Multi-platform data search and access method to compose digital twins using metadata

Ryota Yabe

Intelligent Systems Laboratory, SECOM CO., LTD

ryo-yabe@secom.co.jp

Abstract

Various real-world applications are implemented to make life comfortable, convenient, and safe. Data and applications are tightly coupled. Application implementation costs are high because the data reusability is low. Data reusability is increased by implementing a mechanism that collects data according to the requirements of the real-world application to be implemented, builds a digital twin from the collected data, and combines the data with real-world applications. Therefore, platforms that provide data to creating digital twins are emerging. However, digital twin data platforms have the following problems: As digital twin data platforms proliferate, they are being distributed across multiple platforms. Consequently, obtaining data to create digital twin is difficult for real-world application implementers. Data reprocurement increases application implementation costs. In this study, methods for solving data reusability problems are proposed. A function to output metadata representing the spatial structure was implemented on spatial data platforms. Data delivery interfaces are implemented in data platforms that manage data related to spatial data. The method for accessing related data is described in the Metadata section. In this study, prototype systems for BIM data were implemented based on the proposed method. The IFC model server, which exports metadata, was implemented. The API Server, which manages the building of the 3D model data, was implemented. An endpoint for accessing the 3D data API server is described in the metadata.

Keywords

Digital Twin, BIM, IFC, IFC model server, Linked Data, metadata

1. Introduction

Various real-world applications have been implemented to improve comfort, convenience, and safety. For example, autonomous cars can reduce congestion and resolve traffic bottlenecks in the mobile sector. In the field of disaster prevention, the formulation of evacuation plans using damage-prediction simulations also increases safety. Thus, the implementation of various real-world applications to solve tasks is important. The implementation of real-world applications is becoming increasingly important with the development of smart cities and buildings.

Currently, the high cost of data generation is a barrier to the implementation of real-world applications. Reducing the cost of creating data to implement real-world applications is difficult, because creating data requires surveying and drawing. Therefore, reusing existing data is necessary to reduce the cost of implementing real-world applications. However, reusing data to reduce the cost of data creation is difficult. Real-world data are used to implement specific real-world applications, and are not expected to be reused in other real-world applications. Therefore, data must be created each time real-world applications are implemented. The tight coupling between data and real-world applications inhibits data reuse. The Smart City Guidebook [1] published by the Japanese Cabinet Office highlights this issue.

Loose coupling of data and real-world applications is required to increase data reusability.

The digital twin can be considered a means of increasing the reusability of data. A digital twin is a technology that replicates the real world in the cyber world by using data collected from the real world. Data reusability is increased by implementing a mechanism that collects data according to the requirements of the real-world application to be implemented, builds a digital twin from the collected data, and combines the data with real-world applications.

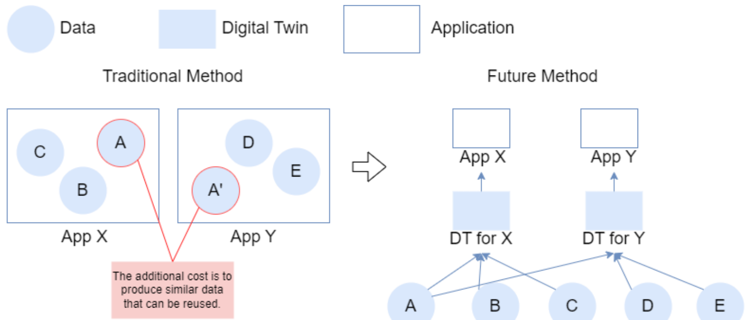

Figure 1 Creating digital twins from various data to implement application

Currently, data platforms that manage data to create digital twins are being developed. For example, some of the data platforms for creating digital twins are PLATEU [2] and Azure Digital Twins[3]. Data platforms for create digital twins will continue to grow.

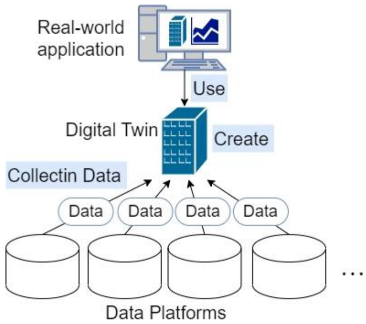

Figure 2 Implement real-world applications using a digital twin created from platform data

However, the proliferation of data platforms reduces their discoverability. As data platforms proliferate, data related to specific locations become scattered across many data platforms. An application implementer must search for many data platforms whose data are suitable for the digital twin to implement the application. Failure to discover the data increases the cost of application implementation because the data must be regenerated.

This study attempts to solve data discoverability issues to improve the reusability of data for creating digital twins. This study proposes the following method. A function to output metadata representing the spatial structure was implemented on spatial data platforms. Interfaces for providing data were implemented on related data platforms. The procedure for accessing interfaces to provide related data is described in the metadata. In our proposed method, it is possible to repeatedly search for available data on data platforms using metadata. Consequently, the reusability of the data is improved.

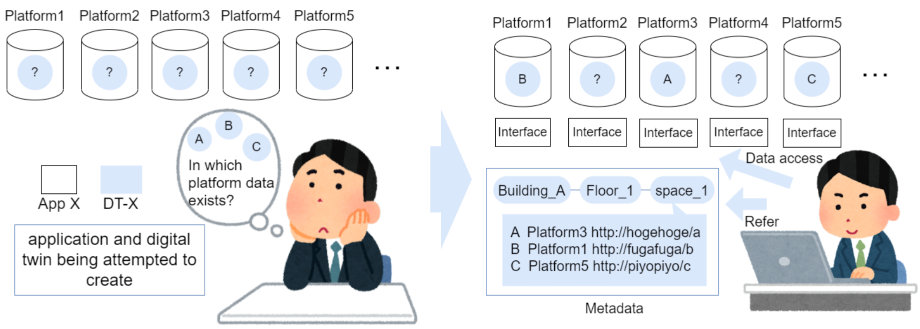

Figure 3 Cross-platform data discovery using metadata

The main feature of the proposed method is the construction of a data catalog that realizes spatial and semantic searches. A data catalog is a collection of metadata for a data search. In the proposed method, data users can search for data to create a digital twin by traversing the semantic connections of the spatial elements. For example, the data user can search for related data based on a query such as "data related to all office rooms on the second floor of this building."

In this study, prototype systems for Building Information Modeling (BIM) as spatial data were implemented using the proposed method. Based on the proposed method, a spatial data management server and related data management server were implemented and linked using metadata. A spatial data server was implemented that had a function to manage BIM data compliant with IFC, and a function to export metadata from stored BIM data. In the case of the related data server, an API server was implemented with a function to manage the 3D building model data. An endpoint for accessing the 3D data API server is described in the metadata.

2. Related Works

This section describes related work. The proposed method searches for related data to construct a digital twin by using spatial metadata from various platforms.

2.1. 3D City DB

This section presents related work linking the geometric and attribute data of spatial data. An example of a method for associating spatial data with related data is 3D City Database (DB). The 3D City DB is open-source software for server-side applications to manage CityGML [5] data. CityGML is a data format used in a GIS and is based on XML. Because CityGML contains not only geometric data but also complex semantic data, such as object type, CityGML is difficult to store in a spatial database. This section describes a method for obtaining information on the API of a related data platform to provide related data from the searched metadata. Data users receive attribute data values describing how to access the related data from the obtained metadata. Data users then access related data based on the data access method obtained from the metadata attributes. The purpose of the 3D City DB is to provide an environment that allows both attribute values-based and spatial search functions to be used with CityGML. Therefore, the 3D City DB approach does not solve the data discoverability problem.

Users who want to use CityGML data need to know that there is a spatial database that manages the geometric data of a CityGML file and a CSV file that manages the attribute data of the CityGML file. In addition, the data user must know a method to access the database and CSV file.

The purpose and approach of our proposed method differ from those of a 3D City DB. The purpose of the proposed method is to create an environment to search and collect data across various spatial data platforms and construct a digital twin. Therefore, in our proposed method, the 3D City DB is also one of the search targets of the data management system for building a digital twin.

2.2. BIMServer

This section describes BIMServer [6] as a related work. BIM is an integrated architectural process that uses 3D building information. Three-dimensional building model data used in the BIM process are referred to as BIM data. BIM data have a format called Industry Foundation Classes(IFC), which is an international standard data format for data exchange. The IFC was developed by buildingSMART International (bSI), an international standardization body for BIM, and is ISO certified. Among the systems that manage BIM data, a system that manages IFC-compliant BIM data is called an "IFC model server," and research has been conducted to realize it. The problem in implementing the IFC model server is storing the IFC data in the database. Because IFC data have a complex object-tree structure, storing IFC data in a database is difficult.

BIMServer is an IFC model server implemented using BIMServer.org to solve this problem. It is open-source software for server-side applications for managing IFC data. BIMServer is a monolithic application that not only stores, manages, and reads IFC data but also provides various functions necessary for the building lifecycle, such as clash detection.

The purpose and approach of our proposed method are different from those of the BIMServer. The purpose of the proposed method is to create an environment to search and collect data across various spatial data platforms and construct a digital twin. Therefore, in the proposed method, the data managed by the BIMServer are also included in the search targets.

2.3. Ontologies for spatial metadata

This section describes the ontologies of spatial metadata. Ontology is a means of formally describing knowledge and the relationships between knowledge. An ontology is used to describe the schemas of metadata on the Web. Ontologies have been proposed to describe the schemas of spatial elements in BIM data.

The Building Topology Ontology (BOT) [7] is an example of an ontology that describes the concept of spatial elements. BOT is an ontology that represents a simplified spatial structure formulated by bSI and The World Wide Web Consortium(W3C) to simplify and express IFC-compliant BIM data. BOT represents spatial elements, such as instances of site, building, storey, space, and element classes. Spatial elements are connected by properties such as "containsZone" in BOT.

Next Generation Service Interfaces - Linked Data (NGSI-LD) [8] is also an example of an ontology for describing the concept of spatial elements. NGSI-LD is an ontology based on the Next Generation Service Interfaces (NGSIs) [9], which is the data specification formulated for use in digital twins, smart cities, etc. NGSI-LD was standardized by the European Telecommunications Standardization Institute (ETSI). NGSI-LD represents spatial elements as instances of the building class, which include instances of sensor class and instances of human class.

