BIM-CITYGML DATA INTEGRATION FOR MODERN URBAN CHALLENGES

SUBMITTED: March 2019
REVISED: May 2019
PUBLISHED: June 2019 at https://www.itcon.org/2019/17
EDITOR: Amor R.

Puyan A. Zadeh, Adjunct Professor,
Department of Civil Engineering, University of British Columbia;
p.zadeh@civil.ubc.ca, https://www.civil.ubc.ca/faculty/puyan-zadeh

Lan Wei,
Google Canada;
weilanweilan@gmail.com

Arianne Dee,
Independent Researcher;
ariannedee@gmail.com

Rachel Pottinger, Associate Professor,
Department of Computer Science, University of British Columbia;
rap@cs.ubc.ca, https://www.cs.ubc.ca/~rap

Sheryl Staub-French, Professor,
Department of Civil Engineering, University of British Columbia;
ssf@civil.ubc.ca, https://www.civil.ubc.ca/faculty/sheryl-staubfrench

SUMMARY: *Modern cities require innovative urban design and development approaches that are efficiently tailored for neighborhood needs. To achieve this, decision makers must deal with information from both the micro (building asset) and the macro (neighborhood) levels, consequently deal with two very different information scopes and standards. This paper addresses this issue and introduces a new conceptual approach for developing a hybrid information infrastructure by integrating building design data, in the form of ifcXML, and 3D neighborhood models, in the form of CityGML. This paper uses examples from the operations and maintenance domain to explain the need for data integration to support decision makers at the neighborhood level by providing access to a wide range of detailed data, starting from the neighborhood scale and zooming in to a room in a building. The BIM-CityGML Data Integration (BCDI) approach that is introduced in this research satisfies both geometric and non-geometric (semantic) information queries in real time. This feature distinguishes BCDI significantly from related works that mainly focus on data conversion from one source to another. Furthermore, this work provides deep insights into the data structure of ifcXML and CityGML and discusses data mapping issues between these two common data standards.*

KEYWORDS: *Building Information Modeling, BIM, IFC, ifcXML, CityGML, Geographic Information System, GIS, Big Data, Data Integration, Decision-Making Support, Urban Design, Urban Facility Management.*

REFERENCE: *Puyan A. Zadeh, Lan Wei, Arianne Dee, Rachel Pottinger, Sheryl Staub-French (2019). BIM-CITYGML data integration for modern urban challenges. Journal of Information Technology in Construction (ITcon), Vol. 24, pg. 318-340, http://www.itcon.org/2019/17*

1. INTRODUCTION

In modern urban design and management, dealing with information from different scales is a particular challenge faced by today's practitioners. Modern technology and interconnectivity is taking over different aspects of our daily life. As a consequence, modern urban design and management professionals must work with large amounts of heterogeneous and unprocessed information. This information can be overwhelming in the decision-making process. Therefore, in the era of the IoT (Internet of Things), Digital Twins and Smart Cities, there is great potential for creating new approaches to support stakeholders in making informed decisions while planning at the neighborhood scale for achieving progressive urban goals.

One practical example in this respect is the heterogeneous information that designers need to design district energy centers (see Section 2). District energy centers, as opposed to having separate mechanical rooms in every building, can improve the energy efficiency of a neighborhood and decrease the impact of built spaces on the environment. However, designing district energy centers with interconnected buildings and assets requires practitioners to handle information from different scales, i.e., from the building asset level to the neighborhood level. FIG.1 illustrates the different information sources required for supporting decision-making processes related to the urban design and management.

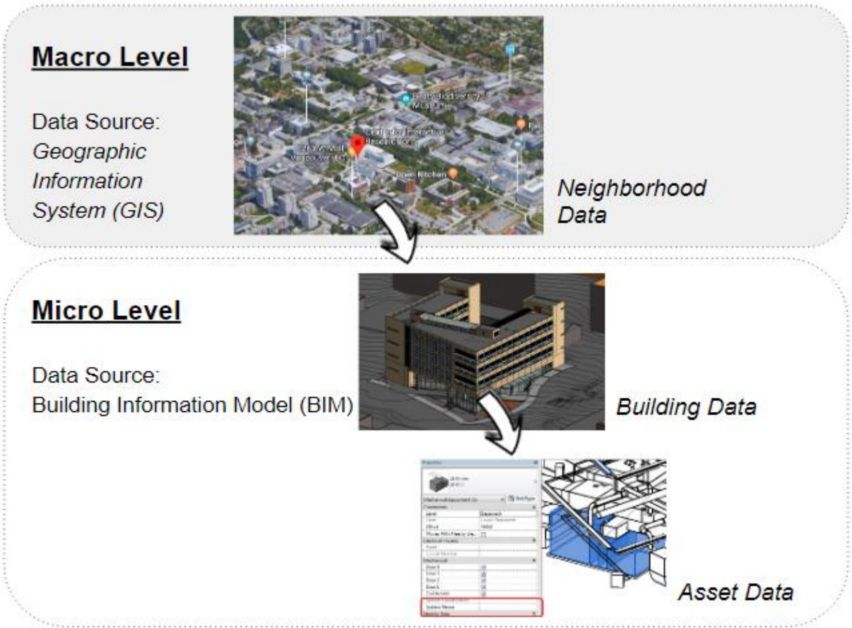

FIG. 1: Different information levels and sources needed for addressing modern urban design and management challenges

Implementing the envisioned interconnectivity needed for a neighborhood to achieve improvements such as district energy centers requires information from the design, construction, and operation phases. In particular, from the facilities in that neighborhood, it is necessary to have access to information from the micro (asset) level and the macro (neighborhood) level at the same time. At present, however, these two information levels are treated as two separate scopes, and as a consequence, their information standards and tools have been developed independently. For instance, IFC (Industry Foundation Classes) (buildingSmart Alliance, 2013a) is the most common information standard on the micro level, while CityGML, developed by the Open Geospatial Consortium (OGC, 2012a), is one of the most common information standards for GIS (Geographic Information System) on the macro level. These two standards have completely different structures and philosophies that make the information exchange between IFC-based and CityGML-based tools tremendously challenging. Hence, there is a strong need to facilitate the interoperability between these two information levels, which is necessary to support decision-making processes related to modern urban neighborhoods.

The vast majority of data structures within the AECO (Architecture, Engineering, Construction and Owner-operated) sector follows the IFC standard, which is maintained by the buildingSmart Alliance. Moreover, IFC is adopted by almost all BIM (Building Information Modeling) tools and so is an inevitable standard for dealing with information on the micro (asset) level. To facilitate interactions with the complicated IFC structure, buildingSmart has also developed ifcXML as an XML schema for the IFC.

CityGML is an open data model and XML-based data exchange format that can be used to describe urban and landscape objects along with their spatial and nonspatial attributes, relations, and complex hierarchical structures at five levels of detail (Yao et al., 2018). This makes CityGML a great standard for information modeling and exchange at the macro (neighborhood) level. Integrating BIM and CityGML is a necessary step for creating a complete 3D model of an urban area with detailed building information. In other words, such integration is necessary to develop an infrastructure that allows access to both micro- and macro-level information at the same time. With such integration, the design and management challenges can be addressed at the neighborhood level using highly detailed information. For instance, in addition to the design of district energy centers, integrating BIM and CityGML can assist in managing routine preventive maintenance tasks by exploring the 3D neighborhood data to locate the desired buildings (using CityGML) and zooming into each building to identify the floor and room where the equipment that needs to be maintained is located (using BIM). The 3D neighborhood data that results from the BIM and CityGML data integration can also be used for calculating the optimal route to allow those in the mechanical trades to proceed efficiently from one identified building to another to perform required preventive maintenance.

This paper proposes a new technical approach for BIM and CityGML Data Integration (BCDI). The goal of BCDI is to build a multilevel information infrastructure for operation and maintenance purposes that allows users to explore and retrieve a wide range of detailed data, starting from the neighborhood scale (macro level) and zooming into a room in a building (micro level). This integration enables the simultaneous querying of ifcXML, to retrieve detailed information about a building component, and CityGML, to retrieve geospatial information data.

This paper describes two attempts to develop BCDI:

1. Extending the CityGML schema to incorporate ifcXML concepts, and
2. Retrieving and integrating information from both ifcXML and CityGML at the time of query through a virtual data integration process, and ultimately creating a hybrid information infrastructure.

In particular, the first attempt to develop the BCDI extended the CityGML schema to accommodate the rich semantic building information that the existing conversion tools could not cover. However, as the experiments in this paper show, this approach did not provide satisfying results. It did, however, provide insights that supported the second attempt.

The proposed second, and ultimately successful, approach to BCDI was developed based on an effort to incorporate ifcXML and CityGML together into one hybrid information system over which the required data can be queried simultaneously from both sources. BCDI 's development started by examining previous conversion approaches (de Laat and van Berlo, 2011; El-Mekawy, 2010; Isikdag and Zlatanova, 2009; Nagel et al., 2007) and commercial software products (BIMserver, 2016; OGC, 2012b; Safe Software Inc., 2018). The two most suitable frameworks to use as a starting point to convert from ifcXML or IFC to CityGML are BIMserver (BIMserver, 2016) and Feature Manipulation Engine (FME) (Safe Software Inc., 2018), both of which are considered and discussed in this research.

In this paper, practical motivations for this research are introduced in Section 2 based on actual projects at the University of British Columbia’s (UBC) Point Grey campus in Vancouver, Canada. In the following section, relevant terminologies and concepts are explained, and related works are discussed in Section 4, followed by an overview of the BCDI attempts in Section 5. Preprocessing of the ifcXML and CityGML data for BCDI is explained in Section 6, and Section 7 is a discussion about how to map ifcXML and CityGML to be able to create hybrid information systems. The research conclusions and outlook are discussed in Section 8.

2. PRACTICAL EXAMPLES FOR BIM-GIS INTEGRATION

The authors’ research team has been closely studying and shadowing the operation and maintenance activities on the UBC campus for the past several years (Bai et al., 2017a; Cavka et al., 2015, 2017; UBC-EWS, 2017a; Zadeh et al., 2015, 2016, 2017; Zadeh and Staub-French, 2016). During these studies, the maintenance of essential assets across different buildings, the design circumstances of the UBC Campus Energy Centre (CEC), and the energy design of the neighborhoods in general caught the research team’s particular attention. Below, we describe two practical examples from the UBC Campus to highlight the motivation of the current study to integrate the information from building assets and the neighborhood level by developing the BCDI approach.

