BG-LoA: A Benchmarking Framework for BIM/GIS Data Integration Based on Meta-Modelling Theory

Junxiang Zhu [1, 2], Ran Wei [1], Mengtian Yin [1], Ioannis Brilakis [1]

[1] Department of Engineering, University of Cambridge, Cambridge, United Kingdom – jz652@cam.ac.uk, rw741@cam.ac.uk, my424@cam.ac.uk, ib340@cam.ac.uk
[2] School of Design and the Built Environment, Curtin University, Bentley, Western Australia – junxiang.zhu@curtin.edu.au

Keywords: BIM, GIS, Industry Foundation Classes, Integration, Benchmarking, Model-Driven Engineering.

Abstract

There are various studies on the integration of building information modelling (BIM) and geographic information system (GIS) that convert BIM data into GIS data. These studies used different technologies, data standards, data models, and data formats such as RDF (Resource Description Framework), graph, CityGML, and Shapefile to address the data incompatibility problem between BIM and GIS. This makes it difficult to compare these studies for the purpose of assessing research innovation, and sometimes, improper comparisons were carried out between studies. To solve this problem, this study adopted the metamodeling theory used in model-driven software engineering (MDSE) and extended its 4-layer framework of level of abstraction. The new benchmarking framework, referred to as level of abstraction for BIM/GIS data integration (BG-LoA), has 5 levels, including meta metamodel (M3), metamodel (M2), model (M1), reality (data) (M0), and reality (real world) (M00). The common types of IFC (Industry Foundation Classes) conversion were analysed using the new framework, including IFC-to-CityGML, IFC-to-CityJSON, IFC-to-Shapefile, IFC-to-GML, IFC-to-3DTiles, IFC-to-RDF, IFC-to-IfcGraph, and IFC-to-UML. The result shows that the new framework can show the difference between those conversion types, which makes it easier for future studies to benchmark their work and assess level of contribution to the body of knowledge.

1. Introduction

1.1 BIM/GIS Integration

The integration of building information modelling (BIM) and geographic information system (GIS) has attracted the attention of researchers from various areas, such as the Architecture, Engineering, and Construction (AEC) domain and the geospatial industry, due to the potential benefits from the fusion of these two technologies (Zhu and Wu, 2022; Chi et al., 2017). By far, the integration of BIM and GIS has contributed to the solution of many problems in the AEC domain and the geospatial industry, such as construction risk management, flood damage simulation and assessment (Amirebrahimi et al., 2016; Rong et al., 2020), offshore platform disassembly (Tan et al., 2018), and supply chain management (Deng et al., 2019).

Behind those applications is an inevitable task of data conversion, due to the data incompatibility between these two systems (Zhu et al., 2018b; Zhu et al., 2018a; Zhu and Wu, 2021a). To use BIM data such as Industry Foundation Classes (IFC) in GIS, the data format needs to be converted, for example, into XML (Extensible Markup Language), JSON (JavaScript Object Notation), Shapefile, or 3D tile.

The eventual goal of data conversion is to enable the full use of BIM information in GIS. Accordingly, various conversion paths have been developed (Zhu et al., 2021). By far, the common types of IFC conversion include IFC-to-CityGML (Deng et al., 2016), IFC-to-CityJSON, IFC-to-Shapefile (Zhu et al., 2019a; Zhu et al., 2019b; Zhu and Wu, 2021b), IFC-to-RDF (Resource Description Framework) (Pauwels et al., 2017; Pauwels and Terkaj, 2016), and IFC-to-Graph (Zhu et al., 2023). A comprehensive review on the data conversion between BIM and GIS has been presented in (Zhu and Wu, 2022).

1.2 Research Problem

While there are so many studies on BIM/GIS data integration and many of these studies appear to be very similar, as they were all about converting BIM data into GIS data, researchers find it difficult to compare and benchmark their studies, and sometimes studies were improperly compared. For example, IFC-to-IfcGraph was compared with IFC-to-RDF in (Zhu et al., 2023) because IfcGraph was thought to be similar to RDF due to their commonality in the use of graph for information visualisation. This comparison is however improper, which will be explained later.

Apart from the above problem, the lack of a benchmarking framework leads to a further problem for future new studies to assess their scientific contribution and clarify their research innovation.

1.3 Research Aim

Therefore, the research question of this paper is how to better benchmark studies in BIM/GIS data integration, and the aim is to propose a framework to address this problem.

The metamodeling theory was adopted in this study to address this problem because this theory, originated in model-driven software engineering (MDSE) or just model-driven engineering (MDE), deals with ‘models’ and ‘modelling’ in software engineering (Brambilla et al., 2017), while most studies in BIM/GIS data integration involves data models or data modelling. Therefore, the assumption of this study is that the metamodeling theory can be used to benchmark studies in BIM/GIS data integration.

