Multiple schema integration trough a common intermediate model: a floorplan extraction case study

Helga Tauscher, (helga.tauscher@htw-dresden.de)
Hochschule für Technik und Wirtschaft Dresden - University of Applied Sciences

Keywords: IFC, CityGML, IndoorGML, OpenStreetMap, multi-schema integration, BIM-GIS integration, UML

Abstract

Many works on the integration of the GIS and BIM domains consider a particular pair of schemas, predominantly the CityGML for city models and IFC for building models. However, there is not just one pertinent standard in each domain. Methods appropriate for a pair of schemas may fail or become inefficient when applied to more than two schemas. In this paper, we present a study on a use case driven approach to integrating multiple schemas, namely IFC, CityGML, IndoorGML and OpenStreetMap for converting building data into different formats for city maps and navigation models to be complemented with indoor information. We facilitate the conversion via a mediating step and describe a workflow to identify pieces in the target models to formulate an intermediate model and populate this model from the source model. The implementation has been tested with a number of buildings and we show the results.

1. Introduction

The integration of the GIS and BIM domains attracts unbroken attention in research, because using data from the two worlds in conjunction promises better use of the data, more comprehensive insight and holistic analysis. With the integration of the two domains' data, new use cases become feasible, which are not within the realms of the single domains, for example the life-cycle assessment of a building with its details and in its context (e.g. Tauscher & Wong, 2022). Researchers have previously studied various operational modes of integration, from conversion, linking, to interconnected retrieval or generic integration method. Besides a particular integration method, most of the works are also dedicated to a particular pair of schemas, predominantly and most notably the pair of buildingSMART's IFC for digital building models and OGC's CityGML for 3D city models.

Many works highlight the differences between the BIM and GIS domains and contrast them by example of these most prominent schemas of each domain. However, despite standardization efforts, in reality the data and information modelling landscapes of the domains in themselves are not as homogeneous as they appear in the light of such study. There are subdomains within each the BIM and GIS domain with different views and representations of the built environment as well as varying approaches to modelling, ecosystems of tooling for data acquisition and management - and therefor also more than one pertinent standard in each domain. Thus, integrating pairs of schemas is not sufficient, we must extend the horizon by looking at how to integrate more than two schemas.

In this paper we present a study on a use case driven approach to integrating multiple schemas, namely IFC, CityGML, IndoorGML and OpenStreetMap. Instead of a generic integration, at this point, we look only at a specific operational mode of integration which is conversion from building models into data sets for city maps and navigation services. This is guided by the practical aim of a feasibility study carried out recently: how to enhance and extend city maps and navigation models with indoor information originating from planning processes.

Simply lumping together isolated pairwise considerations can certainly cover the ground of the various facets of both domains, but not assure consistency overall. It seems likely that compared to pairwise model integration, with multiple models, complexity increases substantially and even subtle inconsistencies between two models potentiate to significant impact. The study was conducted as an experiment to get a first heuristic understanding of the increase in complexity, to identify and try-out different viable methods of forming sub sets or super sets of the models, or complementary structures to the models. These identified model pieces are then to assembled in an intermediate model in order to facilitate the conversion via a mediating step.

The paper is structured as follows: We first summarize the state of the art and related research in Section 2. Then we lay out our developments in a reverse deconstruction from the target models via the intermediate model to the source models. This reasoning is "reversed" as opposed to data flow in the envisaged conversion application: From the target models we deduct a suitable intermediate model (Section 3). Then we describe the identified mediating model elements, demonstrate the details of the intermediate model and how it bridges the partially contrasting modelling paradigms of IFC and the target formats in Section 4. With the implementation of the conversion from the digital building model to the intermediate model we can simultaneously identify model requirements as shown in Section 5. The proof-of-concept implementation has been tested with a set of sample data and we show the results in Section 6. Finally, we discuss the limitations of the study, challenges and opportunities to develop a more formal approach to multiple schema integration which also takes into account the different operational integration modes beyond conversion in Section 7.