2.1 Example 1: Operation and Maintenance on Neighborhood Level

In the operation and maintenance of neighborhoods and building complexes, certain major assets such air handling units (AHUs) and elevators require constant preventive and reactive maintenance. In such cases, it is important to have detailed information regarding the essential assets, as well as their location on a neighborhood map. This coexisting information can help optimize the maintenance process by minimizing the travel time of moving maintenance equipment and personnel from one location to another. In such cases, the trades and facility managers need to know exactly where these assets are located in the different buildings, and they also need to know the different connections between the buildings.

FIG. 2 illustrates such a scenario for UBC's Point Grey campus, where the detailed information related to different assets in the building are provided in BIMs, and the information related to the location of each building is given in the GIS system; combining the information allows the optimized routes to be calculated. However, for such optimization, it is necessary to simultaneously query neighborhood-level information, which is available in a GIS data system, and detailed asset-level information, which is available in a BIM data system. Therefore, the integration of these two data systems is required to address complex tasks, such as optimizing the inspection routes, which is the main objective of our current work that proposes the BCDI concept.

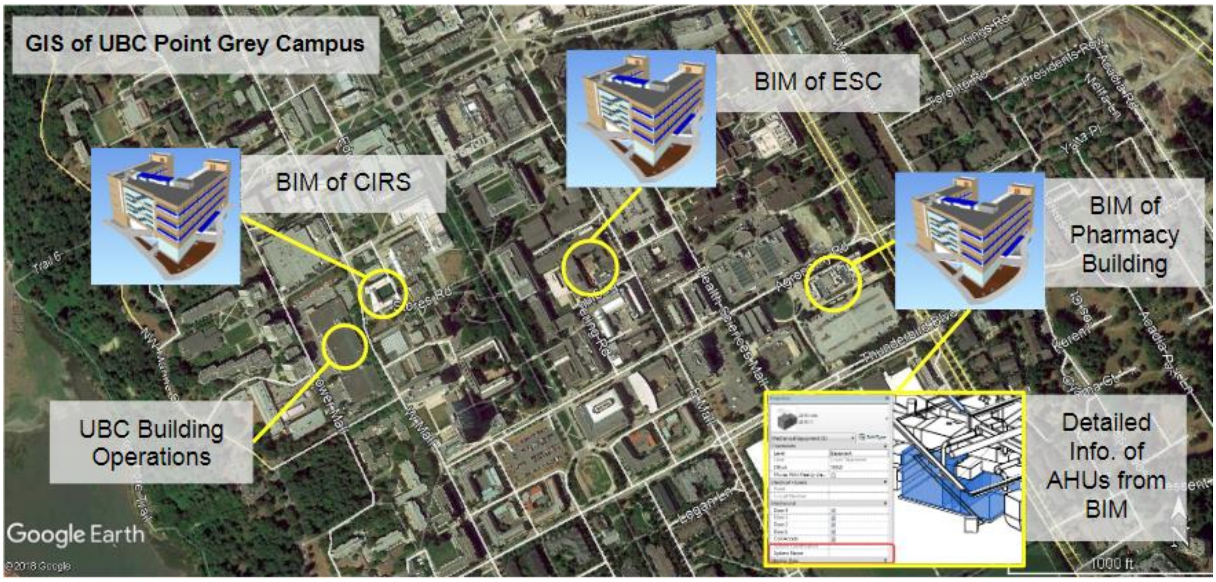

FIG. 2: GIS of UBC Point Grey campus and BIMs of several buildings on campus.

2.2 Example 2: Designing District Energy Centers

Another motivating example for this research is the design and positioning of district heat energy centers. In this regard, UBC's Campus Energy Centre (CEC) is a good example (UBC-EWS, 2017). Even though the CEC is technically capable of meeting all of UBC's heating requirements, currently it is only serving the academic district of the campus and is the primary energy source for this district. FIG. 3 shows a map of the academic district of the UBC campus highlighting all buildings that are connected to the CEC's hot water grid (UBC-EWS, 2017a). The position of the CEC is circled on this map.

The design of district energy centers is a complex process and requires the consideration of various factors. These factors include the detailed information from buildings at a micro level that could be obtained from BIMs, such as information related to building assets, occupancies, and spaces, as well as factors related to broader information at the neighborhood level (macro level), such as information regarding the location of the buildings, geotechnical conditions of the site, and rezoning policies. Considering such factors is tremendously important for design optimization, and in a project such as the CEC, which had a budget of $24 million, any design optimization can have a significant impact.

An important potential optimization in the design of district energy centers is the optimization of the pipes that connect the district energy center with the buildings that are served (turquoise-colored buildings in FIG. 3). The farther a building is from the district energy center, the larger the required pipe diameter and, consequently, the more expensive the costs for materials, transportation, and installation (FIG. 4). In addition, for longer pipes, designers are required by the engineering standards and codes to consider expansion bends to control the pressure (ASME, 2018; BSI Group, 2009). This is an additional motivation for the designers to optimize the length of the pipes, which affects their diameter and other cost factors, by properly choosing the location of the district energy center.

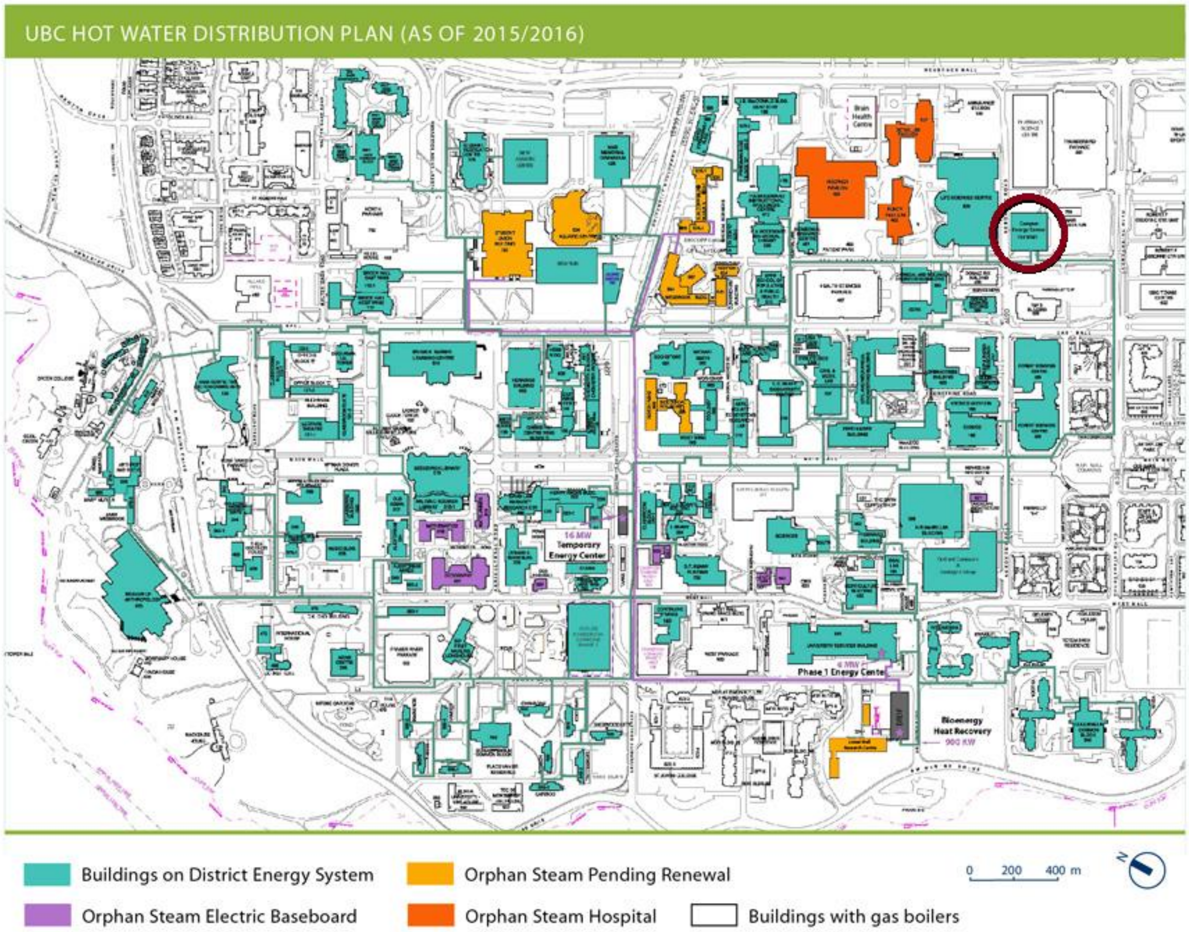

FIG. 3: Map of the academic district at UBC Point Grey campus and the buildings served by the Academic District Energy System (ADES).

FIG. 4: Pipe installation for UBC's Academic District Energy System (ADES).

Obviously, positioning a district energy center is not merely a geometric process; it also requires the consideration of additional information at the neighborhood (macro) level, such as information related to the geotechnical conditions of the site, available free space, long-term district plans, rezoning policies, roads, and accessibility of the site and the surrounding buildings. Therefore, having access to city and landscape models, such as CityGML, in addition to BIM is essential for such optimization.

In the case of UBC's CEC, the location of the energy center was selected without necessarily considering the optimizations described above or the use of city and landscape models. However, the long-term plan for the UBC CEC is to serve the entire 1,000 acres of the UBC campus in the future; from that perspective, the CEC is located in an approximately central position, as shown in FIG. 5.

FIG. 5: Map of the UBC Point Grey campus and the position of CEC.

Undoubtedly, in many projects of this nature, the common process of determining the optimal location for a district energy center relies heavily on the tacit knowledge of experienced designers, with the occasional use of BIMs, but does not take advantage of city and landscape models. This is mainly due to either a lack of available technologies and the associated required expertise or the simplicity of the project layouts and circumstances. Nevertheless, relying on only the tacit knowledge of experts does not necessarily deliver the most optimized solution.

Observing UBC's operation and maintenance processes, as well as the CEC project, along with interviewing relevant stakeholders provided the practical motivation for the authors to find a technical solution to combine and provide information on two different levels of granularity, i.e., the building asset level with BIM and the neighborhood level with CityGML. By introducing technical approaches for implementing BIM and CityGML Data Integration (BCDI), this research aims to support decision-making processes at the neighborhood level, such those described above.