1.4 Structure of the Paper

The remainder of this paper is organised as follows. Section 2 introduces the key concepts of MDE, including model, modelling, metamodeling, and the level of abstraction of models. Section 3 presents the new framework for benchmarking studies in BIM/GIS data integration. A case study is presented in Section 4, which applies the new framework to analyse the common types of IFC conversion. Discussions are presented in Section 5, and Section 6 provides the conclusions of this study.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-4-2024-591-2024

2. The Metamodeling Theory in MDE

2.1 Model-Driven Engineering

MDE is a methodology used in software engineering (Brambilla et al., 2017). This methodology elevates 'model' to the first citizen in software engineering, trying to improve the efficiency of software development (Barmpis and Kolovos, 2013).

The most essential concept in MDE is model (Kent, 2002). A model generally refers to an abstraction of a physical thing. In the context of software engineering, it refers to an abstraction of a computer software system of interest. According to the 'form' of a model, there are two types, i.e., conceptual model (or mental model) and explicit model. A conceptual model is a model in human's mind, while an explicit model has been represented by using text or graph.

Models are conceptualised and then developed by using a tool, such as Eclipse Modelling Framework (EMF) (Steinberg et al., 2008), for communication with stakeholders (such as system engineers, domain experts, and clients) during the design of software and are used to automate the generation of codes that can be used in development (Barmpis and Kolovos, 2012).

2.2 Modelling and Metamodeling

The process of conceptualising and creating models are referred to as modelling, which is confined to a standard procedure and defined by abstract syntax, concrete syntax, and semantics (Brambilla et al., 2017). The outcome of modelling is a model.

The model itself can be modelled as well, referred to as 'metamodeling', which leads to another level of abstraction, and the outcome is a metamodel. While a model describes a computer software system, its metamodel describes the model by specifying the key domain concepts, semantics, constraints associated, and relationships (Schmidt, 2006). Similarly, the 'metamodel' can be modelled as well, leading to 'meta metamodel', and so forth.

This process can be indefinite, but it will not be that helpful after several iterations, this is because models beyond 'meta metamodel' are self-descriptive, according to (Brambilla et al., 2017).

2.3 Levels of Abstraction for Software Modelling

Based on the above concepts, a system containing several levels of abstraction (LoA) can be formed (see Figure 1), while each model represents a level of abstraction. This system has four layers including M3, M2, M1, and M0, representing meta metamodel, metamodel, model, and reality, respectively (Brambilla et al., 2017).

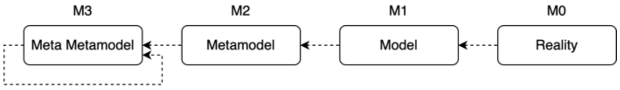


Figure 1. The Levels of Abstraction for software modelling.

The 'reality' has the lowest level of abstraction. In software engineering, the reality (M0) refers to an instance of a computer system (Brambilla et al., 2017). The reality (M0) can be described by a model (M1) at a higher level of abstraction showing the components of the system and their relationships. The model of the system itself (M1) can be described by a metamodel (M2) that presents an even higher level of abstraction, which refers to a modelling language, such as UML (unified modelling language) (Brambilla et al., 2017). The metamodel (M2) can be described by the meta metamodel (M3) that presents the highest level of abstraction in the framework, while M3 is self-describing (see Figure 2).

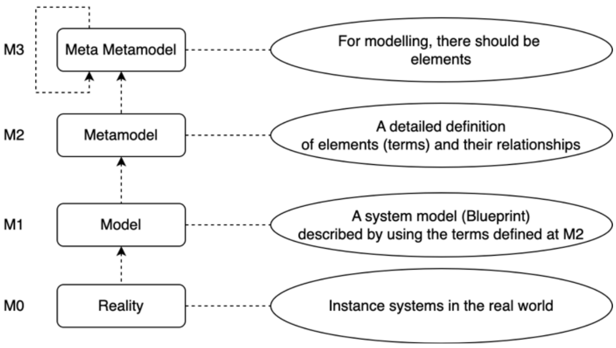


Figure 2. Levels of abstraction of a computer-based system.

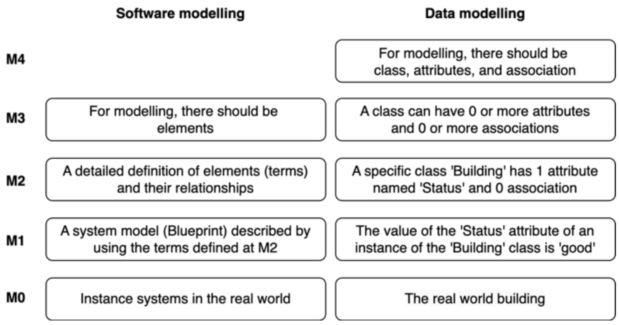
It is worthwhile to note that, meta metamodel and metamodel are 'models' as well and a model at a higher LoA provides useful information for interpreting information in the model at the immediate lower LoA.

