtm {postgres:4fe9ade4-9a48-46bb-b23a-a3109d8b33a8}
    ?dtm ^tto:input / tto:excavation ?excavation .
    ?bim tto:hasFootprint / geo:hasGeometry / geo:sfIntersects / tto:bounds ?dtm .
    ?bim tto:hasSource ?source .
    ?bim tto:ifcProjectId ?projectId .
    Bind(str(?source) as ?bimSource)
}
```

Figure 7: SPARQL query - How big is the excavation of the building pit and which building should be built there? Where do the building data come from and which IFC project number does the building have?

	excavation	bimSource	projectId
1	"1.087827E3"^^xsd:double	"https://terrain.dd-bim.org/minio/jena3/2022-07-08T08-27-45_Haus2.ifc"	"3211265"^^xsd:long

Figure 8: Resulting table of the SPARQL query with excavation, link to source file and project id in BIMserver

In principle, the use of the SPARQL query language makes it possible to answer any question as long as the required linking is implicit or explicit. This has the advantage that, theoretically, individual questions of a project can also be answered. Repeating queries, have been standardised with an API to make them easily accessible to users. This is because the disadvantage of SPARQL is that, due to its high complexity, a minimum level of expertise in both the information and the query language must be present among users.

4. CONCLUSION AND OUTLOOK

The aim of this practical research was to exemplify the benefits of Semantic Web technologies and microservice architectures for the integration of BIM and GIS. The major aspects to reach this goal were following three considerations:

1. SPARQL is a very flexible Query Language. The endpoint provides a unique façade for heterogenous geospatial (GIS) and construction (BIM) domain. In contrast to converters that transfer information from one format to another format of the respective domain our microservice architecture creates links and keeps original data sets. In this way information loss can be minimized and the data can be accessed format-independently using RDF.
2. The concept of linking is essential. Links between documents and elements of different documents provide the backend structure for queries.
3. Links are essential, however creating and maintaining link instances is very challenging. Some of the presented microservices create links to elements between databases. While the links are persistent in a graph database, the domain specific entities are handled in domain databases like PostGIS (geodata, e.g. terrain), BIMServer (information models of buildings and other built assets) or object storage (documents, raw data). Their additional functionality, like georeferencing, geometric queries or built-in queries on IFC relations is mature and well developed.

The more data input is originally given with RDF, the fewer specialised data stores and applications need to be incorporated into the architecture. Until now, we used and developed only a small set of converters to RDF. In the research project, the architecture was designed for specific aspects of landscape and wind turbine planning. For other use cases, the link model ontology, link instance creator and RDF converter have to advanced still very extensive.

Integrating Geospatial Information to the project data environment, is rather easy, geospatial data sets are made available via the web in many ways, including RDF. However, this is not case for BIM data. The terrain twin architecture therefore helped to publish linked models (GIS and BIM) to project the members. Figure 9 shows a scene in landscape planning, where the recent microservice architecture backend was used by an VR frontend. As many other researches, we can state, that Semantic Web and microservices are a powerful team for data integration, with demanding, heterogeneous information sources.

Figure 9: VR model using our resulting microservice architecture for landscape planning

ACKNOWLEDGEMENTS

This research was funded by the Federal Ministry for Economic Affairs and Energy (BMWi), Central Innovation Programme for small and medium-sized enterprises (SMEs), Funding No. 16KN086446 (”TerrainTwin”, 06/2019 - 07/2021) and accomplished with our project partner LandPlan OS GmbH in Osnabrück, Germany.

