7th International Building Physics Conference

IBPC2018

Proceedings

SYRACUSE, NY, USA

September 23 - 26, 2018

Healthy, Intelligent and Resilient Buildings and Urban Environments

ibpc2018.org | #ibpc2018

Big-open-real-BIM Data Model - Proof of Concept

Galina Paskaleva[1,*], Sabine Wolny[1] and Thomas Bednar[1]

[1]Research Division of Building Physics and Sound Protection, Institute of Building Construction and Technology, TU Wien, Vienna

[*]*Corresponding email: galina.paskaleva@tuwien.ac.at*

ABSTRACT

The goal of Building Information Modeling (BIM) is the continuous use of digital construction models from the planning stage onwards. The affected processes are iterative and involve multiple stakeholders who work at varying pace and in varying levels of detail. These stakeholders require highly specific tools based on diverging data models. To satisfy all those requirements one of the best known Open BIM implementations – IFC – offers a data model containing more than one thousand different types – from basic to highly specific. Due to its complexity, potential users must undergo prolonged training. The even bigger challenge for IFC, however, is keeping up with the updates of building regulations or with the ever expanding state of the art in simulation tools. Our approach, SIMULTAN, in contrast to IFC, consist of 26 different basic types. They can be combined to increasingly complex models, which can themselves be used as types for other models. This enables each domain expert to create a custom data structure for any specific task, which is automatically compatible with the data structure of any other domain expert using the same basic types. It shortens the training time and facilitates the loss-, corruption-, and conflict-free exchange of information between domain experts, which is a key aspect of BIM. As a use case, we present the calculation of the U-Value of a multi-layered wall. We compare number, complexity and adequacy of the necessary data modelling steps in IFC4 and in SIMULTAN. The result shows that the flexible data model of SIMULTAN can be better adapted to the task. Another significant advantage of SIMULTAN is its inbuilt separation of responsibilities at the level of the most basic types, which, when combined with secure transaction technologies, can enable safe, effective and easily traceable interaction among stakeholders.

KEYWORDS

data model, BIM, information exchange, multidisciplinary

INTRODUCTION

The idea of the *Building Information Modeling (BIM)* is the consistent use of digital building models from planning to realization, from the operational phase to demolition. Already in the 70s, Eastman (1975) published a concept for the construction and the use of virtual building models. In 1992, van Needervan and Tolman (1992) first used the term BIM. One huge advantage of digital building models is the lossless information exchange. However, to ensure this exchange, the whole model must be interoperable, including all information such as climate data, usage information, variants, etc.

The interoperability of BIM models is present to varying degrees depending on the type of BIM. *Little BIM* refers to the use of a specific BIM software by a single planner. In this case, BIM is used without external communication (Jernigan, 2008). *Big BIM* refers to consistent model-based communication between all stakeholders involved in all phases of a building's life cycle. In addition, a distinction is made between *closed* and *open BIM*, depending on whether vendor-neutral data exchange formats are used or not (Borrmann 2015).

An existing and already widely used standard for BIM is the Industry Foundation Classes (IFC) (buildingSMART, 2018) definition. The IFC models hold geometric data as well as metadata about building objects and are designed to support interoperability. Steel et al. (2012) investigate various issues of model-based interoperability in exchanging building information models between different tools, with particular focus on the use IFC. The authors pointed out that one of the greatest challenges with regard to interoperability is the inconsistency of modelling styles. The modeling language should clearly describe all possibilities so that unexpected alternatives are not possible. The authors Polit-Casillas and Howe (2013) also take up the issue of complex systems and the synchronization of different types of information. In their work, they combine BIM with a systems engineering approach to obtain a model based engineering approach. Thereby, they focus on the interoperability and validation of information in different planning phases, from requirements modelled in a modelling language such as the systems modeling language (SysML) to models drawn in, e.g., Computer Aided Design (CAD) tools.

In this paper, we move towards a big open BIM data model where the information is available at the level of detail at which it is required. This means that general information or information placeholders can be defined at the beginning of planning and later on be refined to become a realistic model that depicts the system behaviour. Our approach has the advantage that the complexity of the data model itself does not increase with the complexity of the represented buildings.