The Digital Twin Definition Language (DTDL) [10] is primarily used to express the data of digital twins managed by Azure Digital Twins. In addition, several domain-specific ontologies based on DTDL have been proposed. An example is Real Estate Core[11] Real Estate Core is the ontology defined to represent smart buildings. Real Estate Core represents spatial elements, such as instances of land, building, level, and room classes. These classes are linked by "isPartOf" property. DTDL is an ontology for representing digital twins created with reference to ontologies such as NGSI-LD

These ontologies are defined to structurally express the relationships between spaces and the IoT devices installed in those spaces. However, these ontologies are not intended for use in cross-platform data discovery. The proposed method uses metadata conforming to these ontologies to search for data and create a digital twin. Our proposed method allows searching for related data across platforms by outputting metadata that conform to such ontologies from spatial data platforms and writing the endpoint information of the related data into metadata. The proposed method effectively uses the metadata described by these ontologies.

2.4. Heterogeneous Spatial data Integration

This section presents a review of the research conducted on the integration of heterogeneous spatial information. Spatial data, which are intrinsic to various domains, require distinct management and utilization systems based on their intended application. For instance, while GIS is employed for processing maps, BIM is utilized for managing building data. These systems embody distinct data models, coordinate systems, and coordinate accuracies, highlighting the level of heterogeneity which necessitates a data conversion process when integrating BIM data within a GIS environment, or vice versa.

Moreover, the data model employed may vary depending on the specific use case, even when utilizing the same tool for data processing. For example, different data models within GIS may be adopted based on the application, and resolving the inherent heterogeneity is crucial for achieving data integration.

Numerous studies have been conducted to address the inherent heterogeneity in spatial data. Within the GIS domain, significant strides have been made concerning the integration of disparate spatial databases harboring different data models from distinct domains. For instance, Ponjavic et al. [12] integrated multiple GIS databases encompassing public administration data (e.g., cadastral and facility registers) by crafting an integrated data model and converting spatial data from various databases accordingly. Similarly, Imen et al. [13] attempted to integrate heterogeneous GIS databases pertinent to disaster management. Given that organizations such as the police, fire department, and EMS maintain distinct GIS databases with varying data models, searching for data across organizations during a disaster has proven to be challenging. To address this issue, Imen et al. transformed disparate data into a knowledge graph in RDF format, striving for a unified integration by establishing interlinks.

The integration of disparate spatial data managed by different systems is a focal point of further research. Specifically, the melding of GIS and BIM data has garnered significant attention, as delineated in reviews by Junxiang [14] and Beck [15]. Noteworthy attempts include the conversion of BIM data to GIS data for integration within GIS [16,17,18] and the semantic translation of BIM and GIS data into an RDF-formatted knowledge graph for amalgamation [19,20,21].

The strategies employed in these studies predominantly focus on data transformation and the construction of integrative data models. A common drawback of these approaches is the potential loss of original data information during transformation. Moreover, knowledge graph-based integrations often culminate in data redundancy and increase in data volumes, incurring data management costs. In contrast to previous studies, our research does not critique data integration methodologies. Instead, we identify and integrate the location of the target data on the Internet using cross-platform search capabilities enabled by spatial metadata. In the method proposed in this study, individual datasets are managed while preserving their originality, retrieved at the time of use, and transformed as necessary, thereby mitigating the loss of information associated with data transformation and the escalation in data redundancy caused by knowledge graph transformation.

2.5 Spatial Data Catalogs

In this section, the discussion pivots towards an examination of prior research concerning data catalogs, particularly focusing on the retrieval of spatial datasets. Noteworthy contributions in this area have been made by Helbert [22] and Vasilis [23], who devised methods to describe and maintain metadata by employing linked data. Their work facilitates the retrieval of spatial datasets based on geographic coordinates, such as latitude and longitude, enabling the construction of effective data catalogs. The primary objective of these studies was to create an environment conducive to spatial dataset retrieval using geospatial search mechanisms.

Unlike these endeavors, this study pursues a slightly divergent goal, considering search queries of a more nuanced nature, such as "I wish to obtain a floor map of a specific floor in a particular building," catering to use cases such as indoor navigation. The goal is to create a retrieval environment that transcends coordinate-based retrieval. Instead, our research aims for semantic retrieval of data by utilizing metadata that express semantic relationships among spatial components.

3. Proposed method

In this section, the proposed method is described. The proposed method comprised four elements. The first element is the metadata output of the spatial information platform. The second element is the method of associating metadata with information regarding the endpoint to access the related data. The third element is a method for spatially and semantically searching the metadata of the spatial element whose related data are retrieved from the metadata group by traversing the metadata graph. The fourth element is a method for obtaining information to access related data from the searched metadata.

Figure 4 presents an overview of the proposed method. The elements that comprise this method are spatial data platforms and data platforms that manage related data in spaces. Examples of related data include 3D model data for visualization, and 2D floor plan data. Spatial data platforms have APIs that expose metadata to represent spatial structures. Data platforms that manage related data have APIs to provide the data. Metadata contains information about endpoints for accessing data about spaces represented by metadata.

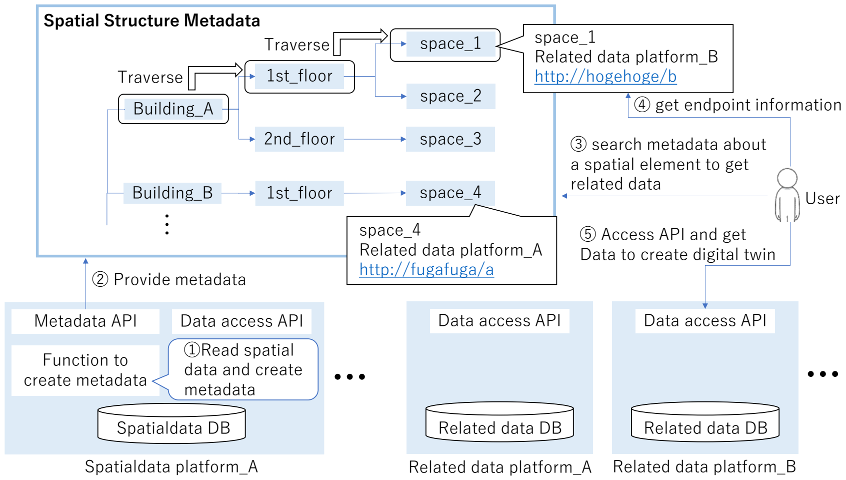

Figure 4 Overview of proposed method

3.1. Spatial structure metadata

This section describes the metadata representing the spatial structure. The spatial structure metadata were tree structured. Each node in the tree represents a spatial component. The edges that comprise the tree represent the connection and inclusion relationships between spatial components. Spatial structure metadata represent the relationships between spatial elements as edges.

The reason for representing metadata as a graph data model is to ensure compatibility with existing spatial metadata data models, such as BOT, DTDL, and NGSI-LD, which represent the spatial structure as graph data.

Metadata is created by the spatial platform from stored spatial data and is provided by the metadata provisioning API.

3.2. Linking method from metadata to API providing related data

This section describes the linking method between the metadata and the API. In this method, metadata contain attributes that contain information about the API endpoint of data platforms that manage data about spaces represented by metadata. For example, a floor metadata attribute for association contains information about the API endpoint of the data platforms that manage floor data, such as 3D model data and floor maps.

3.3. Method for semantic and spatial search

This section describes a method for spatially and semantically searching the metadata of a spatial element whose related data must be retrieved from the metadata by traversing the metadata graph. Data users search for spatial element metadata from which they want to obtain related data from the group of metadata outputs by the spatial data platform by traversing the edges that represent semantic connections between nodes representing a spatial element.

For example, a user may wish to retrieve data related to a room on the first floor of a building. First, the user retrieves floor instances of the building from the building instance by traversing the semantic connections between the building and the floor. Next, metadata representing the first floor were retrieved from the retrieved floor metadata group. Subsequently, by tracing the semantic relationship between the floors and rooms, metadata representing the room associated with the first floor were obtained.

3.4. Method to access API for obtaining related data

This section describes a method for obtaining information about the API of a related data platform to provide related data from the searched metadata. Data users receive attribute data values describing how to access the related data from the obtained metadata. Data users then access related data based on the data access method obtained from the metadata attributes.

4. Implementation

This section describes the implementation of the prototype system based on the proposed method. Figure 5 shows an overview of the prototype system. This implementation primarily deals with the BIM data as spatial data. The prototype consists of two systems. The first is the IFC model server. The second is a system that manages 3D data generated from IFC data, provides functions for exporting 3D data in various CG data formats, and spatially searches for 3D objects. The second system will be referred to as the "IFC geometry server." The Resource Description Framework(RDF) [24], which is the metadata description framework of the Semantic Web, was adopted to write metadata to link these systems. Metadata were provided by the metadata API of the IFC model server.

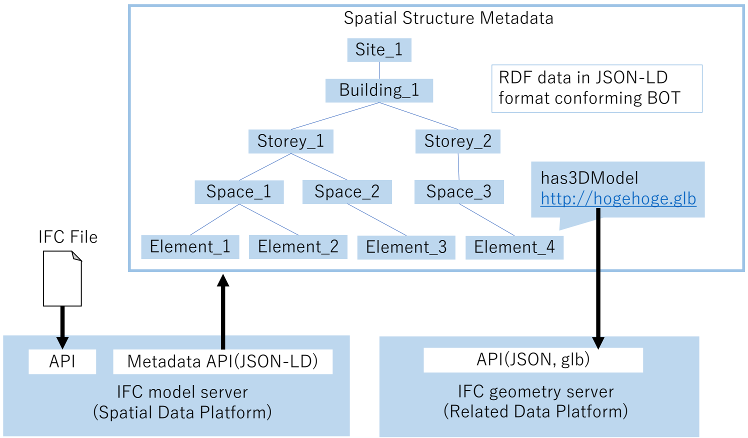

Figure 5 Overview of prototype systems

4.1. IFC model server implementation

This section describes the implementation of our proposed IFC model server. Figure 6 shows the components of the prototype system. The IFC model server was implemented using Python 3.9 and ArangoDB[25]. The function of the IFC model server as a Web API server was implemented using Flask [26]. IFC model server uses IfcOpenshell-python [27] to interpret IFC data. The IFC model server uses RDFlib [28] to export metadata to RDF. In our implementation, the IFC model server stored the IFC data in a graph database. This prototype used ArangoDB as a graph database to store IFC data. To access ArangoDB from Python, our prototype system used Python-Arango [29] as the database driver.