2. Related work and state of the art

Transformations between different representations of the same or similar domain entities are a common matter in software engineering. Information needs to be cast into different data structures suitable for particular tasks. Subsequently methods have been developed to handle model transformation between two data structures or schemas, which have also been adopted in the AEC domain. For example, Tauscher (2020) suggested to apply a generic approach to express relations between two models which can then serve for various operational data integration modes such as transformation, synchronization, linking etc. In this approach, the relation between two models is expressed with a link graph complementing the graphs of the two domain models. This bears some commonalities to the independent link model proposed by Fuchs and Scherer (2017). While the semantics of domain models are subject to harmonization to achieve interoperability, the semantics of such link models are rarely discussed. Fuchs and Scherer (2017) take the stance that link models require defined semantics and Beck et al. (2020) also caution against the assumption that such links would represent a semantic 1:1 matching and argue that this is not possible. However, in this paper we are not covering the question of link semantics, but assume a well-defined semantic.

There is a large body of work on unidirectional conversion between digital building models and 3D city models, in particular for the conversion from IFC to CityGML, for example the work of de Laat and van Berlo (2011), Deng et al. (2016), Donkers et al. (2016)

and in particular the IFC2CityGML project[1], which implemented a flexible conversion method and a rule-based mapping from IFC to CityGML (Tauscher et al., 2021) and for the first time also studied floorplans in CityGML (Konde et al., 2018). Few authors have also attempted to convert from IFC to IndoorGML (e.g. Diakite et al., 2022).
With its reformulated LoD (Level of Detail) concept, the last Version of CityGML (Kolbe et al., 2021) allows for representation of per-level indoor information in 2.5D. IndoorGML (Lee et al., 2020) complements CityGML, particularly with regard to navigation applications, because it holds explicit topological information that can be used to derive routing graphs with little effort. In OpenStreetMap, there are as well possibilities to represent respective geometries and their semantics, for example with Simple Indoor Tagging (SIT)[2].
We resort to the latest versions of the three formats at the beginning of this study: IFC4 ADD2 TC (4.0.2.1), as standardized in ISO 16739-1:2018 (ISO 16739-1:2018, 2018), short IFC4, CityGML 3.0 (Open Geospatial Consortium, OGC-approved) and IndoorGML 1.1 (OGC-approved). Further we used various open-source libraries, such as the Opensource BIMserver[3] version TODO, further CityGML4J, OSM4J and the highsource implementation of JAXB bindings for OGC standards (IndoorGML).

3. Identification of the intermediate model and building the target models

At the end of the conversion process, we want to actually create data sets in three different target formats. We'll ultimately have similar facts in each of these generated model. We work backward from this goal and extract requirements from this target models, hence define subsets of the models that we want to populate from the IFC input. We than merge and condense these target model subsets into an intermediate model. These target models are very similar, although do not match 1:1 semantically. For example one model contains explicit connections and accessibility between rooms (IndoorGML), another in implicit form (OSM). One model requires geospatial coordinates (OSM) whereas another can handle engineering coordinate systems (CityGML). Overall the conversions might require more or less complex adjustments for the semantic differences. The intermediate model is supposed to level these differences out and containing just enough information to ensure simple and consistent transformation into the three target formats.
The requirements for information to arrive in the target models have been defined as follows: We want rooms and traversable openings with their 2.5D outlines as well as the horizontal connections (doors etc.). They should be aggregated in storeys which might bear the 0.5D dimension leaving rooms and openings at 2D. Other building or constructive elements like walls should have no explicit representation for now. We adhere to a thick-walled model, where the lines in the 2D space represent bounding surfaces.
To develop the intermediate model and the conversion to target models in parallel, we take an inductive approach, a programmatic code-centered method that uses stepwise refactorings as follows:

- **Step 1**: We create predefined sample models in the three target formats programmatically and with hard-coded parameters to the various elements, including coordinates.