MOTIVATING EXAMPLE

Let us consider the case of a wall that consists of material layers and has a geometry, which sets an upper limit to its total thickness. The wall, the material layers and their respective materials have properties that are necessary for various calculations (e.g. of the U-Value). Fig. 1 displays an excerpt of the IFC4 data model (buildingSMART, 2013) in Universal Modelling Language (UML) notation (OMG, 2017) that enables the storage of that information. The coloured paths show all types and associations involved in establishing various relationships between objects. In the case of linking a wall with a property set we have just two objects that require the maintenance of 13 types and 3 associations (path 2 in Fig. 1). It must be taken into account that type and object are not synonyms. The distinction between a type *material* and an object *m1* of type *material*, for example, is as follows: The type *material* requires all objects conforming to it to have a text parameter *Name*. Object *m1* conforms to type *material* and therefore has a specific text value (e.g. *Wood*) associated with the parameter *Name*. In other words, the type can be regarded as a template for the creation of (an unlimited number of) objects and the associations between types – as information exchange contracts between the corresponding objects. Fig. 1 contains only types.

Our objective is to decrease complexity by minimizing the number of types and associations required for the production of any object.

MULTI-LEVEL DATA MODEL

The complete data model of IFC4 contains a total of 1167 type definitions and tens of thousands of associations among them (see Fig. 1). This data model has to be implemented and maintained in each software that manipulates IFC4 object models. Partial implementations carry the risk of loss of information across software boundaries.

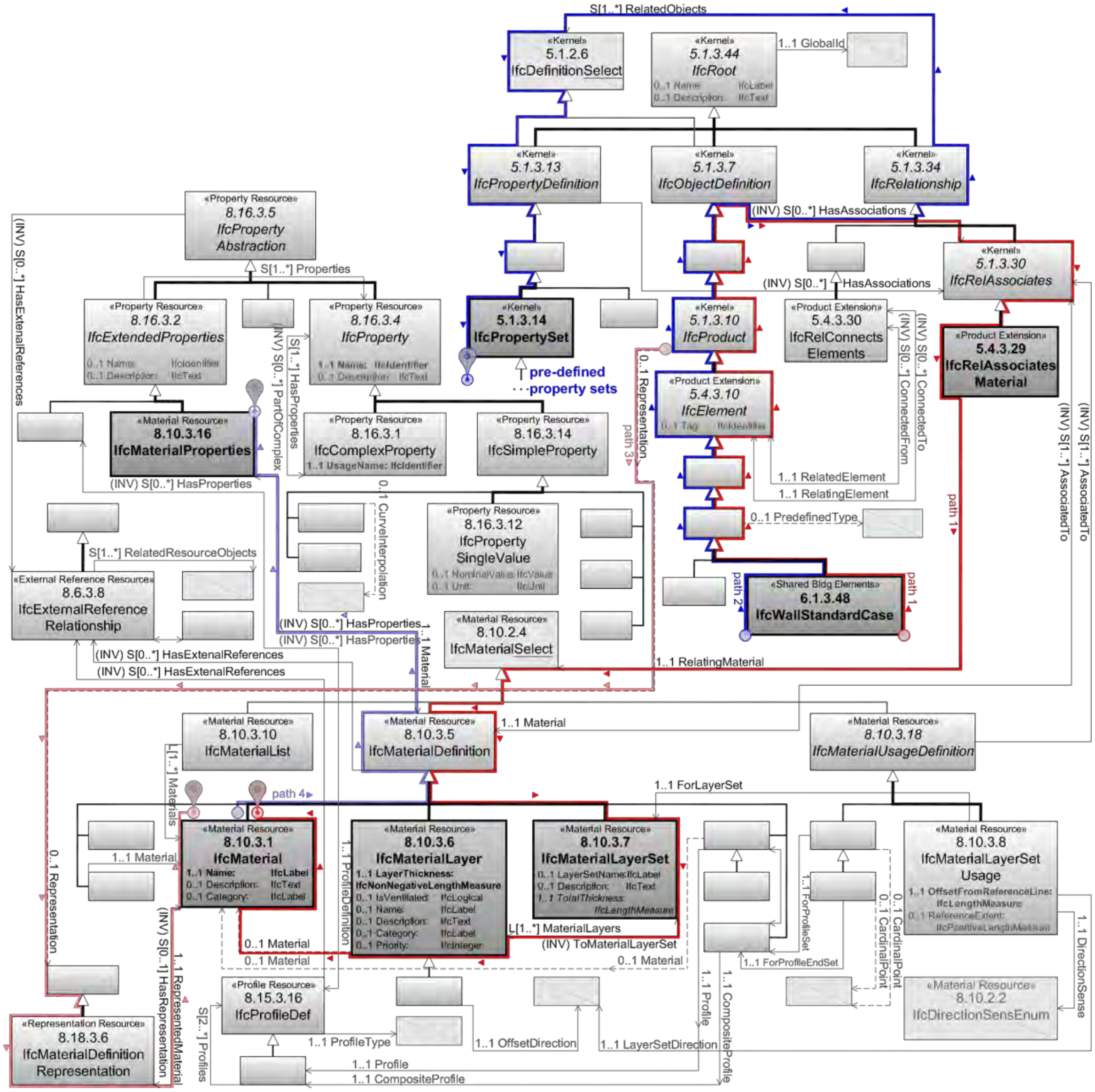

Fig. 1. An abbreviated excerpt of the IFC4 specification. The coloured paths show all types and associations necessary to establish a relationship between two objects: path 1 (in red) - between a wall and a material, path 2 (in blue) - between a wall and its properties, path 3 (in light red) – between a product and a material, and path 4 (in light blue) – between a material and its properties.