3. BACKGROUND

3.1 Technical Terminologies

Undoubtedly, in many projects of this nature, the common process of determining the optimal location for a district energy center relies heavily on the tacit knowledge of experienced designers, with the occasional use of BIMs, but does not take advantage of city and landscape models. This is mainly due to either a lack of available technologies and the associated required expertise or the simplicity of the project layouts and circumstances. Nevertheless, relying on only the tacit knowledge of experts does not necessarily deliver the most optimized solution.

Observing UBC's operation and maintenance processes, as well as the CEC project, along with interviewing relevant stakeholders provided the practical motivation for the authors to find a technical solution to combine and provide information on two different levels of granularity, i.e., the building asset level with BIM and the neighborhood level with CityGML.

By introducing technical approaches for implementing BIM and CityGML Data Integration (BCDI), this research aims to support decision-making processes at the neighborhood level, such those described above.

AHU (Family, Model, Manufacturer, Width, Length, Height, Maximum Water Temperature, OmniClass Number)

Data “models" or “classes” are the mathematical abstractions used to represent the real world by specific data sets (schemas) (Ullman, 1988). An “instance” of a class has specific values for the defined attributes in a schema. For example, an instance of the class AHU in the IFC schema could have the following specific values:

- *Family: AHU-cirs*
- *Model: ITF-I-2*
- *Manufacturer: Johnson Controls*
- *Width: 2,667.00 mm*
- *Length: 5,003.81 mm*
- *Height: 2,336.80 mm*
- *Maximum Water Temperature: 110 °F*
- *OmniClass Number: 21-04 30 60 10*

According to Ullman (1988), “Datalog” is a declarative logic programming language, which has the following format:

$head :- subgoal_1, subgoal_2, \ldots, subgoal_k$

It has two parts separated by “:-” (a.k.a. a turnstile). The part to the left of the turnstile is the head. The part to the right of the turnstile is the body, which consists of a set of subgoals. In the version of Datalog used in this paper, conjunctive queries, the head and the subgoals are all relations, each of which has a name and a set of variables or constants associated with it. Variables are denoted as plain text. Constants appear in quotation marks. The subgoals in the body can be placed in any order. If the same variable name appears at multiple places in the body, it is a joining variable used to link several tables together. The head of the query returns the attributes specified by the variables in the body of the query. The query can be read as, “If the right hand side of the query holds, then the left hand side of the query holds.” The conjunctive query variation of Datalog is exactly as expressive as select, project, join queries in SQL.

For example, given the relation above, the following query returns the models of all air handling units of the type “*AHU-cirs*”:

Ans-models(mo):- AHU(“AHU-cirs”, mo, ma, wi, le, he, maxTemp, OCN)

This can be seen because the constant “*AHU-cirs*” is in the location for the first attribute (“*Family*”), and is put in quotations. The only value in the head of the query above (“*mo*”) is the same value (variable) as in the second attribute of the relation for the air handling unit, which is the location of the “*model*” number.

Furthermore, it is possible to specify that we are looking for the family and model of specific air handling units (for instance AHUs with square shapes) by the query below, which reuses the same variable in the fourth and fifth positions, thus specifying that the width and length must be the same:

Ans-square-ahu(family, model):- AHU(family, mo, ma, square-value, square-value, he, maxTemp, OCN)

3.2 IFC and ifcXML

Industry Foundation Classes (IFC) was originally developed and promoted by the International Alliance for Interoperability (IAI) to facilitate data sharing throughout a construction project’s lifecycle. IFC has been used to assemble semantic models in a neutral computer language to describe different project components and to represent required information throughout all related processes (buildingSmart Alliance, 2013b; Froese et al., 1999). Currently, as an open standard schema, IFC is commonly used to exchange and share BIM data between different applications. Its standard schema comprises information contributing to a building’s entire lifecycle, including conception, design, construction, operation, maintenance and destruction (buildingSmart Alliance, 2013a; Eastman et al., 2011). The IFC schema describes the components of spatial objects as classes and different arrows as different relationships between classes. An IFC schema defines not only the spatial structures of building elements, their properties, and the relationships between them, it also describes 3D geometric information, such as shape representations and locations, as well as non-geometric attributes, such as material and texture properties.

The term "ifcXML" represents the encoding of IFC into XML format to facilitate information retrieval and, ultimately, querying of the models. The ifcXML schema is still complex, and investigating simple questions can become extremely difficult without using any computational tools.

In this regard, FIG. 6 shows an example for finding the length of a wall, assuming that the user knows the ID of the wall (id="i51"). A search for this ID will direct the user to the element type "*IfcWallStandardCase*". Here, the user finds the name and object type. Other information about the wall can be found by following ID references that are connected to the wall. However, if the user wants to know the length of the wall, following these ID references will lead to a dead end. To find the length of the wall, the user needs to go to the element "*IfcRelDefinesByProperties*" (id="i1560"), which references the wall (id="i51"). "*IfcRelDefinesByProperties*" references "*IfcPropertySet*" (id="i1775"), which further references "*IfcPropertySingleValue*" (id="i2051"), which has the attribute "length." This attribute gives us the length of the wall.

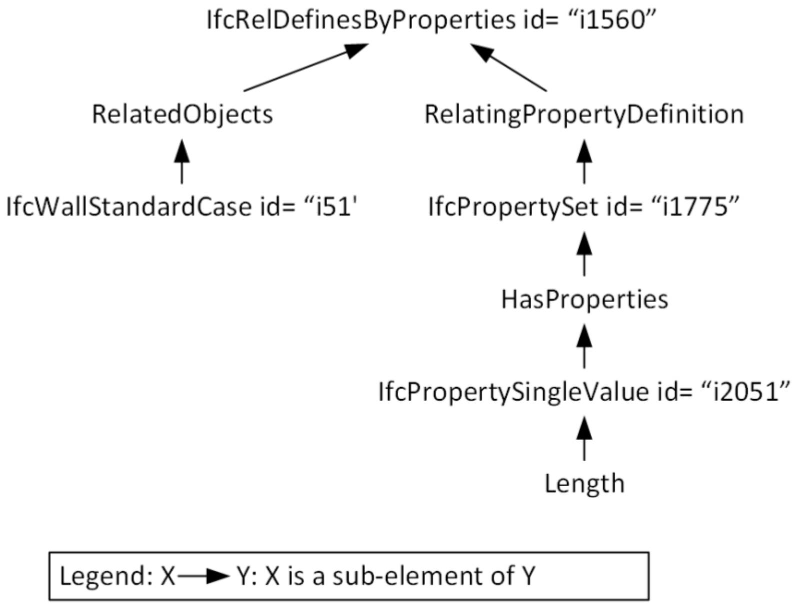

FIG. 6: Finding the length of a wall from an ifcXML data schema.

However, even by tracing through the ID references, not all properties and relationships from models can be extracted from ifcXML. A large part of the information contained within the ifcXML schema is not explicitly expressed, such as whether a wall is clipped or not. Answering such questions requires analyzing the shape representations of the walls, which is not possible in this schema.

3.3 CityGML

CityGML is a semantic information model that is used as a standard representation to store and exchange virtual 3D objects and city models among different applications (Kutzner and Kolbe, 2016; Yao et al., 2018). CityGML uses five different levels of detail (LoD), which are used to represent the city model objects in different degrees of detail regarding both geometry and thematic differentiation. LoD4 is the most complex LoD and contains detailed interior elements of a building, such as rooms, interior doors, interior wall surfaces, stairs, furniture, and electricity units.

Because of these different levels of detail, the CityGML schema is highly scalable; therefore, dealing with building details can be flexible to meet a project's special needs and to perform efficient and sophisticated analysis. Compared to ifcXML, CityGML does not have story information, and it considers rooms as building components rather than spaces, as in ifcXML. CityGML represents the geographic information of spatial objects; thus, the measurement properties of each building component are represented as geographic coordinates instead of lengths and widths.

4. RELATED WORK

As highlighted above in Section 2, having access to detailed information from the micro (building asset) and macro (neighborhood) levels can be a great support for decision makers in addressing modern urban challenges. In a previous related study, our research team worked on integrating building and neighborhood data for energy modeling purposes (Bai et al., 2017). This research demonstrated how such hybrid information systems can be used to optimize the energy consumption of an entire neighborhood. It should be noted that, different than the current research, the proposed solution by Bai et al. (2017) was an iterative approach with several manual steps due to the nature of energy modeling and its related data.

In a more comprehensive study, the same research team provided specific examples for optimization of the energy consumption in parts of the UBC Point Grey campus based on the solution introduced by Bai et al. (2017). This included a consideration of UBC's "Ventilation Policies" and different "Construction Material" options for the purpose of the energy optimization using a hybrid information system. The quantitative results can be found in Bai (2016).

Due to their different domains and objectives, ifcXML and CityGML schemas sometimes use different terminologies and semantics to describe the same matters. Therefore, most of the previous approaches to integrating them are based on the idea of harmonized semantics, and some approaches have focused on a unidirectional method (mostly from ifcXML to CityGML) for the conversion process. In this regard, the remainder of this section describes the most significant research attempts by other researchers and the industry to exchange data between the BIM and GIS systems.

4.1 Research Attempts

In 2007, Nagel et al. (2007) developed algorithms that automatically transform IFC building models into CityGML models through a series of steps that create separate footprints of each story within their own boundary surfaces and are finally merged together. As this research only focused on LoD1 and LoD2 of CityGML, the purpose of the algorithms was to create a geometrically and semantically valid representation of LoD1, which could also be applied to LoD2.

Later Isikdag and Zlatanova (2009) extended the work of Nagel et al. (2007) by proposing a framework for automatically generating building semantics and components in CityGML from corresponding BIM representations. Since CityGML and IFC models are designed for two different domains, they have a very diverse range of object classes and cannot be directly and easily mapped to each other. Thus, Isikdag and Zlatanova (2009) generated semantic and geometric mappings for each CityGML LoD separately; therefore, the same object in one schema could be mapped to different objects in other LoDs. In order to simplify and facilitate the conversion process for each LoD, Isikdag and Zlatanova (2009) proposed mapping rules to identify all necessary objects and attributes that need to be transformed in both schemas.

In 2009, a team led by Thomas Kolbe at the Technical University of Berlin proposed a framework that incorporates semantic spatial context data into 3D graphic and non-graphic building data sets (Nagel et al., 2009). Their framework also included urban areas. In the same year, Léon van Berlo (2009) presented an application domain extension (ADE) that could convert BIM in the format of ifcXML into CityGML. Since CityGML originally represents building information at a low level of detail, the researchers extended the CityGML schemas with extra objects and properties to represent the rich semantic information coming from ifcXML. However, in Léon van Berlo (2009), there are only a few ifcXML classes that are transformed into CityGML extensions that have real meaning.