[1]IFC2CityGML: https://ifc2citygml.github.io. Strict and automatic mapping of IFC-BIM models into semantically enriched 3D CityGML building models (exterior and interior) @ National University of Singapore.

[2]Simple Indoor Tagging (SIT): https://wiki.openstreetmap.org/wiki/Simple_Indoor_Tagging

[3]Opensource BIMServer: https://github.com/opensourcebim/bimserver

- **Step 2**: We structure the code in respective methods such that these create meaningful entities each and identify suitable method parameters. The methods are called with the hard-coded parameters from step 1.
- **Step 3**: The builder method parameters become intermediate model attributes and are bundled in classes according to their joint appearance in method parameters. The methods are then called with objects instantiated from these classes.
- **Step 4**: The classes and attributes originating in the three target models are harmonized to constitute the intermediate model. Now, the intermediate model is instantiated with hard-coded values and passed to the respective methods.

As a result, we obtain a set of classes constituting the intermediate model and three builder classes — one for each target format — which assemble and persist the target models from an intermediate model instance that they traverse. The code can be found in packages *org.opensourcebim.builders* and *org.opensourcebim.intermediatemodel* in the repository https://github.com/bauinformatik/levelout. To be executed in the context of the Opensource BIMserver, the builders are instantiated and triggered in so-called serializers. Serializers are extension points defined in the Opensource BIMserver for development and integration of additional functionality, particularly the export of IFC data in different formats.
In the following, we informally describe the harmonization carried out in step 4. It is based on "overlaps" in the models, where an overlap between two models is made up of pairs of equivalent elements which have an equivalent element in both models. With object-oriented domain models, elements would be objects with their attributes and relations. Instead of elements, one could also define equivalence on the level of groups of elements. The notion of equivalence is based on a bijection — a particular case of an equivalence relation, that is of a mathematical relation which is transitive, symmetric and reflexive. The equivalence classes in the bijective relation must not necessary reflect semantically equivalent objects in the sense of them denoting the identical identifiable entities in the real world. As Beck et al. (2020) have pointed out, such a relation is hard to define concisely. The equivalence classes may also stand for more or less corresponding elements in the domain models.

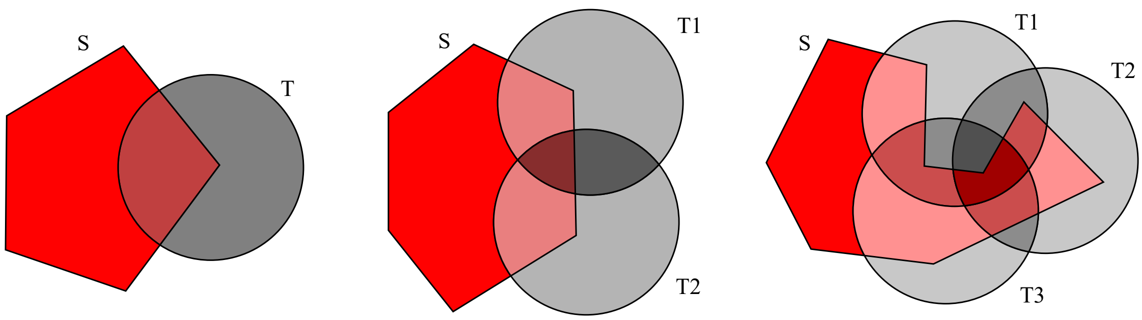


Figure 1: Overlap of source and target models for one (left), two (center), and three (right) target models. The source model is shown in red, the target models in grey tones, and the resulting the intermediate model in lighter and less saturated red tones.