3. BG-LoA for Benchmarking BIM/GIS Data Integration

The above framework provides a benchmark for describing the LoA of models in software engineering. However, this framework cannot be directly applied to 'modelling' in BIM/GIS data integration, which also presents levels of abstraction but is different from system modelling.

3.1 Extending LoA for Data Modelling

Models in MDE are for describing software systems (M0), while models in BIM/GIS data integration are for describing data (M0). Therefore, such models are also referred to as data models (Ledoux et al., 2019). Another unique feature of data in BIM and GIS is that the data itself can represent a model of objects in the real world (Antoniou et al., 2012), such as digital elevation model (DEM) for physical terrain and building models for physical built assets. This feature of BIM/GIS data results in another level of abstraction. Therefore, for data modelling, the number of levels of abstraction should be extended to 5 (see Figure 3).



	Software modelling	Data modelling
M4		For modelling, there should be class, attributes, and association
M3	For modelling, there should be elements	A class can have 0 or more attributes and 0 or more associations
M2	A detailed definition of elements (terms) and their relationships	A specific class 'Building' has 1 attribute named 'Status' and 0 association
M1	A system model (Blueprint) described by using the terms defined at M2	The value of the 'Status' attribute of an instance of the 'Building' class is 'good'
M0	Instance systems in the real world	The real world building

Figure 3. Level of Abstraction for data modelling.

However, the introduction of this level of abstraction will cause a serious problem, i.e., there will be semantic mismatch for the interpretation of each LoA. According to (Brambilla et al., 2017), M3 and M2 are about language engineering, while M1 and M0 are about domain engineering. In the new 5-layer framework, since M2 deals with the definition of the specific classes in a domain (e.g., building), it is still about domain engineering. To address this problem, a new interpretation of 'reality' is required.

3.2 LoA for BIM/GIS Data Integration (BG-LoA)

In MDE, ‘reality’ is generally interpreted as ‘real world’, however, in the new framework, to ensure the consistency in the interpretation of these levels of abstraction, ‘reality’ is divided into two sub-layers, i.e., ‘reality (data)’ and ‘reality (real world)’.

Figure 4 presents the new 5-layer framework, which includes meta metamodel (M3), metamodel (M2), model (M1), reality (data) (M0) and reality (real world) (M00). This new framework is referred to as BG-LoA in this study.

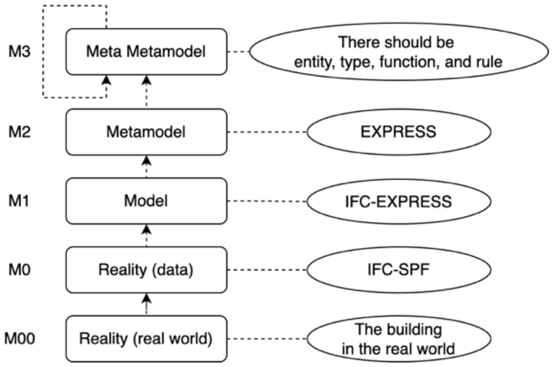


Figure 4. Levels of Abstraction for BIM/GIS data integration (BG-LoA).

An example is also provided in Figure 4. M00 refers to a building in the real world, M0 refers to an IFC-SPF file containing a digital building model of that building, M1 is the IFC standard described by the EXPRESS language, M2 refers to the EXPRESS language, while M3 defines the elements in the EXPRESS language. This new framework not only accommodates the needs of data modelling in BIM/GIS data integration but is also compatible with the original LoA framework defined in MDE.

3.3 Models and Ontology

Ontology is a related concept to model. The relationship between models and ontologies is that models are created to approximate ontologies. The ontology here refers to the ontology in philosophy, which means the nature of existence, or being. It is different from the ontology in computer science, which refers to a list of concepts and categories in a subject area that shows the relationships between them. Ontologies in computer science are described by an ontology language, such as RDF and Web Ontology Language (OWL) (Antoniou et al., 2012). To distinguish these two types of ontology, ontologies in computer science are referred to as digital ontologies if mentioned later. Digital ontologies are data models as well, used to approximate ontologies.

4. Case Study

This section applies the BG-LoA framework to benchmark studies in BIM/GIS data integration. The common types of IFC conversion were analysed, including IFC-to-CityGML, IFC-to-CityJSON, IFC-to-Shapefile, IFC-to-GML, IFC-to-3DTiles, IFC-to-RDF, IFC-to-IfcGraph and IFC-to-UML. For more details of these types of conversion, please refer to (Zhu and Wu, 2022).