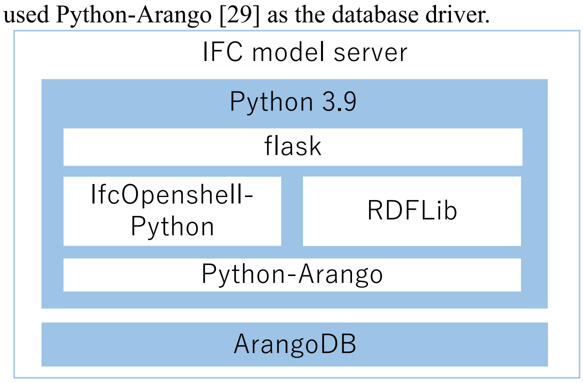

Figure 6 Components of IFC model server

4. 1. 1 Process to upload IFC file

This section describes the process of uploading IFC files. The process of uploading an IFC file to the IFC model server is shown in Figure 7. The IFC model server uses the following procedure to upload IFC files.

1. The IFC model server converted the uploaded IFC file into a Python object tree using IfcOpenshell-Python.
2. IFC model server stores this tree data in ArangoDB by using Python-Arango.

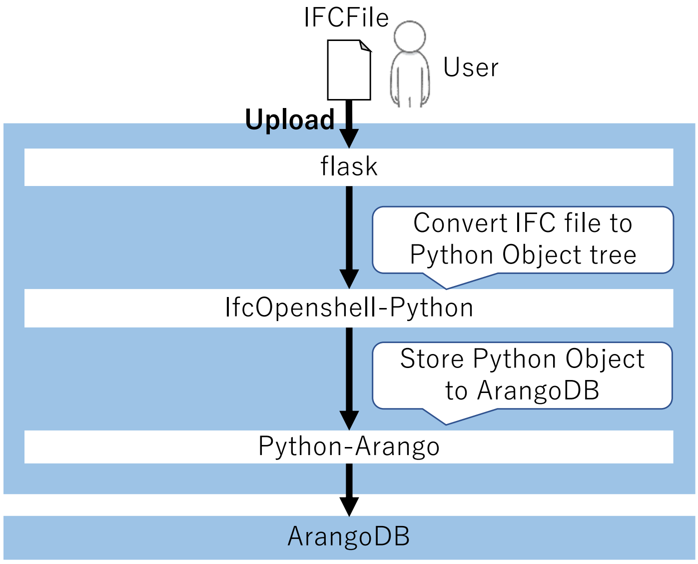

Figure 7 Upload IFC file process to IFC model server

4. 1. 2 Process to provide data

This section describes the data provision process using the API of the IFC model server. The process of providing data using the API of the IFC model server is shown in Figure 8. The IFC model server provides data using the API through the following steps:

1. receive request through API implemented by flask.
2. query data in ArangoDB using Python-Arango and read IFC data.
3. convert data from IFC data to Python object tree.
4. convert python object tree to JSON and provide JSON data through API.

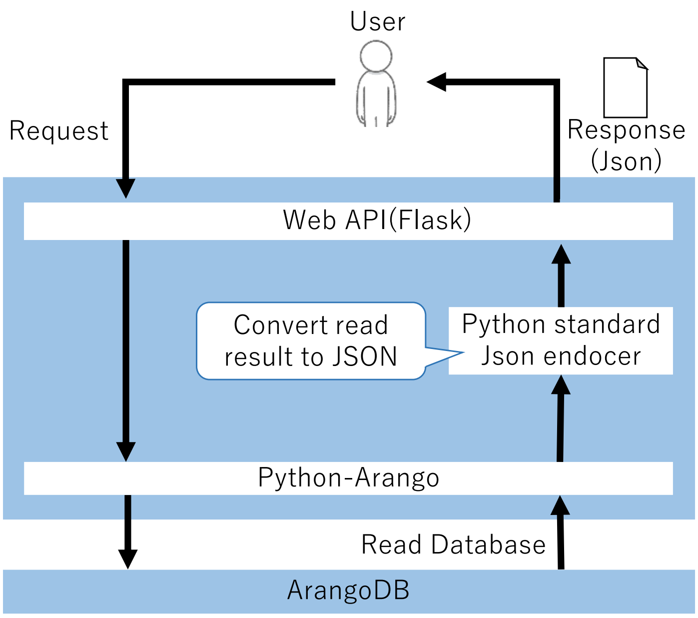

Figure 8 Process to provide data from IFC model server API

4. 1. 3 Process to provide metadata

This section describes the process of providing metadata by using the API of the IFC model server. The process of providing metadata using the API of the IFC model server is shown in Figure 9. The IFC model server provides metadata using the API through the following steps:

1. receive request through metadata API implemented by Flask.
2. query data in ArangoDB using Python-Arango and read IFC data.
3. Convert data from IFC data to Python object tree.
4. convert a Python object tree to JSON and provide JavaScript Object Notation for Linked

Data (JSON-LD) [30] data through an API.

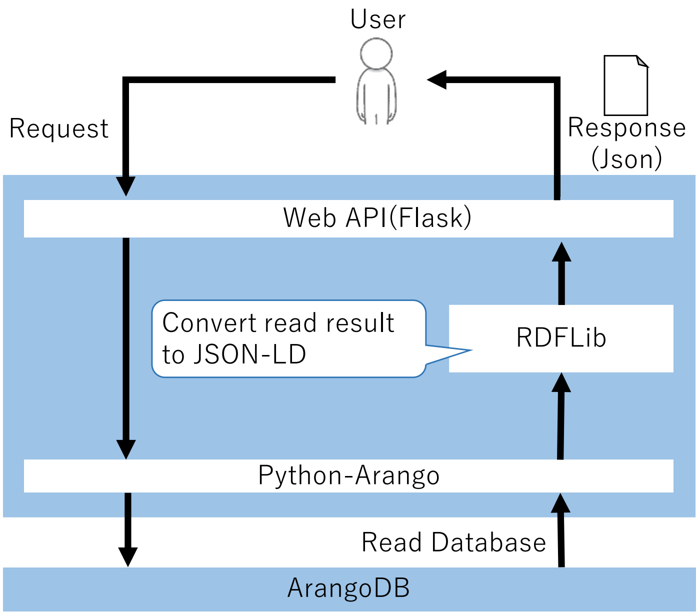

Figure 9 Provide metadata process through IFC model server API

4.2. IFC geometry server implementation

This section describes the implementation of an IFC geometry server. Figure 10 shows the components of the prototype system. The IFC geometry server was implemented using Python 3.9 and PostGIS. The function of the IFC geometry server as a Web API server was implemented using Flask. IFC geometry server uses IfcOpenshell-Python to interpret IFC data. The IFC geometry server used trimesh [31] to export various 3D CG formats. This prototype uses PostGIS [32] as a spatial database to store 3D mesh data. To access PostGIS in Python, our prototype system used psycopg2 [33] as a database driver.

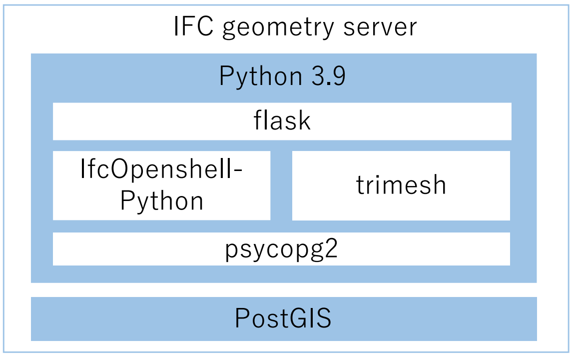

Figure 10 Components of IFC model server

4. 2. 1 Process to Upload IFC file

This section describes the process of uploading an IFC file to the IFC geometry server. The process of uploading an IFC file to the IFC geometry server is shown in Figure 11. IFC model server uses the following procedure to upload IFC files. This IFC geometry server uses the following procedure to upload IFC files.

1. The IFC geometry server converts the uploads IFC file into Python objects that represent the geometry using IfcOpenshell-Python.
2. IFC geometry server stores these Python object data in PostGIS using psycopg2.

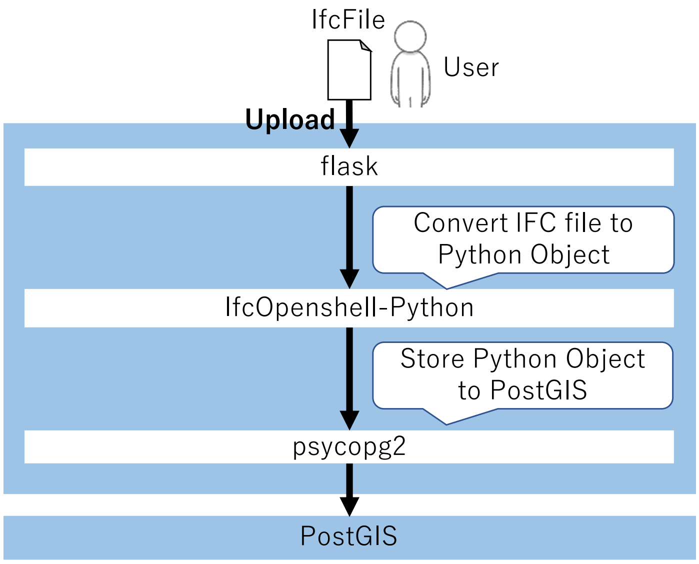

Figure 11 Upload IFC file process to IFC geometry server

4. 2. 2 Process to provide geometry data

This section describes the process of providing geometric data using the API of the IFC geometry server. The process of providing geometric data as a JSON file using the API of the IFC geometry server is shown in Figure 12. The process of providing the geometric data as a GLB file using the API of the IFC geometry server is shown in Figure 13. The IFC geometry server provides data using the API through the following steps:

1. The IFC geometry server receives requests through an API implemented in a flask, and reads the geometric data from PostGIS by querying using psycopg2.
2. IFC geometry server converts data from database query results to Python object.
3. IFC geometry server converts Python objects to JSON and expose them through API.