Figure 1 shows, simplified, how to identify the elements of the intermediate model using a notation inspired by Venn diagrams. Note, that in original Venn diagrams, areas represent sets and their overlaps represent common subsets of the sets represented by the overlapping shapes. There are two aspects which are fundamentally different: First, we do not have elements common to multiple models, but equivalence classes of elements constituting the overlaps. Second, our models are not just sets, but object graph structures, hence we would have to operate with category theory and isomorphisms instead of bijections. Such algebraic considerations go well beyond the scope of this paper and are left for future studies. Simplifying to sets with elements appearing in multiple sets [4], we can describe the contents of the intermediate model for n target models $T_1, \ldots, T_n$ and one source model S as given in the following Equation 1 with the example of three target models ($n = 3$) spelled out in Equation 2.

$$S \cap \bigcup_{i=1}^{n} T_i \tag{1}$$

$$S \cap (T_1 \cup T_2 \cup T_3) \tag{2}$$

Hence, we want to have correspondents to all relevant elements of the target models in the intermediate model regardless of whether they have equivalents in other target models themselves or not. If an intermediate model element has correspondents in multiple target models, we expect correspondence between the elements of the different target models as well. Some intermediate model elements are only used in one target model, others appear in two or all models. For example, the floor numbering is mainly an OSM requirement, whereas storeys as such with their respective height above or below the local project's zero point appears explicitly only in CityGML.

4. Intermediate model

In this section, we present the resulting intermediate model from the workflow and considerations in Section 3.

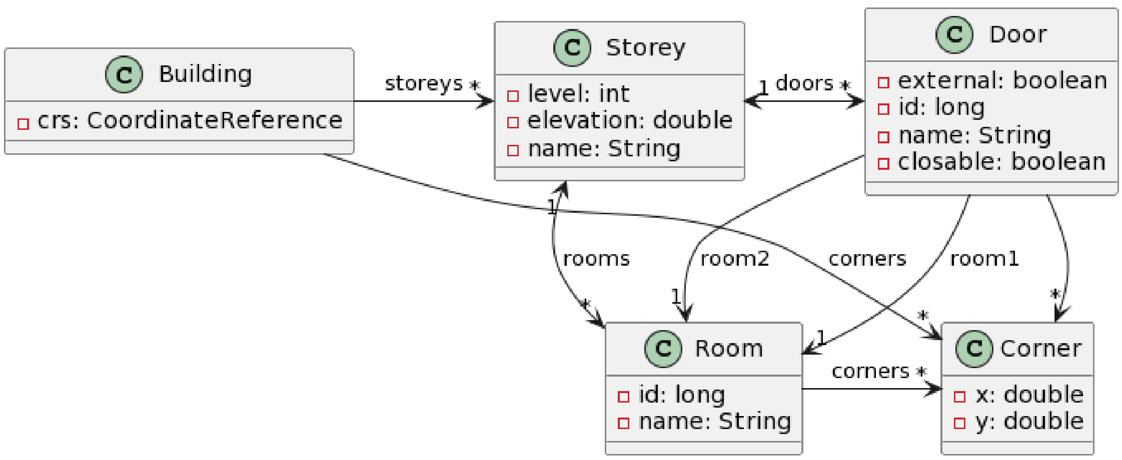


Figure 2: Core intermediate model notated in UML, TODO: label "corners" missing on rightmost association

Figure 2 shows the core intermediate model notated in UML. A preliminary version without the doors has already been published in Krishnakumar et al. (2023). There are few more elements for georeferencing which are left out here for the sake of compactness. The main classes "Building", "Storey", "Room" and "Door" are related such that a "Building" instance aggregates multiple "Storey" instances, which in turn contain "Door" and "Room" instances, expressed with bidirectional associations to cater for target formats

[4] This could, maybe, be an adequate representation of the e.g. real world correspondents to model elements, instead of the model elements themselves.

with implicit storeys [5]. "Door" and "Room" classes are connected via associations such that they represent the accessibility graph. In this graph, rooms are represented as nodes and doors as edges connecting the rooms. Consequently, the "Door" class has associations to two rooms. In this case, the opposite direction is not needed, contrary to typical graph data structures which are optimized for efficient traversal of large graphs across multiple nodes and edges.