4.1 IFC and the Conversion of IFC

4.1.1 Industry Foundation Classes

IFC is an international standard (ISO 16739) approved by ISO (International Organization for Standardization) and managed by buildingSMART. Its purpose is to improve the data interoperability in the AEC domain that uses information from heterogeneous stakeholders in various formats (Sacks et al., 2018). Reading and writing of IFC data is supported by the major BIM authoring tools, such as Autodesk Revit, Tekla, and Bentley. A detailed list of tools supporting IFC can be found in (Buildignsmart International, 2024a).

4.1.2 The IFC Ontology and IFC Data Models

Behind the IFC standard is the IFC ontology (in the sense of philosophy), it has been described by various data modelling languages and data models have been created to approximate this ontology, such as IFC-EXPRESS, IFC-XSD, and ifcOWL (Buildignsmart International, 2024b).

Figure 5 presents these data models (in M1), including IFC-EXPRESS (described by the EXPRESS language), IFC-XSD (described by the XSD language, or XML Schema Definition), and IFC-RDF (described by RDF). There is also an initiative to describe the IFC ontology by using UML (Jetlund et al., 2020).

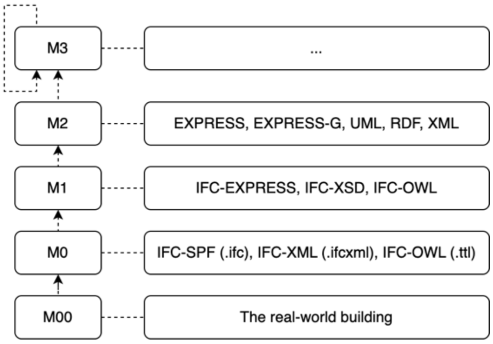


Figure 5. Data models (in M1) to approximate the IFC ontology.

For each of these data modelling languages (EXPRESS, XSD, and RDF), there is a format for data serialisation, including IFC-SPF (.ifc, text-based), IFC-XML (.ifcxml, text-based), and IFC-OWL (.ttl, text-based).

4.1.3 IFC Data Conversion

The conversion of IFC data involves the transformation of data models. To transform models, the definitions of model elements in the higher metamodel need to be referenced. Figure 6 shows a general ‘model’ for model transformation, which shows the transformation of Model A (Ma) to Model B (Mb) via model transformation definition (MT Definition). The MT Definition references Metamodel A (MMa) and Metamodel B (MMb). In BIM/GIS data integration, when IFC data (M0) is converted into another format, it appears to be the change of data format only at M0, but it actually involves the change of data model at M1 (e.g., IFC-EXPRESS to IFC-XML) or even at M2 (e.g., EXPRESS to RDF). The following subsections provide more details of the conversion of IFC using the developed framework.

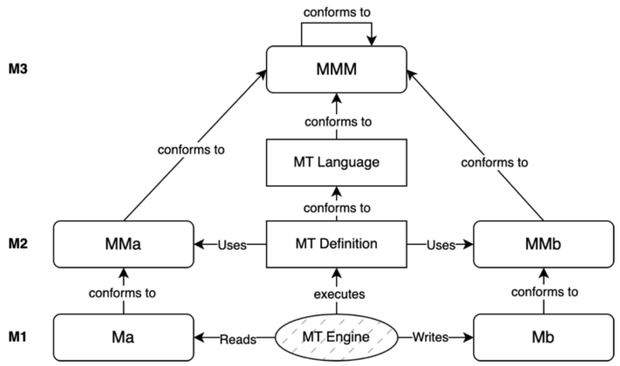


Figure 6. Model conversion from Model A (Ma) to Model B (Mb).

4.2 IFC-to-CityGML and IFC-to-CityJSON

Figure 7 presents the conversion of IFC data into CityGML. The transformation engine needs to read IFC data (e.g., .ifc) and converts the data into CityGML file in the format of GML (Tan et al., 2023). The transformation definitions mainly deal with the mapping of concepts in these two data models, for example, mapping IfcSpace (IFC-EXPRESS) to Room (CityGML). The IFC-to-CityGML conversion is at M0 but needs to reference the data models at M1.

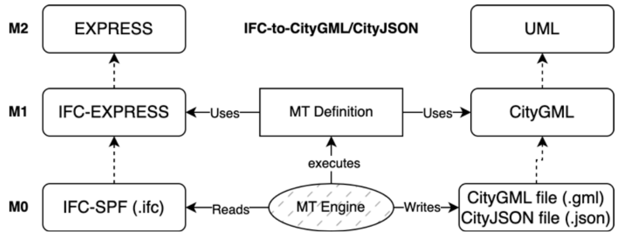


Figure 7. Converting IFC into CityGML/CityJSON.

Since CityJSON uses the CityGML data model (Ledoux et al., 2019), converting IFC into CityJSON follows the same procedure as IFC-to-CityGML, but the final data files are serialised in the JSON format.