As a response format, the IFC Geometry Server API supports the GLB format, which is a binary version of the GL Transmission Format (GLTF) [34] used to create content that runs on web browsers. If the specified response data format is in GLB format, the trimesh will convert the Python objects.

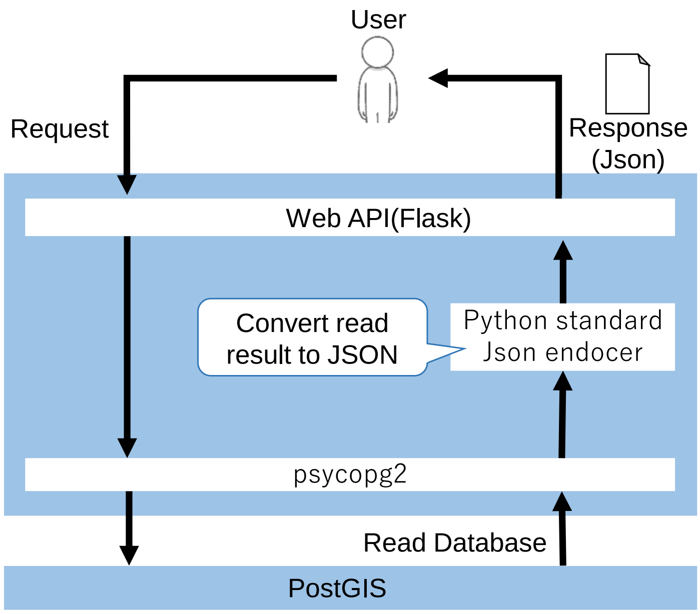

Figure 12 Provide geometry data process through IFC geometry server API

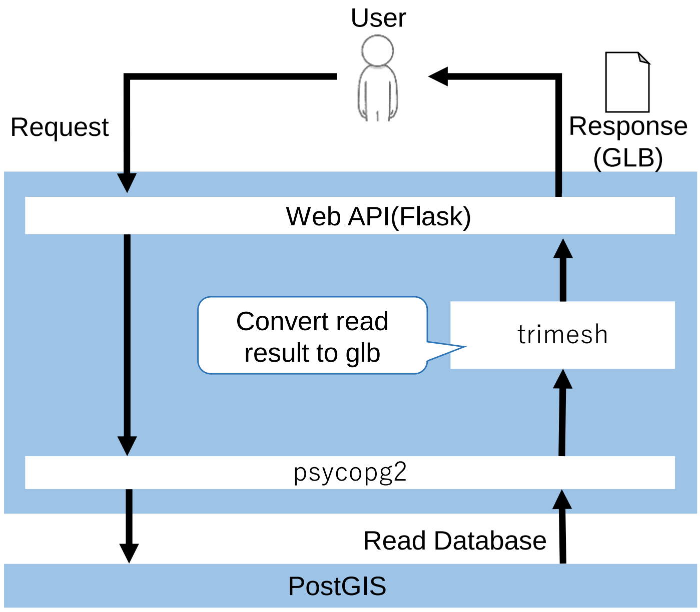

Figure 13 Provide GLB data process through IFC geometry server API

4.3. Metadata data model

In this section, we describe the proposed metadata model. In this implementation, the BOT was used as the metadata data model to represent the spatial structure. BOT reason for using BOT to represent spatial metadata because it has a high affinity for spatial data managed by the IFC model server. The metadata provided by the IFC model server API include site class instances, building class instances, story class instances, space class instances, and element class instances. These classes have an attribute definition called "has3DModel" to represent the associated 3D model. In this implementation, an IFC geometry server endpoint for accessing associated 3D model data is described as an attribute value of "has3DModel." Metadata is created each time that metadata API is called.

4.4. Process to upload IFC file

This section describes the process of uploading data to these two servers. The process of uploading the IFC file to the two servers is shown in Figure 14. The files were uploaded to the

two servers using the following four steps:
1. An IFC data owner uploads an IFC file to the IFC model server.
2. The IFC model server stores the uploaded IFC data and generate a model ID.
3. The IFC model server returns the model ID to the data owner.
4. The owner uploads the same IFC data to the IFC geometry server using the model ID issued by the IFC model server.
5. The IFC geometry server stores the geometric data generated from the uploaded IFC data associated with the model ID.

Once processes 1 to 5 are completed, the metadata can be output from the IFC model server. The metadata contain information regarding the endpoint for accessing the geometric data. Information about the endpoint is registered as a value of the "has3DModel” attribute.

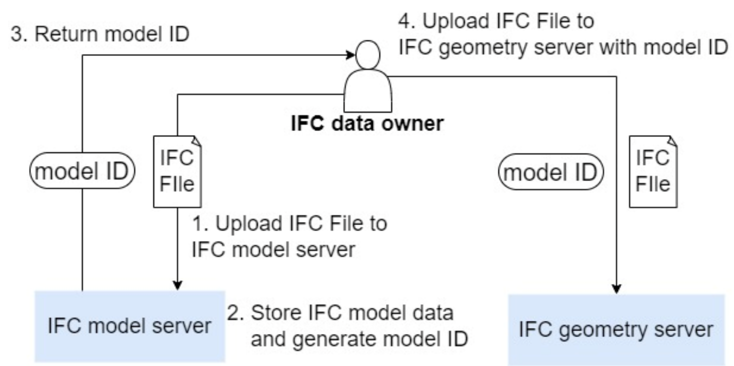

Figure 14 IFC file upload process to two systems

5. Verification of Prototype System Operation

This section describes the verification of the proposed method by using a prototype system. The purpose of this check was to verify that the endpoint of the IFC geometry server could be accessed using the metadata output from the IFC model server. The functionality of the prototype system was verified using the following procedure:
1. Upload a test IFC file to the IFC model server.
2. Upload a test IFC file to the IFC geometry server.
3. Retrieve metadata from the IFC model server.
4. Query the retrieved metadata.
5. Retrieve geometry data from the endpoint registered in the has3DModel attribute.

A test model is shown in Figure 15.

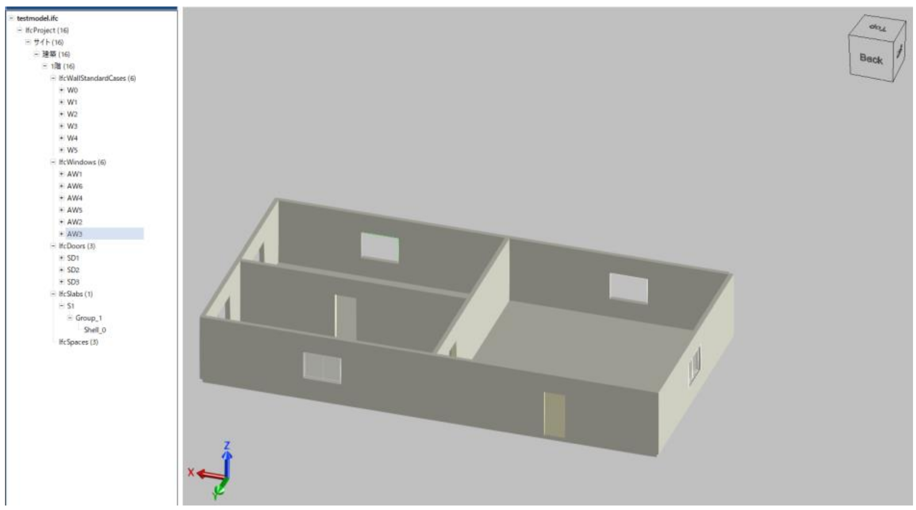

Figure 15 Test IFC model

5.1. Uploading IFC file to IFC model server

This section describes the uploading of IFC data to the IFC model server and the resulting upload, as expected. Figure 16 shows the test web UI for uploading an IFC file using the API of the IFC model server. The Web UI was developed by Swagger [35]. To confirm this, the test IFC file was uploaded using the web UI.

Figure 16 shows the file upload results for the Web UI. The IFC model server results show that the test IFC file was successfully uploaded. In Figure 16, the IFC model server returns the model ID ("Ifcmodel_id") and message ("upload finished").