Unified Building Model (UBM) was the first framework that fully integrated ifcXML and CityGML in which ifcXML can be traced to CityGML and vice versa (El-Mekawy, 2010; El-Mekawy et al., 2012; El-Mekawy and Östman, 2010). The reference schema in this study is defined as an expressive schema for ifcXML and CityGML semantic models, which is a superset model that is extended to contain all the features and objects from both the ifcXML and CityGML building models with respect to all levels of detail, including inner and outer spatial structures. The integration approach in UBM is performed in two steps, where a building model is first converted from the source model into a UBM and is then converted from the UBM into the target model. UBM is considered as a schema that generates mappings from both data sources (BIM and GIS). The UBM schema can also be extended if there is a demand for transformation from a new schema to ifcXML or CityGML.

BIM was developed to deal with building components, while GIS is used for mapping the surrounding real outer world. Therefore, BIM and GIS use different representations to describe the same spatial objects. Furthermore, they have very diverse objects, classes, and properties. To achieve an accurate and efficient integration, the majority of previous work has focused only on integrating the main building components, such as walls, roofs, doors, and windows, for which ifcXML and CityGML have the same semantic and geometric representations.

Although some entities can be semantically mapped from ifcXML to CityGML, it is still difficult to create the geometric matching relations between them. For example, a pipe that runs across two rooms is represented in CityGML as two thematic objects, because it is observable from both rooms, but in ifcXML they are aggregated into one object.

Because of the different geometric representations that ifcXML and CityGML use, sometimes researchers apply an evaluation function to all possible ambiguous conversions in order to optimize them (Nagel et al., 2009). Although CityGML is capable of representing detailed building information in, at most, LoD4, and is extended to model noise, tunnels, bridges, hydro, and utility networks, it still cannot represent several important components, such as the mechanical elements. In addition, during a conversion process, the properties and parameters attached to the components need to be hosted in the target schema; however, CityGML is not capable of hosting them. Under the above restrictions, the complex schemas and components are beyond the scope of the current research, as are any components with complicated geometric shapes.

In a recent study, Jusuf et al. (2017) used the data transition approach from IFC to CityGML to analyze urban microclimates in relation to neighborhood development. However, this approach covered selective parts of IFC elements and focused only on LoD2.

4.2 Industry Attempts: Commercial Software Products

The Feature Manipulation Engine (FME) offers fast and simple translation between different geospatial data formats to facilitate the interoperability of spatial data (Safe Software Inc., 2018). A typical workflow in FME first identifies the data that needs to be read, then it assesses how to translate the data and what format to write out the data. Section 5.1 describes the attempts in BCDI to translate ifcXML data into the CityGML model using FME.

The Building Information Model server, also called BIMserver, is a free and open-source platform that centralizes information from any building related project (BIMserver, 2016). It uses ifcXML as its core standard building model and stores building information in the format of ifcXML in an underlying database, making it possible to query, merge, and filter all of the BIM models and generate ifcXML files on the fly. It also supports exporting functionality in various formats, including CityGML.

Another commercial software product that converts from ifcXML to CityGML is IfcExplorer (OGC, 2012b), which is an implementation of the conversion introduced by Nagel et al. (2009). IfcExplorer is designed to automatically convert an ifcXML model into a CityGML model by selecting specific LoDs and relevant building elements.

5. APPROACHES TO BIM-CITYGML DATA INTEGRATION (BCDI)

As mentioned above, many decision-making processes at the neighborhood level, such as the practical examples introduced in Section 2, require information from two different levels of granularity. On the micro level, BIM contains detailed information about the buildings and their assets, while on the macro level, GIS provides information related to the neighborhood, such as the position of the buildings or the roads. To address complex information requests, such as for supporting the operation and maintenance of a neighborhood or designing a new district energy center, BCDI needs to incorporate a flexible concept so it can support both:

1. Data transformation inquiries (from ifcXML to CityGML)
2. Data integration inquiries (to enable running queries on both sources simultaneously)

These approaches are introduced in the following sections.

5.1 Approach 1: Data Transformation from ifcXML to CityGML

5.1.1 Overview

The first approach that the authors took was to evaluate and compare the currently available conversion tools and select the ones that are most suitable for reforming the required data. After analyzing all the converted results from the first step, we tried to complement the results by adding the missing parts to ensure sufficient information for answering all sorts of queries regarding facility management and maintenance operations. Although in the final BCDI approach we used different steps, it is instructive to examine this process. FIG. 7 illustrates the architecture of Approach 1.

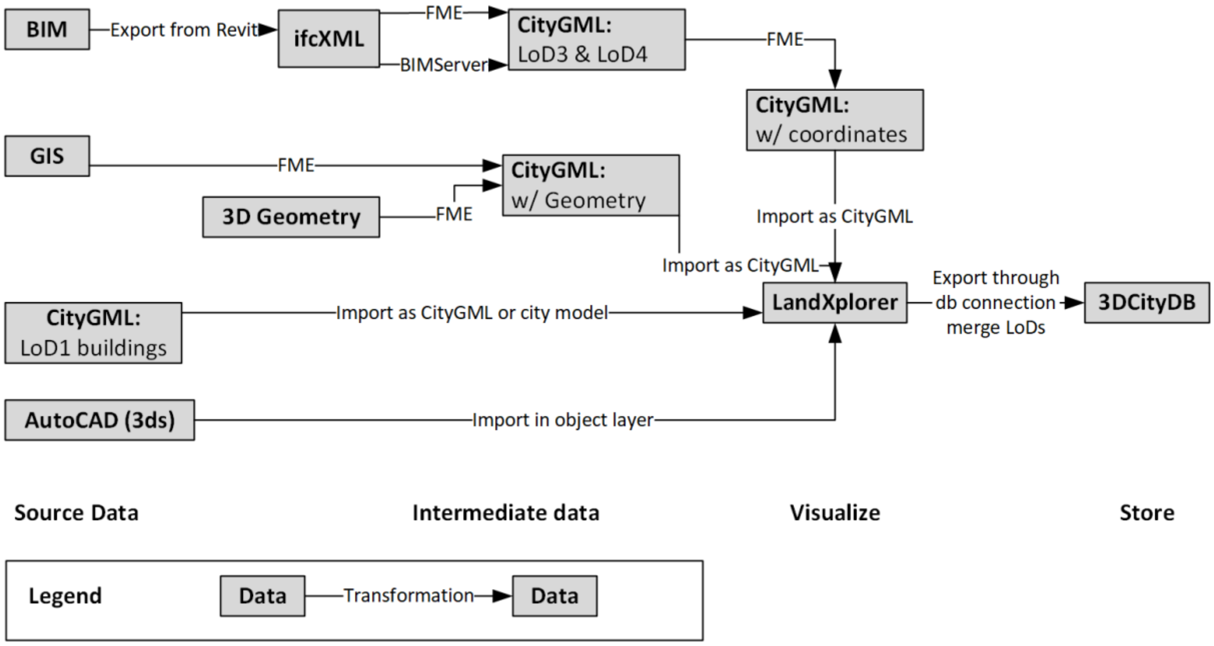

FIG. 7: The concept of Approach 1 for data transformation from ifcXML to CityGML.

5.1.2 Approach Description

After assessing the previous and ongoing conversion approaches introduced in Section 4, we selected FME and BIMserver for this approach. FME Workbench provides various transformers to restructure features and manipulate data through format translation. It also supports the ability of users to change the schemas and mappings. Transformation occurs as the data is passed from reader to writer through a series of these transformers. Approach 1 applies the geometric transformers on the building elements to be converted to CityGML, such as walls, roofs, windows, and doors, to set the levels of detail in CityGML. This includes their corresponding LoD names and defined feature types (e.g., walls, roofs) that correspond to different LoDs. For example, LoD2 includes only walls and roofs. LoD3 adds doors and windows; LoD4 adds rooms.

The conversion process is not always easy and accurate because it is not a simple 1-1 correspondence between each building element. For example, the conversion of ifcXML spatial information into CityGML room information requires identifying the room's associated walls, ceiling, and roof. FME successfully creates the CityGML model in LoD3, but struggles with LoD4. The result correctly processes the buildings' facade textures on the exterior walls and roofs, as well as the stairs and interior walls, with doors and windows attached. However, these features do not contain room information, such as mechanical assets, electrical units, and some geometrically complex building components (e.g., curtain walls).

BIMserver's core model server interprets all the building models in its ifcXML format, stores them in a common database, and exports them in various formats—CityGML being one option. Approach 1 uploads ifcXML files to BIMserver, and it exports the files in the CityGML format in LoD4. Viewing them with the 3D model visualization tools shows that the results do not conform to the standard CityGML schema. For example, in standard CityGML data, the floor surface is part of the boundary surface. In the result in Approach 1, the floor surface is attached to the room. Additionally, the number of floor surfaces and rooms does not correctly match the original ifcXML data.

Autodesk LandXplorer provides various functionalities that can explore, analyze, query, and navigate a 3D virtual city (Autodesk, 2010). It also presents a fundamental raster-based digital terrain model, on top of which it has additional geospatial data, such as buildings, plants, and transportation infrastructure. The geospatial data can be imported and integrated into the city model, as similarly called "city models." LandXplorer can separately represent every LoD of the CityGML model by importing and gradually applying each layer of texture and appearance onto the basic LoD1 city model. LandXplorer offers an exploring panel that provides a hierarchical view of the spatial objects and their attributes within the city model. In addition, it also supports spatial query functions to display the desired buildings or spatial objects that meet certain criteria. For example, LandXplorer can display all of the buildings built before a certain year or those in a certain area. Furthermore, LandXplorer supports a connection to the 3D city database to import, export, and merge CityGML data.

5.1.3 Validating Approach 1

Because LandXplorer has sophisticated navigation ability and query functions, the first BCDI approach transformed all related data models into CityGML and then imported them into LandXplorer to execute complex queries and analyses, as shown in FIG. 7.