4.3 IFC-to-Shapefile, IFC-to-GML, and IFC-to-3DTiles

The IFC-to-Shapefile conversion (see Figure 8) is at M0, as there is no specific semantic data model behind Shapefile. IFC-to-Shapefile mainly deals with geometry conversion, while semantic information can be extracted and stored in tables. As a result, on the GIS side, the transformation engine only needs to reference the Shapefile specification (Esri, 1998) in order to generate correct geometry, instead of a data model. The same procedure applies to GML and 3D Tiles, which do not have a specific data model behind.

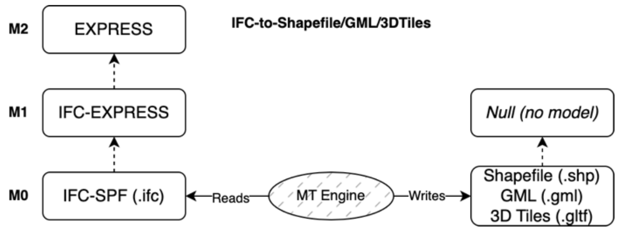


Figure 8. Converting IFC to Shapefile/GML/3DTiles.

4.4 IFC-to-RDF (EXPRESS-to-RDF)

Figure 9 presents the IFC-to-RDF (or more specifically EXPRESS-to-RDF) conversion. The conversion of IFC into RDF is completed in two stages. For the first stage, the transformation engine (MT Engine 1) creates the IfcOWL data model by referencing the EXPRESS and RDF language at M2. EXPRESS is the initial language used for describing the IFC ontology, while RDF is for describing online resources for the construction of semantic web. In the second stage, another transformation engine (MT Engine 2) was created to read IFC data (e.g., IFC-SPF), execute the transformation rules, and generate the new RDF data. Therefore, the IFC-to-RDF conversion involves M2, M1, and M0.

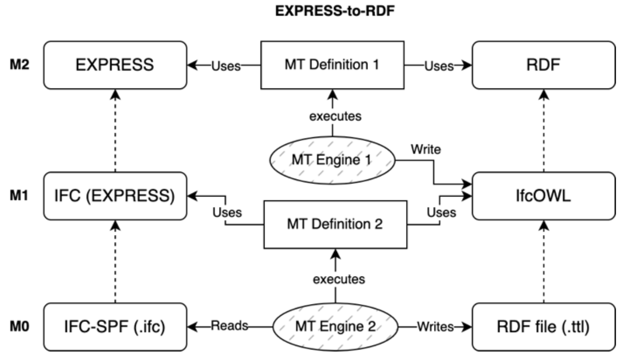


Figure 9. Converting IFC (or EXPRESS) into RDF.

Pauwels et al. have completed a lot of work in this area, they mapped the key concepts of EXPRESS (e.g., Entity and Attribute) onto the key concepts of RDF (e.g., Class and Property) (Pauwels et al., 2011; Pauwels and Terkaj, 2016; Bonduel et al., 2018).

4.5 IFC-to-IfcGraph

The IFC-to-IfcGraph conversion (see Figure 10) aims to convert IFC data into labelled property graphs (LPGs) and store LPGs in graph database, such as neo4j, to support efficient information query from interconnected data (Zhu et al., 2023). This conversion is at M0, just like the IFC-to-Shapefile conversion. This is because there is no specific semantic data model behind graph. The basic ‘node-edge’ structure in graphs does not indicate a semantic data model. Instead, graph is used to implement data models.

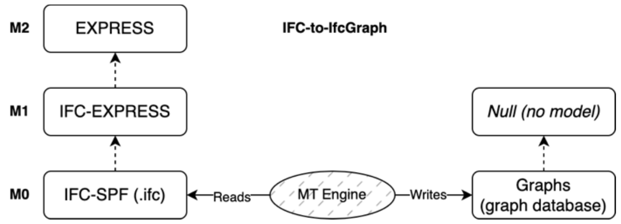


Figure 10. Converting IFC data to IfcGraph.

4.6 IFC-to-UML

The pattern for converting IFC-EXPRESS into IFC-UML is presented in Figure 11. In this conversion, the transformation engine generates the IFC-UML data model by referencing the EXPRESS language and the UML language.

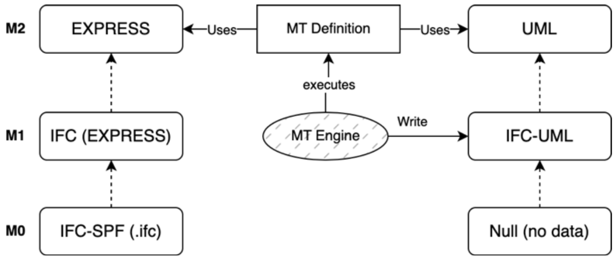


Figure 11. Converting IFC-EXPRESS into IFC-UML.