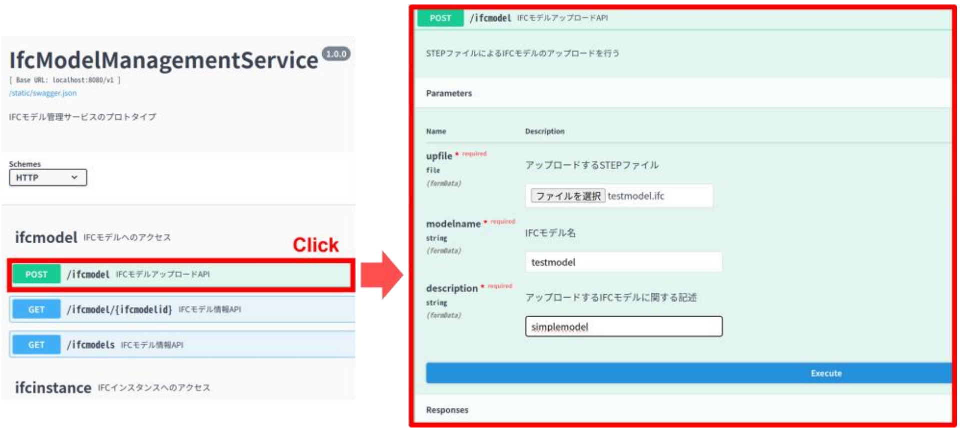

Figure 16 Swagger Web UI of IFC model server

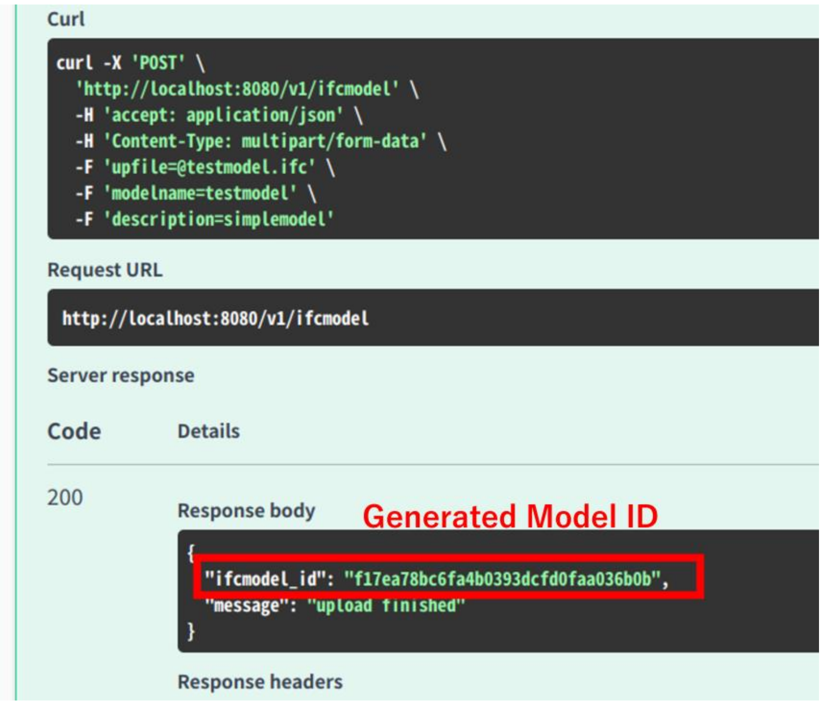

Figure 17 Result of IFC file upload to IFC model server

5.2. IFC file upload to IFC geometry server

This section describes the uploading of the IFC data to the IFC geometry server and the resulting upload. Figure 18 shows the test web UI for uploading an IFC file using the API of the IFC geometry server. For verification, the test IFC file was uploaded using a web UI.

Figure 18 shows the file upload result on the web UI. The IFC geometry server result shows that the test IFC file was successfully uploaded. Figure 18 also shows that the model ID was passed as a parameter when the test IFC file was uploaded to the IFC geometry server. Figure 19 shows the result for uploading IFC file to IFC geometry server.

Figure 18 Swagger Web UI of IFC geometry server

Figure 19 Result of IFC file upload to IFC geometry server

5.3. Retrieving Metadata

This section describes the results of metadata retrieval from the IFC model server. Access to the IFC model server API was performed to retrieve the metadata. Figure 20 shows a portion of the metadata retrieved by the IFC model server API. In this case, the endpoint was “http://localhost:8080/v1/bot/f17ea78bc6fa4b0393dcfd0faa036b0b”. The metadata from the

IFC model server are described in JSON-LD format. Metadata instances contain an attribute representing the object type. The values of the object-type attribute indicate that the instances of metadata are instances of BOT spatial element classes, such as Elements, Space. Metadata objects have other attributes such as type, Class, GlobalId, Name. Metadata instances also have a "has3Dmodel" attribute. The value of "has3Dmodel" is information about the endpoint of the IFC geometry server, but these attributes are tentatively defined and do not conform to BOT and any standard or published specification. This result shows that the IFC model server was able to export the metadata to obtain endpoint information for accessing related data, and that the metadata could be retrieved via the IFC model server API.

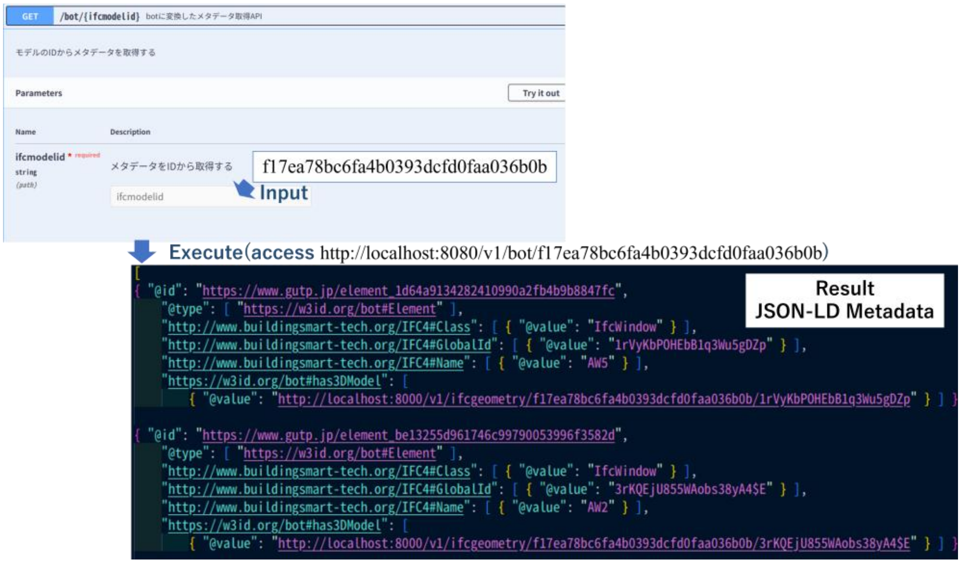

Figure 20 Example of metadata exported from IFC model server API

5.4. Querying metadata

This section describes the results of querying the metadata from the IFC geometry server based on the endpoint information described in the metadata. BOT metadata allow a spatial element to be searched using data exploration means, such as SPARQL Protocol and RDF Query Language (SPARQL) [36]. SPARQL is the standard query language and protocol for RDF data and RDF database. Figure 14 shows the Python code-to-query metadata provided by the IFC model server's API using SPARQL on the Jupyter notebook and the result of the query. This SPARQL query obtains story class instances and the space class instances linked to these instances. The lower part of Figure 19 shows that the executed query was successful. This result shows the metadata exported from the IFC model server, which enables the spatial and semantic searching of related data in the IFC geometry server.

Python code for running SPARQL

```
bot_url = "http://localhost:8080/v1/bot/f17ea78bc6fa4b0393dcfd0faa036b0b.jsonld"

g = Graph()
g.parse(bot_url, format="json-ld")

query = """
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX bot: <https://w3id.org/bot#>

SELECT * WHERE {
    ?storey rdf:type bot:Storey;
        bot:hasSpace ?space.
}"""

qres = g.query(query)
for row in qres:
    print(f"StoreyURI:{row.storey} SpaceURI:{row.space}")
```

Result of SPARQL

```
StoreyURI:https://www.gutp.jp/building_storey_084fbd290dc84f5eb830e17b2a7e5c5c SpaceURI:https://www.gutp.jp/space_385a84acde8a4a848e5a37c0
e9b4ab03
StoreyURI:https://www.gutp.jp/building_storey_084fbd290dc84f5eb830e17b2a7e5c5c SpaceURI:https://www.gutp.jp/space_6a9b0d9ae88f47c69419587b
f38f38bf
StoreyURI:https://www.gutp.jp/building_storey_084fbd290dc84f5eb830e17b2a7e5c5c SpaceURI:https://www.gutp.jp/space_348b36aada094d2083ee2b28
72cd1d5d
```

Figure 21 Example of querying metadata using SPARQL

5.5. Retrieving geometry data

This section describes the results of the geometric data retrieval from the IFC geometry server based on the endpoint information described in the metadata. Figure 22 shows the Python code used to access the IFC geometry server endpoint described in the metadata attribute and visualize the 3D data retrieved from the endpoint in a Jupyter notebook. And figure 23 shows the response of the IFC geometry server endpoint described in the metadata. A response from the IFC geometry server is a serialized JSON with attributes such as globally_unique_id, class_name, vertices, indices, normals, and face_colors. In the Python code shown in Figure 22, the JSON data are visualized using trimesh. The resultant visualization is shown at the bottom right, which shows that the related data can be obtained by accessing the endpoint described in the metadata provided by the IFC model server.