The validation test of the ifcXML conversion in this approach was performed on the ifcXML model of the CIRS (Centre for Interactive Research on Sustainability) building at UBC. The initial model was developed in Autodesk Revit. After exporting these files to the ifcXML version 2x2 format using Autodesk Revit, the file size became too large to upload to the BIMserver. It was necessary to reduce the file size; thus, the validation test was conducted on only two floors of the CIRS building, including the mechanical assets. We then converted the model to the CityGML format using FME Workbench and BIMserver. Both conversions resulted in CityGML files that included parts of the building components in LoD4; however, they were not complete, and neither of them included the mechanical information. Below is the analysis of the resulting CityGML files:

1. The mechanical part is completely missing in the CityGML model.
2. The furniture part (chairs, desks) is also missing in the CityGML model.
3. Stairs and railings are missing in the CityGML model.
4. The curtain exterior surface walls cannot be converted to CityGML components.
5. The number of doors does not match.
6. The properties of rooms (only in CityGML) do not match the space (only in ifcXML).
7. For the CityGML model, while the walls, doors, and windows have area measurements, they are actually only geometric point coordinates, whereas in the ifcXML model, they are measured by width, length, and area.
8. There is no story information in the CityGML model.
9. For each wall, window, door or room, there is no associated information about the story level.

5.1.4 Discussion of Approach 1

There are a number of reasons for facing these difficulties in what seems like a simple task. First, BIM and GIS models are originally designed for different domains and purposes and are meant to serve different areas of interest. Because of this, there is a significant technological barrier preventing any automatic transformation between them. BIM is made for detailed building information on a micro level, whereas GIS was designed to represent the real world at a large scale on a macro level, in an efficient and simple way. As a result, GIS cannot accommodate much of the detailed information that ifcXML contains.

Most of the current research projects have considered the transformation of the architectural building elements only, such as walls, spaces, and doors, by concentrating mainly on geometry transformation issues. There is no systematized study on interoperability between the BIM and GIS systems for utility networks and mechanical elements on a neighborhood level.

The conceptual mappings between the ifcXML and CityGML components are too complicated to satisfy both a geometrical and semantic agreement. With regard to geometric representation, every single object in the BIM can be represented as two (or more) different objects in CityGML with respect to different LoDs. Additionally, they use different geometric representations of the same object, which contributes to difficulties in identifying the correspondence between them.

To address some of these issues, we used GeoBIM (de Laat and van Berlo, 2011; Ohori et al., 2018), an extra CityGML extension, in our data transfer approach. Our GeoBIM experimental results show that the extended objects, such as stairs, in the conversion results cannot be displayed in some inspectors, such as LandXplorer, whereas they work well with some other viewers, such as FZKViewer. After converting ifcXML to CityGML using GeoBIM, the size of the CityGML files becomes significantly larger than the original ifcXML files. In fact, the size increases tenfold or more, creating difficulties for the viewers to read the converted files. Thus, even if CityGML could be extended to satisfy our goals in accessing the data on both the micro and macro levels, the viewers such as LandXplore cannot always display them. Therefore, extending CityGML to accommodate additional information from ifcXML is not a viable solution. Moreover, the accumulation of the geospatial information of a whole city or campus, and the detailed information about each building in the campus range, is large enough that it may exceed the maximum capacity of the current applications. Therefore, the efficiency of running the queries on them is very low, and this affects the user experience. Hence, it is not worthwhile to input all of this information into the system at the same time.

5.2 Approach 2: BIM-CityGML Data Integration (BCDI)

To address the shortcomings in Approach 1, in this approach we ultimately built a data integration system to allow both ifcXML and CityGML data to be queried simultaneously while remaining in their own formats (FIG. 8).

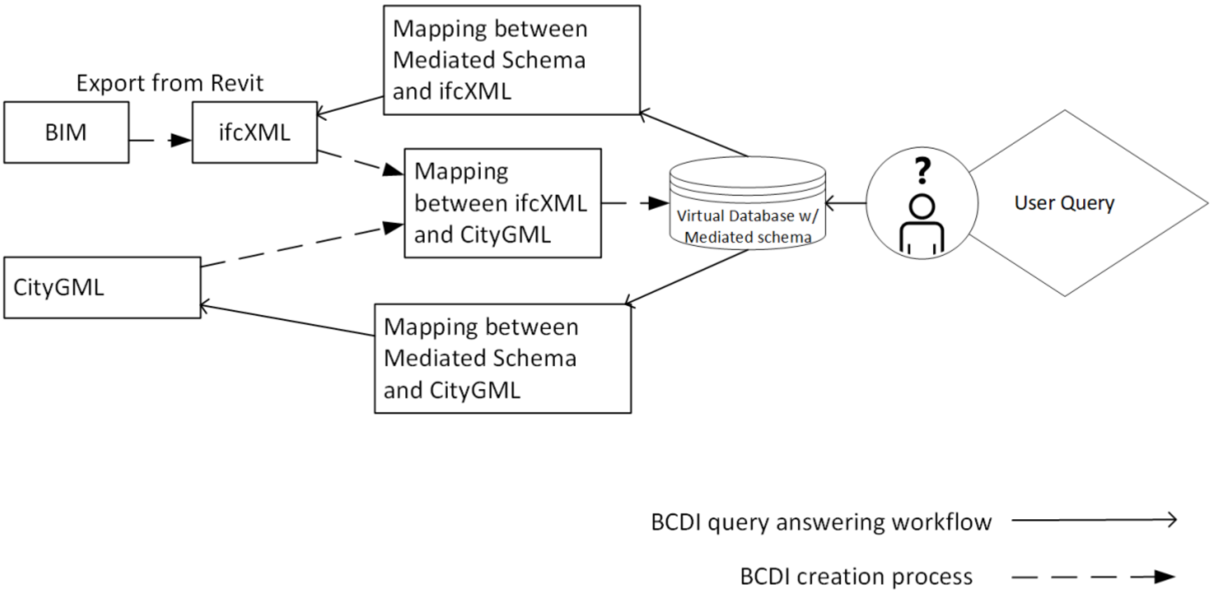

FIG. 8: The concept of Approach 2 for BIM-CityGML Data Integration (BCDI).

There are two common ways to gather data from different sources into one central system. One is to load all of the separate data into a central "data warehouse" and then treat the data warehouse as a traditional database to post queries on. The other way is to use a "data integration" method. In this method, instead of loading all of the data from separate sources into one central location, the data integration system uses a "mediated schema" based on all the source schemas. This mediated schema represents all the information from all data sources in the system, similar to the data warehouse schema; it eliminates duplicate attributes and contains the unique attributes specified to each schema. With the data integration method, all data sources are still stored in their original format, so data integration saves space and transformation time compared to a data warehouse. Furthermore, users can query data from a unified mediated schema without having to deal with the source schemas, the data integration system then translates each query over the mediated schema into queries over the source schemas.

The data warehouse needs to store all the source data in a central data repository, which consumes massive amounts of space. Since answering most maintenance and operations requests requires only a small part of the data from both ifcXML and CityGML schemas, creating a full data warehouse is prohibitively expensive in time and space. Therefore, BCDI uses the integration approach, where the data is left in the CityGML and ifcXML format, and queries over the mediated schema are translated into queries over the underlying CityGML and ifcXML data. Additionally, a data integration system is more flexible than a data warehouse, because its mapping language can represent the implicit corresponding concepts. The following sections describe the BCDI approach in more detail.

6. PREPARING IFCXML AND CITYGML SOURCE SCHEMAS

As described above, both the ifcXML and CityGML schemas must be prepared for integration in order to find commonly needed concepts. Both ifcXML and CityGML are XML (eXtensible Markup Language)-based languages. XML is a semi-structured model that consists of elements that contain sub-elements and attributes. BCDI's preparation process converts ifcXML and CityGML into relational models to have structured relations with suitable attributes. This conversion is necessary because the tools for integrating the schemas use the relational model rather than XML. However, since the transformation was straightforward, this paper does not describe it in detail. The remainder of this paper uses the terms "element" (from XML) and "relation" (from the relational model) interchangeably.

6.1 Preparing the ifcXML Schema

As explained in the example for finding a wall's length in Section 3.2, following the reference paths to extract the relevant data from ifcXML is tedious and time-consuming. Therefore, the first step in BCDI is to create a modified version of the ifcXML schema to expose all of the information that many projects need. This schema, which is basically a representation of the data, is shown in Appendix A.

The modified ifcXML schema is generated based on the attribute sets of spatial objects from Solibri Model Checker, which is software that provides an easy-to-use visualization interface for ifcXML data (Solibri, 2018). There are other similar tools, such as ifcXML Engine Viewer; however, Solibri Model Checker covers more information. In this way, BCDI's modified ifcXML schema contains not only all of the related attributes and elements in ifcXML, it also includes some implicit attributes derived from the original ifcXML schema and even some related elements referenced by the relationships.

At this stage, it is important to identify the scope of the data needed by common applications that users of BCDI may need. Answering the queries regarding BCDI's case scenarios requires the main spatial elements, such as building, building story, space, wall, materials, geometric measurements, and some additional information about the equipment, such as equipment name, model, and manufacturer.

The *IfcProject* in the ifcXML schema is the uppermost container class. The highest spatial container class and other spatial classes, such as *IfcBuilding* or *IfcSite*, are connected to the *IfcProject* through *IfcRelAggregates* relationships. However, since the goal of this paper is not to exhaustively describe ifcXML, this section only describes the hierarchy of *IfcBuilding*. This illuminates the general features and organization of ifcXML, which otherwise can be obscured by the overwhelming complexity of ifcXML.

The spatial containment hierarchy of *IfcBuilding* has a top level that consists of all stories in the building (*IfcBuildingStorey*). All other building elements are aggregated according to the stories in which they are located. The hierarchical structure of a building clearly represents the containment relationships between the building components on different levels. *IfcBuilding* in ifcXML represents a building by providing the overall building information. It also includes references to its building stories through *IfcBuildingStorey*, which itself includes references to all the elements that belong to a certain story. All building elements inherit the following basic attributes: *name*, *type, objectPlacement,* and *representation*.

6.2 Preparing the CityGML schema

The CityGML model is represented in an XML-based format to facilitate the exchange, sharing, storage, and maintenance of the virtual 3D city and landscape models. CityGML provides a comprehensive and extensive representation of the basic entities, attributes, and relations of a 3D city model in terms of their geometry, topology, semantics, and appearance. Similar to ifcXML, CityGML employs a hierarchical structure to generalize the relationships between spatial elements, as well as among aggregation and association relations. Unlike ifcXML, which mainly focuses on building information, CityGML is highly scalable and includes different urban entities, such individual buildings, whole sites, districts, cities, regions, and countries (OGC, 2012c). However, these are beyond the scope of BCDI, and in order to incorporate the ifcXML schema, we only considered building-related schemas in CityGML.