REFERENCES

Beck, S. F., Abualdenien, J., Hijazi, I. H., Borrmann, A [André] & Kolbe, T. H [Thomas H.] (2021). Analyzing Contextual Linking of Heterogeneous Information Models from the Domains BIM and UIM. *ISPRS International Journal of Geo-Information*, *10*(12), 807. https://doi.org/10.3390/ijgi10120807

Clemen, C. (2022). TRENDS IN BIM AND GIS STANDARDIZATION – REPORT FROM THE JOINT ISO/TC59/SC13–ISO/TC211 WG: GIS-BIM. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVI-5/W1-2022*, 51–58. https://doi.org/10.5194/isprs-archives-XLVI-5-W1-2022-51-2022

Clemen, C., Schröder, M., Kaiser, T. & Romanschek, E. (2021). IFCTERRAIN – A FREE AND OPEN SOURCE TOOL TO CONVERT DIGITAL TERRAIN MODELS (DTM) TO OPENBIM INDUSTRY FOUNDATION CLASSES (IFC). *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *VIII-4/W2-2021*, 145–151. https://doi.org/10.5194/isprs-annals-VIII-4-W2-2021-145-2021

DD-BIM. (2022). *IfcTerrain*. https://github.com/dd-bim/IfcTerrain

Deng, Y., Cheng, J. C. & Anumba, C. (2016). Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. *Automation in Construction*, *67*, 1–21. https://doi.org/10.1016/j.autcon.2016.03.006

Ding, X., Yang, J., Liu, L., Huang, W. & Wu, P. (2020). Integrating IFC and CityGML Model at Schema Level by Using Linguistic and Text Mining Techniques. *IEEE Access*, *8*, 56429–56440. https://doi.org/10.1109/ACCESS.2020.2982044

Docker Inc. (2023). *Docker*. https://www.docker.com

El-Mekawy, M., Östman, A. & Hijazi, I. (2012). A Unified Building Model for 3D Urban GIS. *ISPRS International Journal of Geo-Information*, *1*(2), 120–145. https://doi.org/10.3390/ijgi1020120

Fuchs, S. (2015). *Erschließung domänenübergreifender Informationsräume mit Multimodellen: Access of cross-domain information spaces using multi-models* [Dissertation, Technische Universität Dresden, Dresden]. RIS. https://core.ac.uk/download/pdf/236372744.pdf

Google Cloud. (2022). *What Is Microservices Architecture?* https://cloud.google.com/learn/what-is-microservices-architecture

Gulledge, T. (2006). What is integration? *Industrial Management & Data Systems*, *106*(1), 5–20. https://doi.org/10.1108/02635570610640979

IBM Cloud Education. (2020). *Application Integration*. https://www.ibm.com/cloud/learn/application-integration

ISO. (2018). *ISO/DIS 21597-1:2018-09 - Entwurf Informationscontainer zur Datenübergabe - Austausch-Spezifikation - Teil 1: Container*. https://www.beuth.de/de/norm-entwurf/iso-dis-21597-1/295980311

Karan, E. P., Irizarry, J. & Haymaker, J. (2016). BIM and GIS Integration and Interoperability Based on Semantic Web Technology. *Journal of Computing in Civil Engineering*, *30*(3), Artikel 04015043. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000519

Laat, R. de & van Berlo, L. (2011). Integration of BIM and GIS: The Development of the CityGML GeoBIM Extension. In *Advances in 3D Geo-Information Sciences* (S. 211–225). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12670-3_13

Locationtech. (2022). *JTS Topology Suite [Computer software]*. Locationtech. http://locationtech.github.io/jts/
MinIO Inc. (2023). *Multi-Cloud Object Storage*. https://min.io
Neo4j Inc. (2023a). *Cypher Query Language*. https://neo4j.com/developer/cypher/
Neo4j Inc. (2023b). *Neo4j Graph Data Platform*. https://neo4j.com
NetTopologySuite. (2022). *NetTopologySuite [Computer software]*. https://github.com/NetTopologySuite/NetTopologySuite
Noardo, F., Harrie, L., Arroyo Ohori, K., 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 International Journal of Geo-Information*, *9*(9), 502. https://doi.org/10.3390/ijgi9090502