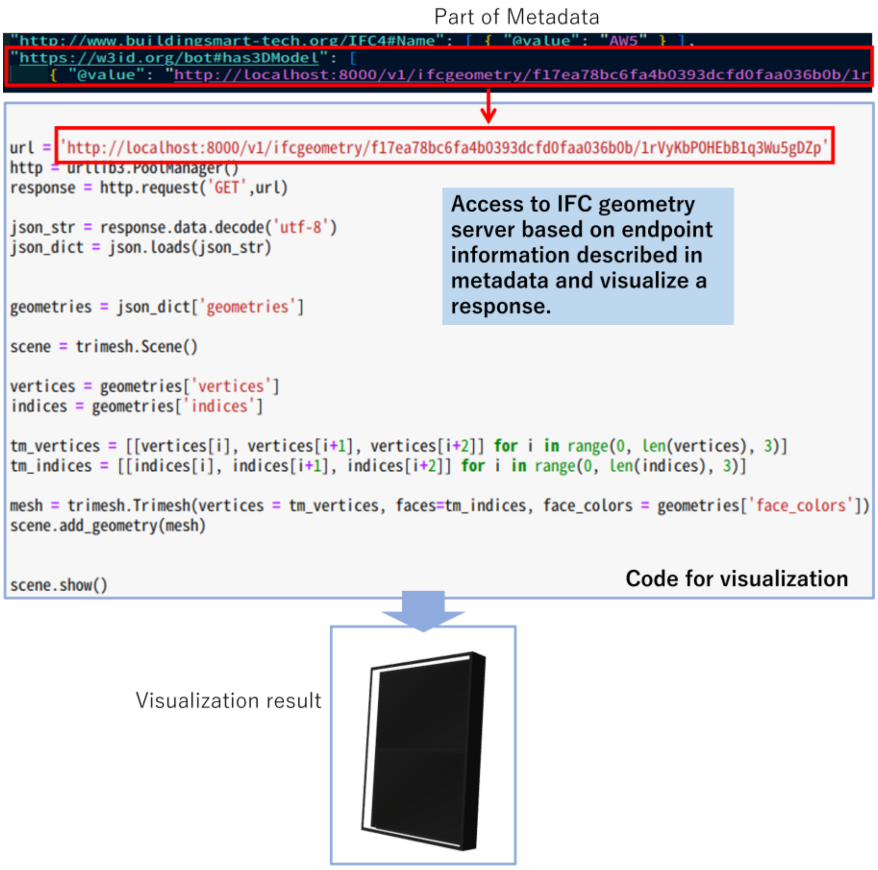

Figure 22 IFC geometry server access example

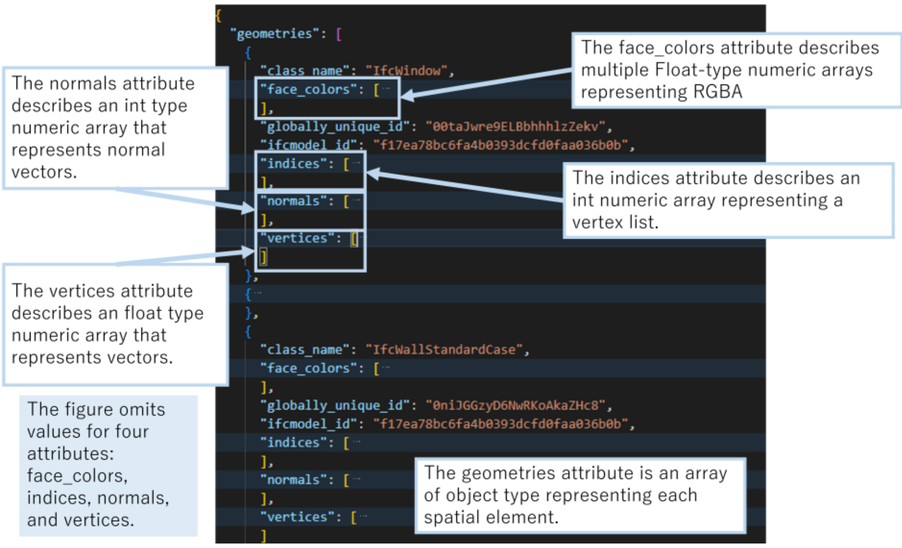

Figure 23 Example of response from IFC geometry server API

5.6. Discussion

This section provides the results of the operational verification of the prototype and discusses of the results.

5. 6. 1 Verification Results

The operational verification of the prototype system undertaken in this study yielded several significant insights that are encapsulated in the following four points:

1. The generation of metadata is successfully executed through the IFC model server API.
2. The metadata notably encompass the endpoint information of the IFC geometry server API.
3. Using SPARQL, it is feasible to extract endpoint information from the output metadata, thereby elucidating the relationship between spatial elements.
4. The endpoint information gleaned from the metadata search facilitates the acquisition of geometry data from the IFC geometry server API.

These findings substantiate the efficacy of the methodology proposed in this study, illustrating its potential in fostering an environment in which metadata sourced from spatial information platforms serve as a pivotal tool in retrieving data related to specific spatial elements.

Furthermore, the outcomes of this verification accentuate the enhancement in data reusability brought about by the proposed method, particularly in enhancing the retrieval of data dispersed across various platforms.

However, the operational verification also highlighted additional challenges that necessitate resolution to further bolster data reusability. These challenges, along with a more detailed discussion, are explored in Section 5.6.2.

5. 6. 2 Issues to solve

The operational verification of the prototype system showed that the proposed method can realize an environment in which data can be explored for the construction of a digital twin through a semantic search against metadata. However, there are issues that need to be resolved before our method can be implemented in practice. The issues to be resolved are described below.

5.6.2.1 Generating spatial data and uploading to the platform

The generation of spatial data from which spatial structure metadata can be generated and uploaded to the platform is an important issue. In the implementation of the prototype system, the IFC data must be populated with instances of the IfcSpace class, to output the metadata from the uploaded IFC data to the IFC model server. However, the IFC specification only allows data that have not been populated with instances of the IfcSpace class. To realize a data exploration environment based on the proposed method, it is necessary to have a mechanism by which spatial data that can be used for metadata generation are created and uploaded to a platform such as the IFC model server.

It is also important to establish a mechanism by which spatial data other than IFC data can be generated for metadata extraction and uploading to the platform. For example, if the method proposed in this study were to be applied to CityGML data, platforms would be required to generate metadata extractable CityGML. A mechanism is required for the generated CityGML data to be uploaded to CityGML data platforms for extraction and generate metadata.

5.6.2.2 Associating metadata with APIs

Associating the API with metadata is important. The prototype system in this study assumes that the same IFC data are uploaded to the IFC model server and the IFC geometry server such that the prototype is designed and implemented to automatically include endpoint information from the IFC geometry server in the metadata output from the IFC model server. However, the data associated with the metadata may have been generated independently of the spatial data from which the metadata were generated, and the platform managing the data may not have implemented the ability to automatically link the API providing the data to the metadata. For example, the proposed method can search for data such as floor map images and 2D drawing data as long as the data are related to spatial elements. It is likely that platforms that handle such data do not have the ability to automatically associate their own APIs with the metadata output from platforms that handle spatial data. A mechanism is required to ensure that when related data are uploaded to a platform for handling, the endpoint information for accessing that data is described in the metadata of the target spatial element.

5.6.2.3 API reusability and interoperability

The reusability and interoperability of APIs are also important issues for different data users so that the different data can be integrated to construct a digital twin in an environment constructed based on the proposed methodology. To construct a digital twin using data related to the space managed by the platform, it is necessary to understand the specifications of the API platform and process the data obtained from the API. In this operation, the prototype system used for verification has only one type of server to connect to and only one data user. However, when a large number of users acquire related data from many platforms, the burden on individual data users increases if the API specifications differ from platform to platform or if individual APIs have specifications that are difficult for users to handle. To realize the construction of a digital twin and the activation of application implementation using a digital twin with the proposed method, it is necessary to reduce the learning cost of APIs and the burden of data acquisition and processing. Therefore, it is desirable to have common specifications for data-provisioning APIs for each domain, or simple specifications for each API and easy data processing.

5.6.2.4 Providing means to understand data quality

It is desirable for data users to know in advance whether the data obtained from the API meet the requirements of the digital twin they want to build. For example, consider the case of building a digital twin for automated robot control using data obtained from an IFC geometry server. The data user would want to know in advance whether the data that can be retrieved from the IFC geometry server API contain information such as furniture, unit system, and coordinate system of the data. However, the information described in the metadata exported from the prototype system is insufficient for the user to determine the quality of the data in advance. To solve this situation, it is necessary to clearly indicate the quality of the data provided by the API to users of the metadata. This would allow users to understand in advance whether the data are suitable for the intended purpose and how data should be processed.

5.6.2.5 Metadata interoperability

Metadata interoperability between multiple related data platforms is also an issue. In this prototype system, there is only one spatial data platform, and the metadata data model is fixed to BOT. However, if multiple spatial data platforms exist in parallel and their metadata models are different, data users will need to learn the metadata specifications of each platform in advance when issuing data-exploration queries. This increases the burden of data exploration on users and inhibits the creation of a digital twin. To create an environment in which data users can easily perform data exploration, it is necessary to improve the interoperability of metadata between platforms.

6. Future Challenges

This section describes the further work required to address this issue. In this study, the feasibility of the proposed method is demonstrated through the implementation of a prototype system. The proposed method is effective for creating digital twins. To realize an environment that enables the creation of digital twins by the semantic and spatial searching of data in the future, our proposed method has several issues that need to be solved. Four possible works for addressing these issues are described below.

6.1. Standardization of platform API and schema of metadata

This section describes the work performed for data standardization. To realize data discovery and linkage across platforms using our proposed method, interoperability of data and metadata is required. The lack of metadata interoperability across spatial data platforms makes the semantic and spatial discovery of data difficult because data users must understand the metadata data model and rewrite queries to search for data for each platform. In addition, the different API data models for each platform make the creation of digital twins costly because data users need to convert the data provided by the platforms. To solve these issues, standardization of the metadata data model and platform API is required.

6.2. Standardization of data creation process

This section describes the work conducted to standardize the process of creating and uploading data to the platform in the future.

The following three steps are necessary to implement our proposed method.

1. A process must be established to generate spatial data from which metadata can be extracted and uploaded to the platform.
2. A process must be established to generate data related to the spatial data to be uploaded to the platform.
3. A process must then be established to generate metadata to link the spatial data uploaded to the platform with the related data.

To realize the proposed method, the data generators need to perform steps 1 to 3. To do this, it is necessary to formulate and standardize the process for each of steps 1-3 so that data generators can generate data according to the standard processes.

6.3. Establishment of data quality description and search method

This section describes the work performed using on data quality description method. To implement an application using a digital twin, the data that comprise the digital twin must satisfy its requirements. There are many requirements about data quality that application implementers want to know in advance; for example, data format, data accuracy, data granularity, level of detail (LOD), etc. However, application implementers have no way to know the quality of data using our proposed method.

To promote our proposed method, it is necessary to establish a method to describe the quality of data and to establish a method to search for data not only spatially and semantically, but also with data quality requirements. In the future, the establishment of a data quality description and search method with data quality requirements will be required.

6.4. Approach to establish process and Standardization

In this section, we discuss the efforts required to solve the issues described in 6.1 to 6.3. Establishing data-creation processes and standardizing data models are key to solving these problems. For stakeholders to adopt the standards of the data model, data creation process, and data quality description, these standards should be practical and useful. To establishing practical and useful standards using knowledge extracted from the experiences of PoC is important.

To develop useful standard data models, processes, and data-quality description methodologies, we proposed the following four steps.

1. Gather stakeholders in a specific domain.
2. Conduct a PoC to implement the application.
3. Develop data models, processes, and data quality descriptions using knowledge gained from the PoC.
4. Promote international standardization by disseminating successful cases and increasing the number of applications.

In future, PoCs for application implementation using digital twins should be more active. Our prototype system is open source [37]; therefore, it can be used as a reference for implementation examples when the PoC is activated.

7. Conclusion

In the research, a method was proposed to link spatial data platforms and related data platforms using spatial metadata to create digital twins. Spatial data platforms are designed to output metadata representing spatial structures. Related data platforms have been designed with interfaces to provide data. The procedure for accessing interfaces to provide related data is described in the metadata. In this study, prototype systems were implemented based on the proposed method. The metadata and related data provided by the prototype system are verified. By querying the metadata output from the prototype system and retrieving data using the metadata information, we showed that our proposed method can spatially and semantically search for the data necessary to construct a digital twin.