IFC4 has a fixed type structure. The information-carrying part – the parameters - are provided as property and quantity sets and are maintained separately (e.g., hosted on a server). It is possible to associate any type with any property or quantity set – a significant loosening of the constraints of the previous IFC version, 2x3. In essence, IFC4 provides a two-part data model that separates data structure from data content. The content has practically unlimited flexibility, since users can create their own property and quantity sets. The structure is rigid and can only be used as a container, but not as a carrier of information.

The SIMULTAN data model takes the next step and uses the type structure as an information carrier in addition to numerical and textual parameters. The type structure of any data model is its ontology (Liu et al. 2013). A flexible, or editable, ontology allows the continuous incorporation of expert knowledge, as opposed to the development iterations of a data model with a fixed ontology that allows this once every few years (Laakso and Kiviniemi, 2012). Fig. 2 shows the SIMULTAN data model. Its central concept is the *Component* - a type that allows the definition of other types. The definition of *Component* includes a recursive relationship named *Subcomponents* (see Fig. 2), which enables each component to have an arbitrary number of sub-components, each with its own name and set of *Parameters* and *Calculations*. In this way, each component depicts its own ontology (or formal type definition) in its structure. On the one hand, the values of its parameters can be set to a default, in which case the component plays the role of a type or a template (e.g., *Material*). On the other hand, they can be set to values specific for light concrete, in which case it plays the role of an object (e.g., object *Light Concrete* conforming to type *Material*). The refinement of types (inheritance level) and the production of objects (instantiation level) are both possible within the same data model and can be applied an arbitrary number of times. For that reason, SIMULTAN is a multi-level data model.