As per requirement, geometries are only 2D and faceted (not curved) with the "Corner" class as smallest unit. The naming might appear inexpertly ignoring terms from geometric and topological modelling such as "point" and "vertex". This has been decided on purpose though, since we decided against separation of geometry and semantics. Instead, classes are named after the semantic concepts and equally carry geometry and non-geometric properties. While this seems to violate most established modelling paradigms, we found it appropriate. With this model we follow Occam's razor, keeping our model as simple as possible; we use a consistent modelling approach throughout; and finally we account for the fact that the target model formats take different routes on privileging geometry or semantics. While OSM treats geometry as first-class element and subordinates semantics, CityGML and IndoorGML take the opposite approach.

To test and develop the conversion into the target formats as described in the previous Section 3, some predefined buildings have been instantiated in terms of the intermediate model for testing purposes. This way, work on the conversion from intermediate and to the target models could be executed independently from the intermediate model population described in the next Section 5.

5. Populating the intermediate model

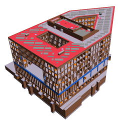
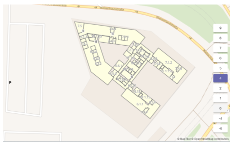
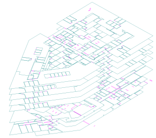
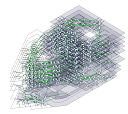
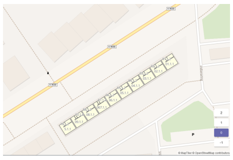
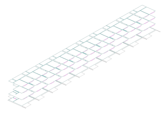
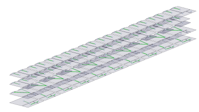
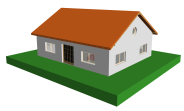
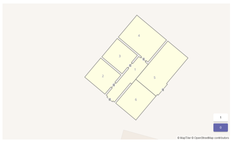
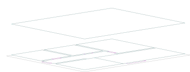
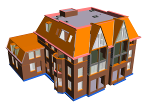
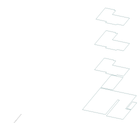
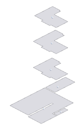
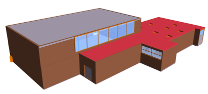
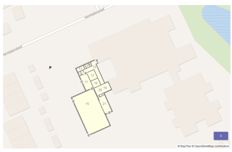
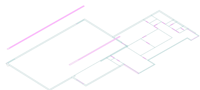
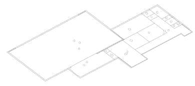
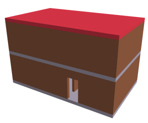
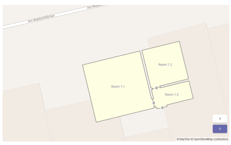
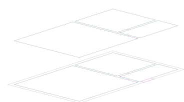
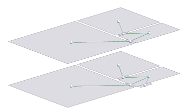
With the intermediate model in place, for every conversion, we first instantiate a respective object network from the IFC data. Thus, for every element (class, association and attribute) of the intermediate model, we have to determine the respective entities to extract from the source model in IFC format based on the IFC schema. The respective mapping is currently hard-coded and can be found in the init method of the class AbstractLevelOutSerializer in package org.opensourcebim.levelout.serializer. The specific serializers for the target formats subsequently use the instantiated intermediate model with the respective builders.

Basically, the mapping process extracts all storeys (currently assuming there is only one building), identifies the groundfloor, takes the building outline from the groundfloor, iterates over the storeys numbering them from 0 for the ground floor upwards and downwards, extracts the rooms and openings per storey with their projected and processed geometries as well as their topological relationships. The georeferencing information for the building is also populated. There are configuration options for the mapping process which are considered during the mapping process. Particularly, certain entities can be ignored and not included in the intermediate model during the mapping process. The two ignore options currently affect rooms or doors without geometry as well as dead rooms without connecting doors.