Overall, CityGML represents spatial objects in geometric, appearance, and thematic models. The geometric model defines the geometrical and topological representations of spatial objects in a 3D city model. The appearance model describes the observable properties of the surface of an object in terms of material and texture. CityGML can be extended to express objects that are not defined in its standard specifications by either the concept of generic objects and attributes or Application Domain Extensions (ADE). The thematic model consists of many extension modules, which separately represent different thematic fields within a 3D city model, such as *building, cityFurniture, bridge, generic CityObject, landUse, digital terrain model (relief), transportation, vegetation*, and *waterBody* models.

As explained in Section 3.3, CityGML defines five different consecutive Levels of Detail (LoD), which describe different ranges of spatial objects with respect to the criteria of each LoD. Some objects can be represented simultaneously in more than one LoD, with different levels of geometric detail according to the corresponding LoD. Furthermore, each object can have individual appearances corresponding to each LoD.

In CityGML, a top-level concept is an *AbstractBuilding*. This describes commonly used concepts, such as "usage", "year of construction", "height", and "stories".

Unlike in the ifcXML building schema, CityGML represents buildings and their components differently with respect to each LoD. With the increasing LoDs, more geometric details and semantic objects are added. LoD1 gives only a generalized geometric representation of a building's outer shell, but in LoD2 and LoD3, more geometric details and textures are added to the roofs and walls. These also include some exterior elements, such as balconies and openings. LoD4 represents the interior of a building, the rooms, stairs, and furniture. Compared to walls in ifcXML that are concrete volume objects, in CityGML walls are represented as surfaces. Thus, they are distinguished as interior or exterior walls, and walls are categorized into many types of boundary surfaces by different functions, i.e., wall surface, roof surface, ground surface, floor surface, ceiling surface, interior wall surface and closure surface. Each surface has its corresponding geometric information.

In BCDI, the "3D City Database" is used to store, represent, and manage virtual 3D city models. 3D City Database is a high-performance spatial relational database system that was developed based on Oracle Spatial Database (Oracle, 2018) by following the CityGML schema. The final CityGML schema that BCDI uses to store the CityGML models is provided in Appendix B.

7. CREATING THE MEDIATED SCHEMA MAPPINGS TO IFCXML AND CITYGML

After preparing the ifcXML and CityGML schemas as described in the previous section, the next step was generating the schema mappings. These mappings identify the corresponding information from the two models. Schema mapping is necessary for both data integration and data warehousing. A schema mapping identifies the same or similar attributes from all of the input schemas. Based on Madhavan et al. (2001), the main criteria used in the mapping process are:

- Similarity of names
- Data types
- Constraints
- Schema structures
- Structural and contextual identifiers

The mapping can be one-to-one, one-to-many or indirect. While both models describe the building information, they address different usages and scopes and have different semantic and geometric representations. Therefore, the overlapping specifications between the ifcXML and CityGML models are not simply one-to-one mappings. Previous mapping results between ifcXML and CityGML focused on direct semantic mappings, but in BCDI, we developed implicit mappings on geometric representations. This allows for a more precise alignment of the concepts between the two representations. BCDI modifies existing algorithms introduced in Pottinger and Bernstein (2008) to use schema mappings between ifcXML and CityGML and produces mediated schema mappings between the mediated schema and the source schemas.

7.1 BCDI's Schema Mapping

IfcXML and CityGML vary in their spatial and geometric representations. Every building element in CityGML is expressed by all the surfaces that construct it, but ifcXML is an element-based volume model, in that every building element is a concrete 3D spatial object. This can lead to very different representations of the same information. For example:

- A wall in the ifcXML schema is a 3D object of the wall, but in CityGML a wall is represented as interior and exterior wall surfaces.
- The ifcXML model uses space-enclosing structures, meaning that it applies a hierarchy of structure with regard to their spatial-enclosing relationships, in the order of building, story, space, etc. However, CityGML defines all the building components in parallel under the building. Instead of using the concept of rooms, ifcXML (or in this case, *IfcSpace*) defines an area or space as bounded by walls or an open area. Thus, the actual rooms in the building are not directly mapped to the *IfcSpace* objects.
- IfcXML defines a building consisting of at least one story, and all the spaces are attached to the story they belong to. In contrast, CityGML does not explicitly define the story concept. Instead, CityGML provides an aggregation function to group all building components at a certain height level into a story class. CityGML defines a building installation concept to represent building objects, such as ramps, chimneys, balconies, beams, and columns. IfcXML represents these as individual building elements with different property sets.

The existing mapping approaches are inadequate for BCDI because they indicate only the correspondence of the building components from both schemas; they do not discover the overlapping information of their corresponding attributes. Additionally, they do not consider elements inside the rooms, such as furniture, mechanical equipment, texture, material, and the geometric properties of the building components. Hence, BCDI uses a different approach, as explained below.

First, BCDI identifies the mapping concepts from both schemas, which includes the relations among building, building address, wall, door, stair, furniture, texture, and material. By comparing these with the mappings from previous work, BCDI improves the schema mapping result by indicating the similarities in the texture and material properties of the building components in the two models. Moreover, BCDI generates the overlapping attributes for each concept. Many of these relationships are not simply direct mappings, highlighting the need for the more complex mappings that BCDI uses. Specifically, BCDI uses an extension of the mappings introduced in Pottinger and Bernstein (2008), which is illustrated in Example 1:

Example 1:

Consider the concept of a building address. This concept appears both in ifcXML and in CityGML. The ifcXML definition, "*IFC.BuildingAddress,*" has the attributes:

- *Building_name*
- *Purpose*
- *Description*
- *Addresslines*
- *Postalbox*
- *City*
- *Region*
- *Postalcode*
- *Country*

The CityGML version, "*CityGML.Address*" has the following attributes:

- *Building_name*
- *Street*
- *House_number*
- *Postalbox*
- *Postalcode*
- *City*
- *Region*
- *Country*
- *Xal_source*
- *Multipoint*

Both representations have some similar items (e.g., "*Building_name*") but also some different items. For example, CityGML contains separate entries for "*House_number*" and "*Street*", whereas ifcXML contains the single concept of "*Addresslines*". The mapping to combine the concepts from these elements is as follows:

- *BuildingAddress(Building_name, Addresslines, Postalbox, City, Region, Postalcode, Country)* **:-** *IFC.BuildingAddress(Building_name, Purpose, Description, Addresslines, Postalbox, City, Region, Postalcode, Country)*
- *BuildingAddress(Building_name, House_Number+Street, Postalbox, City, Region, Postalcode, Country)* **:-** *CityGML.Address(Building_name, Street, House_number, Postalbox, City, Region, Postalcode, Country,Xal_source, Multipoint)*

The semantics of Example 1 are such that everything that occurs before ":-" in each line describes the common concepts, whereas what occurs after ":-" is the representation in the input schemas (ifcXML and CityGML) that are under consideration. On either side of ":-", the items inside the parentheses are attributes of the relations. Example 1 covers only one relation in each mapping (*IFC.BuildingAddress* in the first entry and *CityGML.Address* in the second). Some of BCDI's mappings also use multiple relations in ifcXML and CityGML to create a single concept in the mediated schema.

The schema-mapping language used in BCDI is a set of conjunctive queries in Datalog (see Section 3.1), which is a common approach in the schema mapping literature. Because Datalog is a very powerful language, it is capable of representing BCDI's complex arithmetic mapping expressions. In this way, the schema mapping defines any overlapping elements of the same concept in terms of source relations. The heads of a mapping on one concept are the same, which is a relation of all the overlapping attributes from all the relations for this concept. The bodies of each query are expressions of the relations of this concept from each respective schema.

In addition to the directly corresponding relations in both schemas, BCDI also retrieves some implicit shared or overlapping elements and their equivalent attributes. While CityGML does not explicitly represent the width and height of a building component, the ifcXML schema defines the geometric measurements associated with each building element. However, the geometric information of building components is stored in the table *SURFACE_GEOMETRY* in the CityGML schema, which has an attribute of *GEOMETRY* of type *SDO_GEOMETRY. SDO_GEOMETRY* defines the geometry of any surface, which is represented by polygons.

To obtain the area of a polygon, BCDI passes a parameter called "*Tolerance*" to the area function. This parameter defines a level of precision with the spatial data. For example, *SDO_GEOM.SDO_AREA(shape, 0.005)* sets a tolerance of 0.005 for the polygon represented by "*shape*". At this point, BCDI can build a connection between the two models in terms of geometric representation.

To simplify the mappings shown in this paper, we only show the mappings for the concepts of building, building address, wall, door, room, stair, furniture, texture, and material. However, BCDI also has mappings that enumerate all the types of boundary elements, such as roof, ceiling, curtain wall and slab. This paper also omits non-movable objects, such as interior stairs, railings and pipes, which are classified into the *IntBuildingInstallation* class in the CityGML schema. The resulting simplified mapping is shown in Appendix C. In summary, BCDI's schema mappings make the following contributions to the integration of the ifcXML and CityGML models:

- Even though the ifcXML and CityGML have different geometric representations of their objects, BCDI identifies their basic corresponding measurements, e.g., length, height, width, perimeter, and area.
- In contrast to previous mappings, BCDI's mapping enriches the overlapping information on attributes of the primary building components by a thorough study of both schemas.
- BCDI specifies the material and texture information of building components in the mappings in terms of texture type, texture image, texture coordinates, texture wrapping mode, transparency of material, etc.
- BCDI applies arithmetic, complex mapping expressions to identify elements' implicit corresponding relationships.
- Since BCDI's schema mappings are bidirectional, BCDI can flexibly integrate data from both ifcXML and CityGML into a comprehensive and rich building representation.

7.2 BCDI's Mediated Schema

The mediated schema in this work was developed based on the work of a previous study (Pottinger and Bernstein, 2008). The mediated schema in our study provides a unified representation of all related source data for data query purposes without having to know all of the source schemas. For example, a mediated schema for ifcXML and CityGML should allow concepts from both to be queried simultaneously. This approach also creates mappings to the existing schemas so that queries over the mediated schema can be translated into queries over the source schemas. The mediated schema preserves all the information in all the source schemas, which covers both the overlapping elements and the source-specific elements. In other words, someone who queries the mediated schema is able to ask queries about things that exist solely in ifcXML or CityGML, or query overlapping concepts.

All overlapping elements are contained in the mediated schema. Specific source elements are represented in the mediated schema either by having one new concept in the mediated schema (if there is an overlapping concept to represent) or representing both initial source schemas in the resulting schema (if the concepts do not overlap). Example 2 helps to further describe the mediated schema.