Literally, this conversion is about redescribing the IFC ontology (described in EXPRESS) by using the UML language. Different from all the above conversions that eventually aim at data conversion, the IFC-to-UML conversion is to support MDE. The established IFC-UML data model can be used to facilitate software development that involves the use of IFC data model.

5. Discussion

Based on the metamodeling theory in MDE, this study developed a 5-layer framework of level of abstraction for benchmarking studies on BIM/GIS data integration.

5.1 Differences between Common Types of IFC Conversion

Table 1 shows a summary of the case study, which applied the new framework to analyse the 8 common types of IFC conversion.

		M0	M1	M2	M3
1	IFC-to-CityGML	X	X		
2	IFC-to-CityJSON	X	X		
3	IFC-to-Shapefile	X			
4	IFC-to-GML	X			
5	IFC-to-3DTile	X			
6	IFC-to-RDF	X	X	X	
7	IFC-to-IfcGraph	X			
8	IFC-to-UML		X	X	

Table 1. The conversion of IFC and the level of abstraction involved.

This study makes it easier to identify the differences between these types of IFC conversion and their purpose. The first 7 types of conversion deal with data conversion from IFC data (e.g., IFC-SPF) into another format or form (e.g., .gml, .shp, and .ttl), while the last type does not.

Among the first 7 types, some types do not involve the change of data model (at M1), such as IFC-to-Shapefile and IFC-to-IfcGraph, while others convert IFC data into a different data model, e.g., IFC-to-CityGML and IFC-to-RDF.

5.2 Patterns in IFC-to-X Conversion

Based on the feature of the destination of the IFC-to-X conversion, studies can be grouped into 4 patterns/categories. (a) Category 1 (C1): converting IFC data and save that into databases, such as IFC-to-IfcGraph. (b) Category 2 (C2): converting IFC data into non-semantic data formats, such as IFC-to-Shapefile, IFC-to-GML, and IFC-to-3DTiles. (c) Category 3 (C3): converting IFC data into semantic data models, such as IFC-to-CityGML and IFC-to-CityJSON. (d) Category 4 (C4): describing IFC data model using another modelling language, such as IFC-to-RDF and IFC-to-UML.

For C1 and C2, they are not confined to specific data models, so they can provide a flexible way to convert IFC data and store them in any data models that would suit the needs of project. For C3, the conversion is confined to specific data models, and there are constraints on the conversion, but the good side is the converted data in the new data model can be well used (such as CityGML in the geospatial industry). For C4, IFC data model is redescribed by a new modelling language. This also involves the change of data model but opens new opportunities for using IFC. For example, IFC-to-RDF can contribute to semantic web, while IFC-to-UML can support model-driven software engineering.

5.3 Conversion Patterns Affecting Data Query

Data models are an abstraction of data, such an abstraction provides useful information for information query from the data. However, that also means when the data model has been changed, the way for information query needs to be changed as well. This applies to studies in C3 and C4 that change data models.

For example, to query IFC-RDF data, users need to learn the structure of the new data model (IFC-RDF) first and create data queries accordingly, as presented in (Zhu et al., 2022; Zhu et al., 2023). In contrast, in the case of IfcGraph, the original IFC data model can be used when querying IfcGraph in graph database (e.g., neo4j), as IfcGraph respects the original IFC data model. This means using C1 and C2 can flatten the learning curve.

5.4 Scientific Contribution of Studies in Different Categories

With this framework, researchers can benchmark their studies and identify their degree of contribution to the knowledge body. Conversions involving a higher level of abstraction usually mean more difficulties to be faced. For example, EXPRESS-to-RDF involves the transformation of both data model and data.

However, it is important to note that work carried out at a lower abstraction level does not necessarily mean less contribution to the body of knowledge. A new form of information may provide new ways for information query and knowledge discovery. For example, the IFC-to-IfcGraph conversion is at M0, but the new graphical form of IFC data provides a new possibility of identifying hidden information in asset data, such as the various types of relationships in IFC.

5.5 Benchmarking Studies in BIM/GIS Data Integration

Studies in the same category can be compared to each other, for example, IFC-to-Shapefile, IFC-to-GML, and IFC-to-Shapefile, while studies within different categories should not be compared with each other, such as IFC-to-RDF and IFC-to-IfcGraph.

5.6 Future Work

This paper is an initial attempt to establish a benchmarking framework for BIM/GIS data integration, further investigation should be carried out by considering data granularity, spatial and temporal dynamics, and interoperability requirements that can significantly impact the conversion process. Further, a deeper exploration of the implications of each conversion pattern on data interoperability and system performance, as well as the practical difficulties in implementing conversion and integration patterns, should be carried out.