However, the proposed method has the following limitations. The proposed method cannot ensure interoperability of API specifications and metadata between platforms. It is necessary to establish standardization of API specifications and metadata. It is also necessary to establish a standardized process for generating spatial data from which metadata can be exported, and for uploading them to digital twin platforms. To resolve these problems, formulating a process to generate spatial data which can be converted to metadata, developing specifications of metadata and API, and standardizing the process and these specifications are important. To develop useful standard specifications and processes, gathering stakeholders in a specific domain, conducting a PoC to implement the application, and using the knowledge obtained from the PoC are required. We opened our prototype system because we believe that various people can handle spatial information using the proposed method and develop it into a standard method through various PoCs.

In addition, the proposed method can be extended to IoT device data. Extending the proposed method to the IoT domain will be a future task.

References

[1] Cabinet Office. Smart City Guidebook. Accessed: March. 23, 2023. [Online]. Available: https://www8.cao.go.jp/cstp/society5_0/smartcity/index.html

[2] PLATEAU. Accessed: March. 23, 2023. [Online]. Available: https://www.mlit.go.jp/plateau/

[3] Azure Digital Twins. Accessed: March.23, 2023. [Online]. Available: https://azure.microsoft.com/ja-jp/products/digital-twins/

[4] 3D City DB. Accessed: March. 23, 2023. [Online]. Available: https://www.3dcitydb.org/3dcitydb/

[5] CityGML. Accessed: March. 23, 2023. [Online]. Available: https://www.ogc.org/standard/citygml/

[6] BIMServer. Accessed: March. 23, 2023. [Online]. Available: https://github.com/opensourceBIM/BIMserver

[7] Building Topology Ontology Draft Community Group Report 28 June 2021. Accessed: March. 23, 2023. [Online]. Available: https://w3c-lbd-cg.github.io/bot/

[8] NGSI-LD. Accessed: June. 21, 2023. [Online]. Available: https://ngsi-ld-tutorials.readthedocs.io/en/latest/

[9] FIWARE-NGSI v2 Specification. Accessed: June. 21, 2023. [Online]. Available: https://fiware.github.io/specifications/ngsiv2/stable/

[10] DTDL: Accessed: March. 23, 2023. [Online]. Available: https://azure.github.io/opendigitaltwins-dtdl/DTDL/v3/DTDL.v3.html

[11] Real Estate Core: Accessed: July 4, 2023. [Online]. Available: https://github.com/Azure/opendigitaltwins-building

[12] M. Ponjavic, A. Karabegovic, E. Ferhatbegovic, and I. Besic, "Spatial Data Integration in Heterogeneous Information Systems' Environment," in 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), May 2019, doi: 10.23919/MIPRO.2019.8756736.

[13] I. Bizid, S. Faiz, P. Boursier, and J. C. M. Yusuf, "Integration of Heterogeneous Spatial Databases for Disaster Management," in SeCoGIS 2013 7th International Workshop on Semantic and Conceptual Issues in GIS, Nov. 2015, Hong Kong, China, doi: 10.1007/978-3-319-14139-8_10.

[14] J. Zhu, H.-Y. Chong, H. Zhao, J. Wu, Y. Tan, and H. Xu, "The Application of Graph in BIM/GIS Integration," *Buildings*, vol. 12, no. 12, p. 2162, Dec. 2022, doi: 10.3390/buildings12122162.

[15] F. Beck, A. Borrmann, and T. H. Kolbe, "The Need for a Differentiation Between Heterogeneous Information Integration Approaches in the Field of 'BIM-GIS Integration': A Literature Review," in ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. VI-4/W1-2020, Sep. 2020, doi: 10.5194/isprs-annals-VI-4-W1-2020-21-2020.

[16] S. Meschini, D. Accardo, M. Avena, E. Seghezzi, L. C. Tagliabue, and G. M. Di Giuda, "Data integration through a bim-gis web platform for the management of diffused university assets," in Proceedings of the 2022 European Conference on Computing in Construction, 2022, doi: 10.35490/EC3.2022.217.

[17] T.-W. Kang, S.-H. Park, and C.-H. Hong, "BIM/GIS-based Data Integration Framework for Facility Management," in Proceedings of the 2016 GEOProcessing: The Eighth International Conference on Advanced Geographic Information Systems, Applications, and Services, 2016, pp. 100-105. ISBN: 978-1-61208-469-5.

[18] M. La Guardia and M. Koeva, "Towards Digital Twinning on the Web: Heterogeneous 3D Data Fusion Based on Open-Source Structure," Remote Sens., vol. 15, no. 3, p. 721, Jan. 2023, doi: 10.3390/rs15030721.

[19] A.-H. Hor, G. Sohn, P. Claudio, M. Jadidi, and A. Afnan, "A Semantic Graph Database for BIM-GIS Integrated Information Model for an Intelligent Urban Mobility Web Application," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV-4, pp. 89-96, Sep. 2018, doi: 10.5194/isprs-annals-IV-4-89-2018.

[20] A.-H. Hor, A. Jadidi, and G. Sohn, "BIM-GIS Integrated Geospatial Information Model Using Semantic Web and RDF Graphs," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. III-4, pp. 73-79, Jun. 2016, doi: 10.5194/isprs-annals-III-4-73-2016.

[21] S. Schilling and C. Clemen, "Practical Examples on BIM-GIS Integration Based on Semantic Web Triplestores," in The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLVI-5/W1-2022, pp. 211-216, Feb. 2022, doi: 10.5194/isprs-archives-XLVI-5-W1-2022-211-2022.

[22] H. Arenas, B. Harbelot, and C. Cruz, "Implementing a Semantic Catalogue of Geospatial Data," in Proceedings of the 10th International Conference on Web Information Systems and Technologies (WEBIST-2014), Barcelona, Spain, 2014, pp. 152-159, doi: 10.5220/0004820101520159.

[23] V. Kopsachilis and M. Vaitis, "GeoLOD: A Spatial Linked Data Catalog and Recommender," Big Data Cogn. Comput., vol. 5, no. 2, p. 17, Apr. 2021, doi: 10.3390/bdcc5020017.

[24] RDF Accessed: March. 23, 2023. [Online]. Available: https://www.w3.org/RDF/

[25] ArangoDB. Accessed: March. 23, 2023. [Online]. Available: https://www.arangodb.com/

[26] Flask. Accessed: March. 23, 2023. [Online]. Available: https://github.com/pallets/flask

[27] IfcOpenshell-python. Accessed: March. 23, 2023. [Online]. Available: https://ifcopenshell.github.io/docs/python/html/ifcopenshell-python.html

[28] RDFlib. Accessed: March. 23, 2023. [Online]. Available: https://rdflib.readthedocs.io/en/stable/

[29] Pytyon-Arango. Accessed: July. 4, 2023. [Online]. Available: https://docs.python-arango.com/en/main/

[30] JSON-LD. Accessed: March. 23, 2023. [Online]. Available: https://json-ld.org/

[31] trimesh. Accessed: March. 23, 2023. [Online]. Available: https://github.com/mikedh/trimesh

[32] PostGIS. Accessed: March. 23, 2023. [Online]. Available: https://postgis.net/

[33] psycopg2. Accessed: March. 23, 2023. [Online]. Available: https://www.psycopg.org/

[34] glTF. Accessed: March. 23, 2023. [Online]. Available: https://github.com/KhronosGroup/glTF/blob/main/specification/2.0/README.md

[35] Swagger. Accessed: March. 23, 2023. [Online]. Available: https://swagger.io/

[36] SPARQL. Accessed: March. 23, 2023. [Online]. Available: https://www.w3.org/TR/sparql11-overview/

[37] ifcdata-management-systems. Accessed: March. 23, 2023. [Online]. Available: https://github.com/ryouyabe/ifcdata-management-systems

Reply Letter for the Review

October 2, 2023

Dear Reviewers, Journal of Architectural Informatics Society.

Thank you for providing us with the opportunity to submit our manuscript for publication in your journal. We appreciate the time and effort devoted by the reviewers to evaluating our manuscript. We are pleased that the manuscript has been considered for publication, and we would like to address the comments and concerns raised by the reviewers.

We have carefully considered each of the reviewers' suggestions and criticisms and have made substantial revisions to the manuscript in response. In this letter, we provide point-by-point responses to the reviewers' comments and explain the changes made.

We hope that the revised manuscript is now suitable for publication in your journal. Thank you again for your time and consideration and look forward to hearing from you.

yours sincerely
Ryota Yabe
SECOM CO., LTD.

Responses to reviewer 1

The validation using the prototype is described meticulously, and the content of the verification is highly commendable. However, on the other hand, I find that important sections such as previous research and discussion are lacking in the paper.

Thank you for your thoughtful review and appreciation of the validation we performed using our prototype. Your feedback prompted us to make the manuscript more comprehensive and insightful. We acknowledge the gaps that you pointed out in the sections concerning previous research and the discussion. Thank you for your invaluable comments.

In response to your comments, we have extensively revised and expanded these sections to provide a more thorough examination of prior work. We now provide a deeper analysis in the discussion section. Specifically, we added references to seminal and recent studies in the field to bolster the literature review. This not only places our work in a broader context but also highlights its novelty and significance.

The discussion section now elaborates on the implications of the operational verification results of the implemented prototype system for the envisioned challenges and identifies the limitations and future directions of our research.

We hope that these revisions adequately address your concerns and enhance the overall quality of the manuscript. Your feedback has helped us refine our paper, and we are grateful for the time and effort you invested in reviewing our work.

I believe that it provides highly important insights for realizing the digital twin. I consider the following points to be lacking in the paper. Therefore, I will evaluate it again under the condition of the following revisions.

Thank you for recognizing the potential importance of our work in realizing digital twins and providing constructive feedback. We value your insight and understand the importance of addressing these points. We are committed to making the necessary revisions and improvements to the manuscript following your suggestions. We appreciate the expert feedback and time invested in reviewing our work and ensure that your feedback is duly considered.