Example 2:

Consider the input relation and mapping from Example 1. BCDI's algorithm will create a single relation in the mediated schema that has the name "*BuildingAddress*". The name of the relation is determined by the name of the relation in the head (i.e., on the left hand side) and will have the common attributes as listed by the entries in the left hand side parenthesis:

- *Building_name*
- *Addresslines/House_Number + Street*
- *Postalbox*
- *City*
- *Region*
- *Postalcode*
- *Country*

Additionally, the "*BuildingAddress*" relation in the mediated schema will contain the remaining attributes that appear only in the ifcXML representation:

- *Purpose*
- *Description*

Finally, the "*BuildingAddress*" relation in the mediated schema will contain the remaining attributes that appear only in the CityGML representation:

- *Xal_Source*
- *Multipoint*

Taken together, the resulting "*BuildingAddress*" relation in the mediated schema is:

- *BuildingAddress(Building_name, Addresslines, Postalbox, City, Region, Postalcode, Country, Purpose, Description, Xal_Source, Multipoint)*

The resulting mediated schema satisfies the correctness criteria that are required in Pottinger and Bernstein (2008). These criteria are completeness, preservation of overlapping concepts, and minimality. In this way, we can ensure that not only does the mediated schema represent all the information of the source schemas, but the queries over it can be rewritten to be answered by every single source or combination of all sources. BCDI's algorithm for generating the mediated schema guarantees that all overlapping information can be queried using only one relation and that all information in each of the original CityGML and ifcXML schemas can still be accessed.

7.3 Mappings from Mediated Schema to the Sources

In addition to creating the mediated schema, BCDI extends the methods introduced in Pottinger and Bernstein (2008) to create mappings from the mediated schema to the original ifcXML and CityGML models. In this way, it is possible to translate queries made over the mediated schema into queries over the ifcXML and CityGML data. In particular, two sets of mappings are created in BCDI: one set of global-as-view (GAV) mappings, and another set of local-as-view (LAV) mappings. This combination is known as global-local-as-view (GLAV) mappings. GLAV mappings combine the expressive power of GAV and LAV together for completeness in expressibility.

GAV mappings describe the global source, in this case mediated schema, as views on local source schemas, i.e., ifcXML and CityGML. The LAV mapping defines local sources as views over the global source. BCDI's GLAV mappings consist of a set of GAV mappings from an intermediate schema "*I*" to the ifcXML and CityGML schemas and a set of LAV mappings from "*I*" to the mediated schema. The intermediate schema "*I*" represents the attributes for each mediated schema relation that are available from each source. In BCDI's case, the mediated schema is the union of the attributes from all the source schemas on the same concept; therefore, the intermediate schema for each source schema is the intersection of the mediated schema and the source schema.

BCDI employs an intermediate schema as a helper schema to connect between mediated schema and source schemas. The GLAV mappings are defined by the following rules from mediated schema "*M*" to source schema "*S*":

GLAV Mapping $\mathbf{Map_{M\_S}}$:

LAV:

1. For each completeness relation "*M.R*" in mediated schema "*M*", "Map_{M_S}" includes the following query: *I.R(attr(M.R)) :- M.R(attr(M.R))*
2. For each relation "$I.R_i$" in "*I*" that corresponds to some overlap "*O*", "Map_{M_S}" includes the following query, where "*M.R*" corresponds to "*O*": $I.R_i(attr(I.R_i))$ *:- M.R(attr(M.R))*

GAV:

1. For each completeness relation "*I.R*" in intermediate schema "*I*", "Map_{M_S}" includes the following query: *R(attr(I.R)) :- R(attr(I.R))*
2. For each relation "$I.R_i$" in intermediate schema "*I*" that corresponds to some query "Q_i" in overlap "*O*" in "*O*", "Map_{M_S}" includes the following query: $I.R_i(attr(I.R_i))$ *:- body(Q)*

Example 3 helps clarify the mapping process from the mediated schema to the data sources.

Example 3:

This example continues from Example 2. To reduce confusion, the mediated schema relation is denoted by prepending an "*M.*" to it: *M.BuildingAddress(Building_name, Addresslines, Postalbox, City, Region, Postalcode, Country, Purpose, Description, Xal_Source, Multipoint)*. Similarly, the relations in the intermediate schema "*I*" are prepended with an "*I.*".

The mappings from the mediated schema "*M*" to the intermediate schema "*I*" are:

- *I.BuildingAddress(Building_name, Purpose, Description, Addresslines, Postalbox, City, Region, Postalcode, Country):- M.BuildingAddress(Building_name, Addresslines, Postalbox, City, Region, Postalcode, Country, Purpose, Description, Xal_Source, Multipoint)*
- *I.Address(Building_name, Addresslines, Postalbox, City, Region, Postalcode, Country, Xal_source, Multipoint):-M.BuildingAddress(Building_name, Addresslines, Postalbox, City, Region, Postalcode, Country, Purpose, Description, Xal_Source, Multipoint)*

The mappings from the intermediate schema "*I*" to the ifcXML and CityGML schemas are:

- *IFC.BuildingAddress(Building_name, Purpose, Description, Addresslines, Postalbox, City, Region, Postalcode, Country):-I.BuildingAddress(Building_name, Purpose, Description, Addresslines, Postalbox, City, Region, Postalcode, Country)*
- *CityGML.Address(Building_name, addressLines, House_number, Postalbox, City, Region, Postalcode, Country,Xal_source, Multipoint):-I.Address(Building_name, Street, House_number, Postalbox, City, Region, Postalcode, Country, Xal_source, Multipoint), addressLines = house_number + " " + street.*

BCDI makes a small modification to the GLAV mapping from Pottinger and Bernstein (2008). As shown in Example 3, the value of the "*addressLines*" attribute in the "*IFC.BuildingAddress*" schema is equal to the expression, "*house_number+" "+street*", both of which are in the "*CityGML.Address*" schema. The resulting mapping between the mediated schema and IFC source schema are the same as in Pottinger and Bernstein (2008). By using the arithmetic expressions, BCDI can return richer and more complete information even when information from one schema is missing. BCDI uses arithmetic expressions in the GLAV mappings of every building component and for a building's texture, material, and geometric properties, which justifies the need for the arithmetic expressions.

7.4 Query Rewriting

Once BCDI creates the mediated schema and its mappings, users can query the mediated schema. However, since there no data is stored in the mediated schema, any queries written over the mediated schema must be rewritten as queries over the source schemas (ifcXML and CityGML) where the data is stored.

In BCDI, if the queries are asked over a source schema, no rewriting is needed. When queries are asked over the mediated schema, the rewriting strategy is to first look at all the LAV views that have this mediated schema relation in their body and then find the corresponding GAV views that have the same heads as the resulting LAV views' heads. The rewriting replaces the mediated schema with all of the possible source schemas in the GAV views returned in the previous step. If the query has more than one subgoal in the body of the query, the bodies of the rewritten queries are the result of all combinations of possible source schemas replacing each subgoal in the queries. Finally, the variable names in the source schemas are replaced if they are different from the corresponding names in the queries.

There is one important change in BCDI's algorithm from the algorithm in Pottinger and Bernstein (2008). BCDI must check the variables in all the subgoals of the rewritten queries to see if they contain all the variables in the query's head. If not, BCDI deletes that query. Because the query is posed over the mediated schema, which is the union of variables from all of the source schemas, the source schemas used to replace it may not have the variables in the query's head. Therefore, this source should be deleted from the query because it cannot answer the query. In this way, the BCDI algorithm guarantees that the returned query result can be from any single source data or a union of the results from both sources.

Example 4 helps to clarify the query rewriting process.

Example 4:

Consider a query "*q*" over the mediated schema of "*BuildingAddress*" by using the GLAV mappings from Example 3:

- *q(building_name, addressLines):- M.BuildingAddress(building_name, purpose, description, addressLines,postalBox,city,region,postalCode,country,street,house_number, xal_source, multipoint)*

This query can be translated using the GLAV mappings "*MapM_IFC_BuildingAddress*" and "*MapM_CityGML_ BuildingAddress*" into:

- *q(building_name, addressLines):- IFC.BuildingAddress(building_name, purpose, description,addressLines,postalBox,city,region,postalCode,country)*
- *q(building_name, addressLines):- CityGML.Address(building_name, postalBox, city, region,postalCode,country,street,house_number,xal_source,multipoint), addressLines = house_number +" "+street*

As shown in Example 4, the arithmetic expression "*house_number+" "+street*" is calculated to answer the attribute "*addressLines*" in the CityGML schema, which is not answerable by any single attribute in the CityGML schema. The arithmetic expressions are easy and fast to calculate, and they do not increase the complexity of the query rewriting algorithm because the body of the GAV view replaces the mediated schema regardless of how many arithmetic expressions the body has.

8. CONCLUSION AND OUTLOOK

BIM and GIS are two major information models to describe spatial objects, such as buildings, streets, land, cities, furniture, and transportation. BIM is concerned with building details and is a semantic representation of buildings and their inner components. In contrast, GIS is a 3D city model, which is a digital representation of the earth's surface and its spatial objects. Integrating building details into their broader context is a promising solution to meet the demand for applications that support decision-making processes in the operations and maintenance domain at a neighborhood level.

Taking advantage of the previous integration approaches and the current 3D city visualization and management applications, in this work, we first built a new data transformation approach on top of the BIMserver and FME WorkBench, which appeared to be the top-two most promising available applications. However, in this data transformation approach, the building models can achieve the details of LoD3, but only part of LoD4, as they do not have the inner furniture, stairs, and electrical units, and the transformation approach cannot convert the mechanical components in the buildings. Answering the queries about finding a mechanical system requires adding this part into the current CityGML files. Because the standard CityGML schemas do not cover the mechanical and utility part of BIMs, BCDI extends the standard CityGML schemas and then integrates them.

After comparing data integration systems and data warehouses, the approach used for BCDI was to create an interoperable building model based on data integration architecture. BCDI uses a modified version of the ifcXML and CityGML standard schemas and identifies overlapping information within them. BCDI applies complex arithmetic expressions not only to semantic representations but also to geometric representations of building components. After modifying the mediated schema generation and query rewriting algorithms, BCDI generates a new mediated schema along with the corresponding relationships between the mediated schema and all of the source schemas, with respect to the mappings, using arithmetic expressions.