6. Conclusion

This study proposes a framework to benchmark studies in BIM/GIS data integration. The established framework, referred to as BG-LoA in this paper, has 5 levels of abstraction (i.e., M3, M2, M1, M0, and M00), where the ‘reality’ (M0) in the original 4-layer framework used in MDE was extended into ‘reality (data)’ (M0) and ‘reality (real-world) (M00) to ensure consistency in the interpretation of each level of abstraction. The common types of IFC conversion were analysed using the new framework, including IFC-to-CityGML, IFC-to-CityJSON, IFC-to-Shapefile, IFC-to-GML, IFC-to-3DTiles, IFC-to-RDF, IFC-to-IfcGraph, and IFC-to-UML.

The main findings of this study are as follows.

(a) The result shows that this framework can effectively distinguish studies by using the concept of level of abstraction. IFC-to-CityGML/CityJSON and IFC-to-RDF can reach M2 or M3, while IFC-to-Shapefile/GML/3DTiles and IFC-to-Graph are mainly at M0.

(b) Among the common types of IFC conversion, IFC-to-CityGML, IFC-to-CityJSON, IFC-to-Shapefile, IFC-to-GML, IFC-to-3DTiles, IFC-to-RDF, and IFC-to-IfcGraph are mainly for data conversion, i.e., from IFC data to data in another format that can be used in GIS, while IFC-to-UML is mainly for creating reusable data models to support software development.

(c) These conversion types can be further grouped into 4 categories depending on the feature of the destination of the conversion, i.e., if there are semantic data models involved or if there are new data modelling languages involved. Studies involving higher level of abstraction usually face more challenges and a higher degree of difficulty, and the change in data model also means the change of ways for data query.

(d) Studies in different categories should not be compared to each other. Therefore, it is not appropriate to compare IFC-to-CityGML with IFC-to-Shapefile/GML/3DTiles or comparing IFC-to-IfcGraph with IFC-to-RDF.

This study makes it easier to benchmark studies in BIM/GIS data integration in a qualitative way and assess their contribution to the body of knowledge. Even though this framework was initially developed for BIM/GIS data integration, it can be applied to any other areas that involve data modelling.

Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101034337. We would like to thank Dr. Lavindra De Silva, Dr. Pieter Pauwels, Dr. Stefano Cavazzi, Yogesh Patel, Matt Peck, Katrin Johannesdottir, George Economides, Ajay Gupta, Peter Lindgren, and Alex Gillies for their contributions to this paper.

References

Amirebrahimi, S., Rajabifard, A., Mendis, P., and Ngo, T., 2016. A framework for a microscale flood damage assessment and visualization for a building using BIM–GIS integration, *International Journal of Digital Earth*, 9, 363-386, https://doi.org/10.1080/17538947.2015.1034201.

Antoniou, G., Groth, P., Van Harmelen, F., and Hoekstra, R., 2012: A semantic web primer, Third, MIT press, United States of America

Barmpis, K. and Kolovos, D., 2013. Hawk: Towards a scalable model indexing architecture, *Proceedings of the workshop on scalability in model driven engineering*, 1-9.

Barmpis, K. and Kolovos, D. S., 2012. Comparative analysis of data persistence technologies for large-scale models, *Proceedings of the 2012 Extreme Modeling Workshop*, 33-38.

Bonduel, M., Oraskari, J., Pauwels, P., Vergauwen, M., and Klein, R., 2018. The IFC to linked building data converter: current status, *Linked Data in Architecture and Construction Workshop*, London, United Kingdom, 19-21 June, 34-43, http://ceur-ws.org/Vol-2159/04paper.pdf.

Brambilla, M., Cabot, J., and Wimmer, M., 2017: Model-driven software engineering in practice, Morgan & Claypool Publishers

buildignSMART International, 2024a. Software Implementations, https://technical.buildingsmart.org/resources/software-implementations/ (24 March 2024).

buildignSMART International, 2024b. IFC Specifications Database, https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ (25 March 2024).

Chi, H.-L., Chai, J., Wu, C., Zhu, J., Wang, X., and Liu, C., 2017. Scaffolding progress monitoring of LNG plant maintenance project using BIM and image processing technologies, *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)*, 1-6.

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

Deng, Y., Gan, V. J., Das, M., Cheng, J. C., and Anumba, C., 2019. Integrating 4D BIM and GIS for construction supply chain management, *Journal of Construction Engineering and Management*, 145, 04019016, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001633.

ESRI, 1998. ESRI Shapefile Technical Description, https://support.esri.com/en/white-paper/279 (24 March 2024).

Jetlund, K., Onstein, E., and Huang, L., 2020. IFC Schemas in ISO/TC 211 Compliant UML for Improved Interoperability between BIM and GIS, *ISPRS International Journal of Geo-Information*, 9, 278.

Kent, S., 2002. Model Driven Engineering, Berlin, Heidelberg, 286-298.