1) In the paper, a comprehensive examination of previous research is essential to establish a clear background for the current study. The references cited in this paper's bibliography are mostly composed of specifications and explanations. Research papers within this field are by no means scarce. The author should select and review several papers deemed significant, providing an overview of the current state of research within this field and articulating the distinctions between those studies and the present research in Chapter 2.

Thank you for your insightful comment regarding the necessity for comprehensively examining previous research to establish a clear background for our study. We understand the importance of situating our work within the existing body of knowledge and articulating the unique contributions of this study.

In response to your feedback, we have introduced two new sections, 2.4 and 2.5, in Chapter 2. Section 2.4 now provides a detailed review of key research papers on heterogeneous spatial data integration, while Section 2.5 focuses on surveys of data catalogs for spatial data retrieval. Through these additions, we aim to offer a thorough overview of the current state of research in this field. In addition, we articulate the distinctions between the studies discussed and our present research, thereby highlighting the novelty and significance of our work. The References section of this paper has been meticulously updated and expanded to include additional references.

By expanding the literature review in this manner, we believe that we have addressed the gap you have pointed out and enriched the contextual foundation of the study. These revisions undoubtedly enhance the comprehensibility and rigor of our manuscript, and we hope they satisfy your expectations.

2) The paper should address in the 'Discussion' section what conclusions have been reached regarding the initial challenges. In this paper, only a few lines are dedicated to discussing the verification results. Please provide a detailed discussion, such as identifying which aspects of the proposed methodology were successful in addressing issues related to data discoverability in the context of digital twins. Also, it might be appropriate to incorporate the content from Chapter 6, 'Future Challenges,' into the discussion.

Thank you for your valuable feedback. We understand the importance of a thorough discussion on the conclusions reached regarding the initial challenges posed in our study, particularly concerning the verification results and their implications for data discoverability within digital twins.

Based on your suggestions, we have extensively revised Chapter 5. In the revised Chapter 5, we provide a detailed discussion on the proposed methodology to address the identified issues, thereby offering a clearer understanding of the findings. Additionally, we highlight some of the unfulfilled tasks, establishing a connection to future efforts, as elaborated in Chapter 6.

Furthermore, we altered the title of Chapter 6 from 'Future Works' to 'Future Challenges,' which we believe better encapsulates the content of the chapter and aligns with the expanded discussion on unresolved challenges and prospective directions for future research.

We hope that these revisions provide a more thorough and insightful discourse on the key topics and enhance the overall coherence and contribution of the manuscript.

Responses to reviewer 2

> This paper proposes a solution to the issue of data search compatibility and its difficulty resulting from the coexistence of multiple description specifications in the field of digital space description, particularly focusing on Building Information Modeling (BIM), which is commonly used in architecture and construction industry. The proposal is timely and the direction it suggests holds significant societal potential. In current society, the methods for describing both geometric data and attribute data vary significantly depending on the specification systems. In a social circumstance where the transmission, editing, and processing of higher-dimensional information are becoming increasingly valuable, this proposal represents an important step towards the extraction and accessibility increase of these informatic relationships. While the scope covered by this proposal is still limited, its approach is concrete and feasible, as it combines existing technologies to provide a generic method for describing the linkage between different geometric data and attribute data based on specific specifications.

Thank you for your insightful review and recognition of the timeliness and societal potential of the proposed solution to the challenges of data search compatibility in digital space descriptions, particularly within the realm of Building Information Modeling (BIM). Your summary eloquently describes the core objectives of our study.

We appreciate your acknowledgement of the concrete and feasible approach employed in leveraging existing technologies to propose a generic method that describes the interconnection between different geometries and attributes of data. We agree with your observation regarding the limited scope of this proposal.

To address the points raised in Chapter 6, we discuss the ongoing efforts and steps envisioned towards expanding the scope of our approach to further enhance the accessibility and extraction of the information on relationships in this domain. We believe that elaborating on future efforts will help provide a clearer roadmap for addressing the identified limitations in subsequent research endeavors.

We are grateful for your encouraging feedback and constructive observations, which have provided a richer context for our work and its potential impact. Your review has indeed sharpened our focus, helping us articulate the broader implications of our research more effectively.

> This paper represents an ambitious attempt to construct a universal descriptive interface, particularly focusing on enhancing searchability, in an area where the definition of the interrelation between geometric data and attribute data has not been commonly systematized. The methodology is properly described, and the proposed methodology is validated. The obtained results are explicitly displayed.

Thank you for your insightful recognition of the ambitious nature of our work and its focus on enhancing searchability within a domain in which the interrelationships between geometries and attributes of data are yet to be systematically defined. We appreciate your positive remarks regarding the clarity of the methodology description, validation of the proposed methodology, and the explicit presentation of the results obtained.

Your acknowledgment encourages us to pursue this research direction further, aiming to significantly contribute to constructing a universal descriptive interface, which we believe holds promise for advancing data retrieval in the realm of digital space description, particularly in Building Information Modeling (BIM).

We are grateful for your constructive feedback and encouragement, as it validates the effort and rigor invested in this research. Your positive review has provided a significant morale boost as we work towards further refinement and expansion of the concepts presented in this paper.

> The proposal is well-suited to the needs of an era characterized by rapid diversification in description specifications, objectives, and applicable industry domains. Especially in the field of BIM, effective standardization in the methods of describing the meaning and usage of its form data has not been achieved at the moment. Additionally, it is difficult to anticipate effective cross-software standardization would be established considering the diversity of their current data and industry structure. In this context, this proposal is commendable for its approach to using the IFC (Industry Foundation Classes), which has gained recognition as a suitable standard description specification in the BIM field, as a foundation. It suggests linking the form and attribute descriptions to an existing and widely-used search infrastructure, aligning with the current state of the industry. Looking ahead, when architectural form and attribute data are expected to be regularly used in fields beyond construction, such as autonomous driving and XR, the ability to search without being confined to specific BIM specifications is expected to become even more valuable.

Thank you for your insightful analysis and appreciation of the timeliness and relevance of our proposal in the context of evolving industry standards and applications, particularly in BIM. Your thorough understanding of the industry's challenges and potential broader applications of architectural forms and attribute data beyond construction, such as autonomous driving and XR, aligns well with the broader vision of our research.

We concur with your observation regarding the challenges surrounding effective cross-software standardization called by the diverse nature of current data and industry structures. Your commendation of our approach for utilizing Industry Foundation Classes (IFC) as a foundational standard is greatly appreciated.

Following the detailed review process, we have enriched the discussion in Section 5.6 of the paper to reflect on the necessity of standardizing data provision APIs across different domains to further enhance data reusability. This section delves deeper into the challenges and potential pathways for improving data searchability and reusability across various service domains during the operational phase.

We hope that this enriched discussion provides a more thorough understanding of the challenges and possible solutions, aligned with evolving industry standards and applications. We are sincerely grateful for your thorough and constructive feedback, which has helped refine the discourse in our manuscript.

> On the other hand, while IFC has gained recognition as a standard BIM interoperability specification, it is important to note that not all data from real projects are necessarily converted to IFC. Moreover, one must consider the reality that relevant information can be lost during the conversion process to IFC. Additionally, due to the nature of BIM as a static database, in the scenarios envisioned in this paper, it is acceptable to assume that a reasonable amount of time may be allowed for conversion and searching. However, if the purpose of the search extends to the areas where latency rate become more critical, such as autonomous driving and XR (Extended Reality), there will be a need for further consideration regarding how well the information transfer and conversion processes can adapt to the dynamic responsiveness. This will involve a discussion of data interface strategy between areas that are more suitable for adequate preparation and areas that require instant responsiveness.

Thank you for your insightful observations regarding the limitations associated with the IFC standard and the static nature of BIM, especially when considering more dynamic, latency-sensitive scenarios such as autonomous driving and XR. We acknowledge these limitations and appreciate your suggestion for a more robust data interface strategy that can adapt to the different requirements of data responsiveness.

1.Data Conversion Limitations:

We recognize that not all project data will be converted to IFC and that it is a reality that information will be lost during such a conversion. Although our methodology considers a wider range of spatial data types than IFC data, we recognize that the conversion of all spatial data types is an important issue.

In Chapter 6, we discuss the need to establish a standard process for data generation to ensure that building professionals are compliant in mitigating information loss and improving data quality.

2. Dynamic Responsiveness:

Your comment highlights the necessity of a more flexible data interface strategy, especially for applications that demand lower latency and higher dynamic responsiveness.

As a proactive step towards this, a proof-of-concept (PoC) is currently being carried out under the Green University of Tokyo Project (GUTP), which is an industry-academia collaborative initiative. This PoC aims to develop a standardized process for data generation and a data interface strategy tailored to different application requirements.

Insights from this PoC will help refine our approach to better cater to the diverse needs across various domains, including those requiring real-time data processing and interaction.

3. Addressing Metadata Limitations:

- We agree that even with refined data generation/conversion processes and improved data interface strategies, the loss of information content in metadata, compared to the original data, remains a limitation. We believe that ongoing and future efforts within and beyond the GUTP will continue to explore solutions to minimize information loss and optimize data representation to enhance searchability and usability across diverse applications.

We appreciate your thoughtful feedback and nuanced understanding of these challenges. This has provided us with a richer perspective, enabling us to refine our discussion in the manuscript to better address real-world complexities surrounding data interoperability and responsiveness.

> More fundamentally, as pointed out in this paper, the linkage between form and attribute information is still based on the assumption of an intuitive process as a basis. It should be noted that addressing the fluctuations in ontology due to this human-process remains unresolved at the current stage.

We appreciate your insights into the role of human intuition in linking form and attribute information and recognize that ontology variation due to human processes remains an open problem. To address this concern, it is important to standardize metadata generation and association processes in the future and implement functionality to support these processes across different platforms. Through such standardization and cross-platform implementation, we believe it is important to provide a more structured and less subjective basis for the association of form and attribute information, thereby reducing the influence of human intuition and the resulting ontology variation. In addition, we need to explore automated solutions and objective methodologies to ensure consistency and accuracy in linking form and attribute information. The insights gained from your feedback will be invaluable in guiding our ongoing and future work in this area.