As some challenges remain regarding BCDI, there is potential for future research endeavors. Since this work provides an “approach” to data integration on the micro and macro levels, the immediate future work would be the operationalization of this approach in a case study, with actual operation and maintenance tasks. Furthermore, as discussed earlier, the standard schema specifications of both ifcXML and CityGML do not carry much information related to facility management; therefore, further enhancing these schemas for more specific FM use could be another subject of future research.

There is some other equipment-related information stored in several individual data repositories, such as manufacturer information, serial number, maintenance history, service manual, or spare parts of the equipment that need to be repaired or replaced. If BCDI can incorporate all of these, the enriched mediated schemas could answer more maintenance related requests.

REFERENCES

ASME, 2018. ASME B31. 1: for Power Piping. American Society of Mechanical Engineers.

Autodesk, 2010. Autodesk LandXplorer Studio Professional. URL: http://download.autodesk.com/us/landxplorer/docs/LDX11_Studio/index.html?topic.htm (accessed 3.10.19).

Bai, Y., 2016. Integrating GIS and BIM for Community Building Energy Design (Master’s Thesis). University of British Columbia. https://doi.org/10.14288/1.0340680

Bai, Y., Zadeh, P.A., Staub-French, S., Pottinger, R., 2017a. Integrating GIS and BIM for Community-Scale Energy Modeling, in: International Conference on Sustainable Infrastructure 2017. https://doi.org/10.1061/9780784481196.017

Bai, Y., Zadeh, P.A., Staub-French, S., Pottinger, R., 2017b. Integrating GIS and BIM for Community-Scale Energy Modeling, in: International Conference on Sustainable Infrastructure 2017. https://doi.org/10.1061/9780784481196.017

Batini, C., Lenzerini, M., Navathe, S.B., 1986. A Comparative Analysis of Methodologies for Database Schema Integration. ACM Comput Surv 18, 323–364. https://doi.org/10.1145/27633.27634

BIMserver, 2016. Open source BIMserver. Open Source BIMserver. URL: http://bimserver.org/ (accessed 12.10.18).

BSI Group, 2009. BSI - BS EN 13941. British Standards Institution (BSI).

buildingSmart Alliance, 2013a. Industry Foundation Classes Release 4 (IFC4). IFC4 Doc. URL: http://www.buildingsmart-tech.org/ifc/IFC4/final/html/ (accessed 12.4.18).

buildingSmart Alliance, 2013b. IfcExplorer CityGML Export - IfcWiki. IfcExplorer CityGML Export - IfcWiki. URL: http://www.ifcwiki.org/index.php/IfcExplorer_CityGML_Export (accessed 12.4.18).

Cavka, H.B., Staub-French, S., Poirier, E.A., 2017. Developing Owner Information Requirements for BIM-enabled Project Delivery and Asset Lifecycle Management. Autom. Constr.

Cavka, H.B., Staub-French, S., Pottinger, R., 2015. Evaluating the Alignment of Organizational and Project Contexts for BIM Adoption: A Case Study of a Large Owner Organization. Buildings 5, 1265–1300.

Codd, E.F., 1970. A Relational Model of Data for Large Shared Data Banks. Commun ACM 13, 377–387.

https://doi.org/10.1145/362384.362685

de Laat, R., van Berlo, L., 2011. Integration of BIM and GIS: The Development of the CityGML GeoBIM Extension, in: Kolbe, T.H., König, G., Nagel, C. (Eds.), Advances in 3D Geo-Information Sciences. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 211–225. https://doi.org/10.1007/978-3-642-12670-3_13

Eastman, C.M., Teicholz, P., Sacks, R., Liston, K., 2011. BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, Second. ed. John Wiley and Sons.

El-Mekawy, M., 2010. Integrating BIM and GIS for 3D City Modelling : The Case of IFC and CityGML, Trita-SOM. KTH, Geoinformatics.

El-Mekawy, M., Östman, A., 2010. Semantic Mapping: an Ontology Engineering Method for Integrating Building Models in IFC and CityGML, in: Digital Earth in the Service of Society: Sharing Information, Building Knowledge. Presented at the 3rd ISDE DIGITAL EARTH SUMMIT, Nessebar, Bulgaria.

El-Mekawy, M., Östman, A., Hijazi, I., 2012. A Unified Building Model for 3D Urban GIS. ISPRS Int. J. Geo-Inf. 1, 120–145. https://doi.org/10.3390/ijgi1020120

Froese, T., Fischer, M., Grobler, F., Ritzenthaler, J., Yu, K., Sutherland, S., Staub, S., Akinci, B., Akbas, R., Koo, B., others, 1999. Industry foundation classes for project management-a trial implementation. Electron. J. Inf. Technol. Constr. 4, 17–36.

Isikdag, U., Zlatanova, S., 2009. Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models, in: Lee, J., Zlatanova, S. (Eds.), 3D Geo-Information Sciences. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 79–96. https://doi.org/10.1007/978-3-540-87395-2_6

Jusuf, S.K., Mousseau, B., Godfroid, G., Hui, V.S.J., 2017. Integrated modeling of CityGML and IFC for city/neighborhood development for urban microclimates analysis. Energy Procedia 122, 145–150. https://doi.org/10.1016/j.egypro.2017.07.329

Kutzner, T., Kolbe, T.H., 2016. Extending Semantic 3D City Models by Supply and Disposal Networks for Analysing the Urban Supply Situation, in: Kersten, T.P. (Ed.), Lösungen Für Eine Welt Im Wandel, Dreiländertagung Der SGPF, DGPF Und OVG, 36. Wissenschaftlich-Technische Jahrestagung Der DGPF, Publikationen Der Deutschen Gesellschaft Für Photogrammetrie, Fernerkundung Und Geoinformation (DGPF) e.V. Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., Bern, pp. 382–394.

Léon van Berlo, 2009. CityGML Extension for BIM/IFC information.

Madhavan, J., Bernstein, P.A., Rahm, E., 2001. Generic Schema Matching with Cupid.

Nagel, C., Stadler, A., Kolbe, T., 2007. Conversion of IFC to CityGML, in: Meeting of the OGC 3DIM Working Group at OGC TC/PC Meeting. Paris, France.

Nagel, C., Stadler, A., Kolbe, T.H., 2009. Conceptual Requirements for the Automatic Reconstruction of Building Information Models from Uninterpreted 3D Models, in: Kolbe, T.H., Zhang, R., Zlatanova, S. (Eds.), Proceedings of the Academic Track of the Geoweb 2009 - 3D Cityscapes Conference in Vancouver,Canada, 27-31 July 2009, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. ISPRS, pp. 46–53.

OGC, 2012a. City Geography Markup Language (CityGML) - Open Geospatial Consortium (OGC). URL: https://www.citygml.org/ (accessed 12.4.18).

OGC, 2012b. IfcExplorer CityGML Export - Open Geospatial Consortium (OGC). URL: http://www.ifcwiki.org/index.php/IfcExplorer_CityGML_Export (accessed 12.4.18).

OGC, 2012c. OGC City Geography Markup Language v 2.0 Adopted - Open Geospatial Consortium (OGC). URL: http://www.opengeospatial.org/pressroom/pressreleases/1599 (accessed 3.10.19).

Ohori, K.A., Krijnen, T., Diakité, A., Ledoux, H., Stoter, J., 2018. GeoBIM project: Final report. Delft University of Technology and Eindhoven University of Technology.

Oracle, 2018. Oracle Spatial and Graph. Oracle Spat. - Doc. URL: https://www.oracle.com/technetwork/database/options/spatialandgraph/documentation/index.html (accessed 3.21.19).

Pottinger, R., Bernstein, P.A., 2008. Schema Merging and Mapping Creation for Relational Sources, in: Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology, EDBT '08. ACM, New York, NY, USA, pp. 73–84. https://doi.org/10.1145/1353343.1353357

Pottinger, R., Levy, A.Y., 2000. A Scalable Algorithm for Answering Queries Using Views, in: VLDB. Presented at the Proceedings of the 26th VLDB Conference, Cairo, Egypt, pp. 484–495.

Safe Software Inc, 2018. FME Desktop Documentation. URL: https://support.safe.com/KnowledgeDocumentation#GroupA (accessed 12.10.18).

Solibri, 2018. Imagine. Reliable information on demand.. Solibri. URL: https://www.solibri.com/ (accessed 3.10.19).

UBC-EWS, 2017a. Hot Water District Energy System - UBC Energy and Water Services. Hot Water Dist. Energy Syst. - UBC Energy Water Serv. URL: http://energy.ubc.ca/ubcs-utility-infrastructure/district-energy-hot-water/ (accessed 12.2.18).

UBC-EWS, 2017b. Campus Energy Centre - UBC Energy and Water Services. Campus Energy Cent. - UBC Energy Water Serv. URL: http://energy.ubc.ca/projects/district-energy/campus-energy-centre/ (accessed 12.2.18).

Ullman, J.D., 1988. Principles of Database and Knowledge-base Systems. Computer Science Press, Incorporated.

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., Kolbe, T.H., 2018. 3DCityDB - a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. Open Geospatial Data Softw. Stand. 3, 5. https://doi.org/10.1186/s40965-018-0046-7

Zadeh, P., Staub-French, S., Pottinger, R., 2015. Review of BIM Quality Assessment Approaches for Facility Management, in: ICSC15: The Canadian Society for Civil Engineering 5th International/11th Construction Specialty Conference. University of British Columbia, Vancouver, Canada, pp. 1887–1896. https://doi.org/10.13140/RG.2.1.2367.7282

Zadeh, P.A., Cavka, H.B., Staub-French, S., 2016. BIM Information Quality Analysis for Space Management. Presented at the 16th International Conference on Computing in Civil and Building Engineering (ICCCBE 2016), Osaka University, Osaka, Japan.

Zadeh, P.A., Staub-French, S., 2016. Assessing the Quality of Building Information Models for Facility Operations.

Zadeh, P.A., Wang, G., Cavka, H.B., Staub-French, S., Pottinger, R., 2017. Information Quality Assessment for Facility Management. Adv. Eng. Inform. 33, 181–205. https://doi.org/10.1016/j.aei.2017.06.003

9. APPENDICES

Since the appendix content is extensive, it is available in an online repository:

9.1 Appendix A: http://bit.ly/BCDI-A

9.2 Appendix B: http://bit.ly/BCDI-B

9.3 Appendix C: http://bit.ly/BCDI-C

9.4 Appendix D: http://bit.ly/BCDI-D

9.5 Appendix D: http://bit.ly/BCDI-E