OGC. (2012). *OGC GeoSPARQL - A Geographic Query Language for RDF Data*. OGC. https://www.ogc.org/standards/geosparql
Ontotext Inc. (2022). *Ontotext GraphDB*. https://www.ontotext.com/products/graphdb/
Open Geospatial Consortium. (2019). *OGC API - Features - Part 1: Core*. http://docs.opengeospatial.org/is/17-069r3/17-069r3.html
The open source BIM collective. (2022). *BIMserver*. https://github.com/opensourceBIM/BIMserver/
OpenAPI Initiative. (2021). *OpenAPI Specification v3.1.0*. https://spec.openapis.org/oas/latest.html
Oracle. (2022). *What is a Graph Database?* https://www.oracle.com/autonomous-database/what-is-graph-database/
Pauwels, P. (2021). *IFCtoRDF* [GitHub]. https://github.com/pipauwel/IFCtoRDF
Pauwels, P., Oraskari, J. & et. al. (2016). *IFC-to-RDF-converter*. https://github.com/aothms/IFC-to-RDF-converter
POSTGIS. (2022). *PostGIS*. https://postgis.net
RDF Working Group. (2014). *Resource Description Framework (RDF)*. https://www.w3.org/RDF/
Roxin, A. & Hbeich, E. (2019). *Semantic interoperability between BIM and GIS – review of existing standards and depiction of a novel approach*. https://hal.archives-ouvertes.fr/hal-02279633/
Schilling, S. & Clemen, C. (2022a). PRACTICAL EXAMPLES ON BIM-GIS INTEGRATION BASED ON SEMANTIC WEB TRIPLESTORES. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVI-5/W1-2022*, 211–216. https://doi.org/10.5194/isprs-archives-XLVI-5-W1-2022-211-2022
Schilling, S. & Clemen, C. (2022b). *TerrainTwin*. https://github.com/dd-bim/TerrainTwin
Vilgertshofer, S., Amann, J., Willenborg, B., Borrmann, A [A.] & Kolbe, T. H [T. H.] (2017). Linking BIM and GIS Models in Infrastructure by Example of IFC and CityGML. In K.-Y. Lin, N. El-Gohary & P. Tang (Hrsg.), *Computing in civil engineering 2017: Information modeling and data analytics : selected papers frm the ASCE International Workshop on Computing in Civil Engineering 2017, June 25-27, 2017, Seattle, Washington* (S. 133–140). American Society of Civil Engineers. https://doi.org/10.1061/9780784480823.017

W3C. (2013). *SPARQL 1.1 Query Language*. https://www.w3.org/TR/sparql11-query/
W3C. (2014). *RDF 1.1 Turtle*. https://www.w3.org/TR/turtle/
Zhu, J., Tan, Y., Wang, X. & Wu, P. (2021). BIM/GIS integration for web GIS-based bridge management. *Annals of GIS, 27*(1), 99–109. https://doi.org/10.1080/19475683.2020.1743355

BIOGRAPHICAL NOTES

Sebastian Schilling, born 1996. Graduated in 2019 as B.Eng. in Surveying and 2020 as M.Eng. in Geomatics/Management from the HTW Dresden. He received the Trimble Award in 2020 for his outstanding degree. Since then he is research assistant at the Faculty of Spatial Information at HTW Dresden.

Christian Clemen, born 1976. Graduated in 2004 as a Dipl.-Ing. in Surveying from the Technical University of Berlin. From 2004 to 2010 he was an Assistant at the Department of Geodesy and Geoinformation, Technical University of Berlin. Since 2013 he has been a full professor at the HTW Dresden, University of Applied Sciences. He teaches adjustment calculation, CAD and BIM. His research topic is BIM from a geomatics perspective.

CONTACTS

Sebastian Schilling
Research Assistant, Faculty of Spatial Information
HTW Dresden, University of Applied Sciences
Friedrich-List-Platz 1
D-01069 Dresden, Germany
E-Mail: sebastian.schilling@htw-dresden.de

Prof. Dr.-Ing. Christian Clemen
Surveying and BIM, Faculty of Spatial Information
HTW Dresden, University of Applied Sciences
Friedrich-List-Platz 1
D-01069 Dresden, Germany
E-Mail: christian.clemen@htw-dresden.de