Ledoux, H., Arroyo Ohori, K., Kumar, K., Dukai, B., Labetski, A., and Vitalis, S., 2019. CityJSON: A compact and easy-to-use encoding of the CityGML data model, *Open Geospatial Data, Software and Standards*, 4, 1-12.

Pauwels, P. and Terkaj, W., 2016. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology, *Automation in Construction*, 63, 100-133, https://doi.org/10.1016/j.autcon.2015.12.003.

Pauwels, P., Krijnen, T., Terkaj, W., and Beetz, J., 2017. Enhancing the ifcOWL ontology with an alternative representation for geometric data, *Automation in Construction*, 80, 77-94, https://doi.org/10.1016/j.autcon.2017.03.001.

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van de Walle, R., and Van Campenhout, J., 2011. A semantic rule checking environment for building performance checking, *Automation in Construction*, 20, 506-518, https://doi.org/10.1016/j.autcon.2010.11.017.

Rong, Y., Zhang, T., Zheng, Y., Hu, C., Peng, L., and Feng, P., 2020. Three-dimensional urban flood inundation simulation based on digital aerial photogrammetry, *Journal of Hydrology*, 584, 124308, https://doi.org/10.1016/j.jhydrol.2019.124308.

Sacks, R., Eastma, C., Lee, G., and Teicholz, P., 2018: BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors, Third, John Wiley & Sons, Hoboken

Schmidt, D. C., 2006. Model-driven engineering, *Computer-IEEE Computer Society*, 39, 25.

Steinberg, D., Budinsky, F., Merks, E., and Paternostro, M., 2008: EMF: eclipse modeling framework, Pearson Education

Tan, Y., Liang, Y., and Zhu, J., 2023. CityGML in the Integration of BIM and the GIS: Challenges and Opportunities, *Buildings*, 13, 1758.

Tan, Y., Song, Y., Zhu, J., Long, Q., Wang, X., and Cheng, J. C. P., 2018. Optimizing lift operations and vessel transport schedules for disassembly of multiple offshore platforms using BIM and GIS, *Automation in Construction*, 94, 328-339, https://doi.org/10.1016/j.autcon.2018.07.012.

Zhu, J. and Wu, P., 2021a. A Common Approach to Geo-Referencing Building Models in Industry Foundation Classes for BIM/GIS Integration, *ISPRS International Journal of Geo-Information*, 10, 362, https://doi.org/10.3390/ijgi10060362.

Zhu, J. and Wu, P., 2021b. Towards Effective BIM/GIS Data Integration for Smart City by Integrating Computer Graphics Technique, *Remote Sensing*, 13, 1889, https://doi.org/10.3390/rs13101889.

Zhu, J. and Wu, P., 2022. BIM/GIS data integration from the perspective of information flow, *Automation in Construction*, 136, 104166, https://doi.org/10.1016/j.autcon.2022.104166.

Zhu, J., Wang, P., and Wang, X., 2018a. An Assessment of Paths for Transforming IFC to Shapefile for Integration of BIM and GIS, *2018 26th International Conference on Geoinformatics*, Kunming, China, 1-5, https://doi.org/10.1109/GEOINFORMATICS.2018.8557099.

Zhu, J., Wu, P., and Anumba, C., 2021. A Semantics-Based Approach for Simplifying IFC Building Models to Facilitate the Use of BIM Models in GIS, *Remote Sensing*, 13, 4727, https://doi.org/10.3390/rs13224727.

Zhu, J., Wu, P., and Lei, X., 2023. IFC-graph for facilitating building information access and query, *Automation in Construction*, 148, 104778, https://doi.org/10.1016/j.autcon.2023.104778.

Zhu, J., Wright, G., Wang, J., and Wang, X., 2018b. A Critical Review on the Integration of Geographic Information System and Building Information Modelling at the Data Level, *ISPRS International Journal of Geo-Information*, 7, 66, https://doi.org/10.3390/ijgi7020066.

Zhu, J., Wang, X., Chen, M., Wu, P., and Kim, M. J., 2019a. Integration of BIM and GIS: IFC geometry transformation to shapefile using enhanced open-source approach, *Automation in Construction*, 106, https://doi.org/10.1016/j.autcon.2019.102859.

Zhu, J., Wang, X., Wang, P., Wu, Z., and Kim, M. J., 2019b. Integration of BIM and GIS: Geometry from IFC to shapefile using open-source technology, *Automation in Construction*, 102, 105-119, https://doi.org/10.1016/j.autcon.2019.02.014.

Zhu, J., Chong, H.-Y., Zhao, H., Wu, J., Tan, Y., and Xu, H., 2022. The Application of Graph in BIM/GIS Integration, *Buildings*, 12, 2162, https://doi.org/10.3390/buildings12122162.