As a side effect to carrying out the mapping, the source code simultaneously documents the requirements for the input model and thus allows to extract procedures for checking potential input data for its suitability to these particular conversions. By examining the control statements of the program, the respective conditions can be derived. For example, a condition of a branching statement with only one branch (if) might directly indicate requirements. A for-loop requires a non-empty set for the code inside to be

[5] An alternative approach would keep the current storey in the context of the builder and eliminate the need to access the storey during conversion of the doors and rooms.

Table 1: Screenshots of sample buildings with source and target models

	IFC	OSM	CityGML	IndoorGML
A				
B				
C				InViewer issue
D				
E				
F				

executed at all and the respective intermediate model elements to be created. This way, one could find and warn, for example, that in project D (see Section **??**, only the rooms of the vertical access system (staircase etc.) comes with the required geometries.
Similar to the target model builder code, the intermediate model population code — currently in one large method with few helper methods — could be split in to smaller methods. This would reduce control structure depth (nested for and if branches) and also constitute a first step towards a rule-based expression of the mapping.

6. Results

The conversion into the three target formats via the intermediate model has been tested with a series of sample projects of different size and origin as shown in the following list and in Table 1.

- A Public administration building, unpublished
- B Smiley West, KIT sample file
- C FZK house, KIT sample file
- D Schependomlaan, buildingSMART sample file
- E Gymzaal, buildingSMART sample file
- F Two storey residential building, custom sample file

With the exception of project A, the data is published (in source and target formats) at https://mobilithek.info/offers/620975229629235200. For the original IFC files from other sources (projects B–E), the references are provide. Some files have been enhanced with geo-references and are converted to IFC4 with BIMserver (version 1.5.185 or higher). The converted files in target formats OSM-SIT, CityGML, and IndoorGML have been published from conversion with the BIMserver LevelOut plugins version 0.20.

7. Discussion and outlook

With this study, we have shown how an intermediate model can be used to simplify conversion with different but similar target models from the same domain. The practical workflow to derive such intermediate model can serve as a blue-print for other conversions. By decoupling a source-model-oriented conversion part from a target-model-oriented part, implementation work can be carried out from both ends and by different domain specialists who synchronize through discussion about the intermediate model as the common ground. Further, it breaks the larger problem into smaller, more manageable ones — the target model creation should be almost trivial. It also helps to separate concerns and draw module boundaries — the model checking can be confined to the source-model-oriented part. We have also touched upon how checking rules might be derived during instance model creation.

Yet, for this preliminary study, the intermediate model was very restricted due to the low-level and shortened requirements for information to include in the target model. With more requirements (e.g. vertical access, complementary thin-walled model) and removal of sensible, but not always applicable simplifications (e.g. one building per project), the intermediate model will become more complex and certain decisions might be revised (e.g. geometry and semantics separation).

A rule-based version of the two-step conversion could then help to manage the growing number of conversion pieces and Builder code is already structured in small methods (mostly below 10 lines of code) with parameters to receive intermediate model elements and already built target model elements as input. These would be rather straight forward to convert to graph transformation rules similar to the rules in Tauscher et al. (2021). Once the intermediate model population code (from IFC during serializer initialization) would be refactored in a similar fashion, rule derivation would be equally straight forward. These rules could then constitute four different rule sets: one for integrating the building model in IFC with the intermediate model, as well as one each for integrating the intermediate model with the three target models. It would have to be questioned where the rule-based approach is useful and necessary and whether some of the mappings are equally or better manageable with the traditional method of directed traversal.

Most interesting would be the uplift towards a formal approach with algebraic methods, potentially combined with the rule-based approach, such that certain parts of the analysis and implementation could be automated .

Acknowledgements

Financial support of the German Federal Ministry of Digital and Transport (BMDV) through grant 19F1095A is greatfully acknowledged. I would like to mention the work done by project members Subhashini Krishnakumar and Dominik Heigener at Bauhaus-Universität Weimar. The outcome of this paper has also been supported by the City of Dresden by providing the demonstration object *Public Administration*, which is gratefully acknowledged. We also like to thank the contributors of the buildingSMART sample files, Schependomlaan Nijmegen and Gymzaal Amersfoort.

References

Beck, S. F., Borrmann, A., & Kolbe, T. H. (2020). The need for a differentiation between heterogeneous information integration approachse in the field of "BIM-GIS integration": A literature review. *VI-4/W1-2020*, 21–28. https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-21-2020

de Laat, R., & van Berlo, L. (2011). Integration of BIM and GIS: The development of the CityGML GeoBIM extension. In T. H. Kolbe, G. König, & C. Nagel (Eds.), *Advances in 3D geo-information sciences* (pp. 211–225). Springer. https://doi.org/10.1007/978-3-642-12670-3_13

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

Diakite, A. A., Díaz-Vilariño, L., Biljecki, F., Isikdag, Ü., Simmons, S., Li, K., & Zlatanova, S. (2022). ifc2indoorgml: An open-source tool for generating IndoorGML from IFC. *Proc. of the XXIV ISPRS Congress "Imaging today, foreseeing tomorrow"*, *XLIII-B4-2022*, 295–301. https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-295-2022

Donkers, S., Ledoux, H., Zhao, J., & Stoter, J. (2016). Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. *Transactions in GIS*, *20*(4), 547–569. https://doi.org/10.1111/tgis.12162

Fuchs, S., & Scherer, R. J. (2017). Multimodels: Instant nD-modeling using original data. *Automation in Construction*, *75*, 22–32.

ISO 16739-1:2018. (2018). *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries. part 1: Data schema* (tech. rep. 16739-1:2018) [IFC 4.0.2.1 (IFC4 ADD2 TC1)]. International Organization for Standardization. Geneva, Switzerland.

Kolbe, T. H., Kutzner, T., Smyth, C. S., Nagel, C., Roensdorf, C., & Heazel, C. (Eds.). (2021). *OGC city geography markup language (CityGML) part 1: Conceptual model standard* (standard version 3.0.0) [OGC document number 20-010]. Open Geospatial Consortium (OGC). http://www.opengis.net/doc/IS/CityGML-1/3.0

Konde, A., Tauscher, H., Biljecki, F., & Crawford, J. (2018). Floor plans in CityGML. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences: Proc. of the 3D GeoInfo Conference*, *IV-4/W6*, 25–32. https://doi.org/10.5194/isprs-annals-IV-4-W6-25-2018

Krishnakumar, S., Tauscher, H., & Heigener, D. (2023). Floor plan extraction from digital building models. *Proceedings of FOSSGIS (academic track) 2023*, 146–152. https://doi.org/10.5281/zenodo.7576205

Lee, J., Li, K.-J., Zlatanova, S., Kolbe, T. H., Nagel, C., Becker, T., & Kang, H.-Y. (Eds.). (2020). *OGC IndoorGML 1.1* (standard version 1.1) [OGC document number 19-011r4]. Open Geospatial Consortium (OGC). http://www.opengis.net/doc/IS/indoorgml/1.1

Tauscher, H. (2020). Towards a generic mapping for IFC-CityGML data integration. *Proceedings of the 3rd BIM/GIS Integration Workshop and the 15th 3D GeoInfo Conference*, *XLIV-4/W1-2020*, 151–158. https://doi.org/10.5194/isprs-archives-xliv-4-w1-2020-151-2020

Tauscher, H., Lim, J., & Stouffs, R. (2021). A modular graph transformation rule set for IFC-to-CityGML conversion. *Transactions in GIS*, *25*, 261–290. https://doi.org/10.1111/tgis.12723

Tauscher, H., & Wong, K. W. (2022). A testbed for applying operationalized graph grammars to sustainability analysis in integrated BIM-GIS scenarios. *Proceedings of the*

17th 3D GeoInfo Conference in conjunction with 3D cadastres, 147–152. https://doi.org/10.5194/isprs-archives-XLVIII-4-W4-2022-147-2022