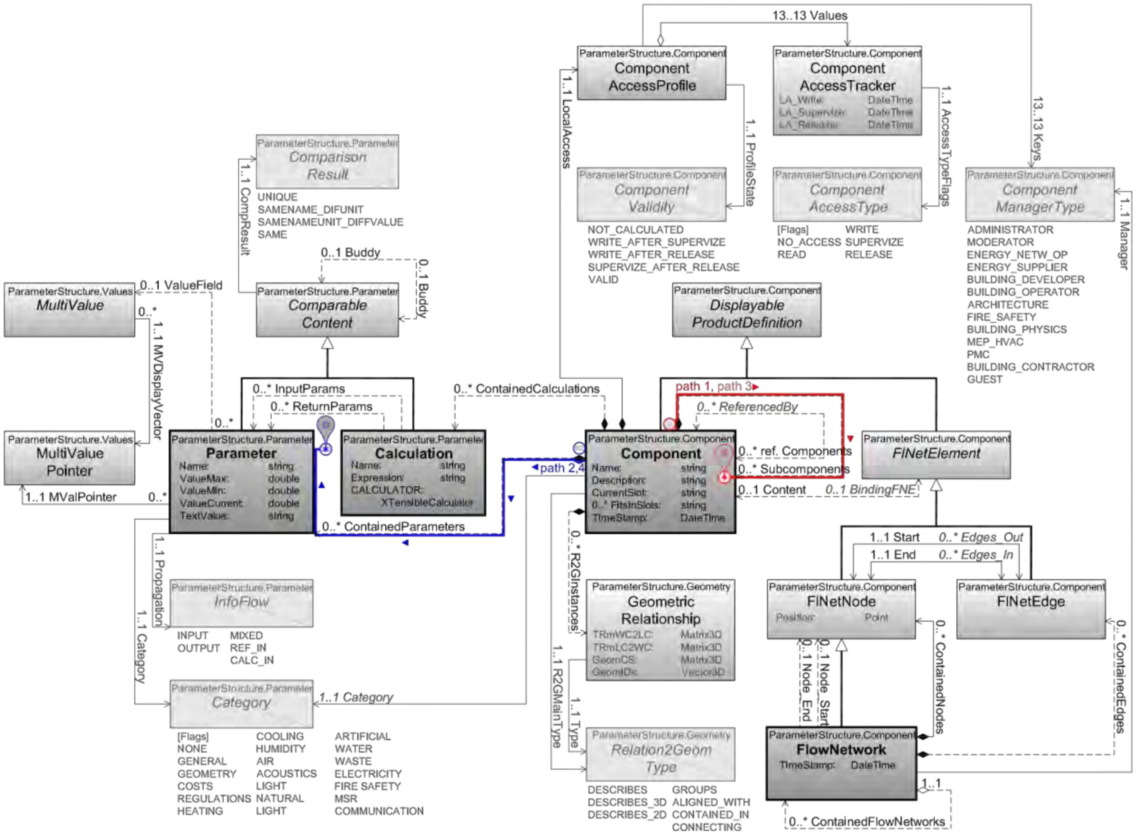

Fig. 2. SIMULTAN data model (the 5 subclasses of *MultiValue* were omitted). The coloured paths show the same relationships as in Fig. 1.

The main advantage of SIMULTAN's simple structure is its ability to depict any data model more complex than itself (e.g., IFC4) through nesting and referencing of components or their incorporation in a flow network (for multivalent dependencies). Thus, it is a universal translator between data models, as it is very easy to implement, maintain and map to and from other structures. It is also flexible enough to incorporate any data structure necessary for any calculation or simulation method. We will demonstrate this in the next section.

PROOF OF CONCEPT

As demonstration case for our proof of concept, we use the calculation of the U-Value of a multi-layered wall according to EN ISO 6946:2017. As shown in Fig. 1, the association of a wall with a material in IFC4 involves 14 types and 5 associations. For the same purpose in SIMULTAN we need a component *wall* referencing a component *wall construction* (see path 1 in Fig. 1, Fig. 2 and Table 1). Table 1 summarizes the difference in complexity of the IFC4 and SIMULTAN data models and their limitations.

Table 1. Quantitative comparison between the IFC4 and the SIMULTAN data models.

Comparison Criteria	IFC4 Data Model	SIMULTAN Data Model
total no of type definitions	776 entities and 391 types, including 206 enumerations and 59 select types: **1167** in total	19 classes and 7 enumerations: **26** in total
total no of pre-defined parameter collections	408 property sets and 91 quantity sets: **499** in total (not contained in the formal IFC4 specification)	-
total no of user-defined parameter collections	unlimited	unlimited
total no of user-defined calculations	-	unlimited
min. no of types, objects and associations included in path 1	14 types, 2 objects, 5 associations	1 type, 2 objects, 1 association
min. no of types, objects and associations included in path 2	13 types, 2 objects, 3 associations	2 types, 2 objects, 1 association

For the U-Value calculation we now need the following parameters – the external and internal surface resistance R_{se} and R_{si}, and, for each homogenous material layer, the thickness d and the design thermal conductivity λ. In IFC4, this calculation requires the association of a material object with the property set *8.10.5.10 Pset_MaterialThermal,* containing a *ThermalConductivity* property in addition to *BoilingPoint*, *FreezingPoint* and *SpecificHeatCapacity*. The wall object also needs to be associated with *6.1.4.23 Pset_WallCommon,* containing a *ThermalTransmittance* property (according to the documentation, corresponding to the U-Value) in addition to 10 others (buildingSMART, 2013). The thickness of each material layer is a direct attribute of the type *IfcMaterialLayer* (see Fig.1). IFC4 has a type *IfcThermalResistanceMeasure* but no pre-defined property or quantity set containing a property corresponding to R_{se} or R_{si}, which necessitates the definition of a custom property set. In summary, in order to depict the calculation of the U-Value in IFC4, we need two pre-defined and one user-defined property set, we need to maintain 14 redundant entries in these property sets, and any calculation method using this structure has to read and write both to object values (material layer thickness, total wall thickness) and to parameter sets.

The SIMULTAN data model, on the other hand, allows the user to define exactly the parameters needed for the specified calculation, since additional parameters or calculations can be added later. One possible expression of such data structure can consist of the following: A component *wall construction* contains parameters R_{se}, R_{si} and *U-Value*. Each material layer is a sub-component of it, with parameters d and the thermal resistance of the layer R, references a component carrying the material properties, in this case only λ, and calculates R on its own. The component *wall construction* gathers the information from its sub-components into its own calculation(s) and determines its own *U-Value*. In this way, SIMULTAN enables not just the efficient storage of information but also manages its flow.

CONCLUSION AND OUTLOOK

In order to be able to incorporate new methods and technologies data models need flexibility. IFC4 has already made the first step by decoupling data structure from data content. SIMULTAN takes the next step and introduces an adaptable data structure that can define any domain-specific ontology (e.g. building physics) in addition to the traditional numeric and textual parameters and can act as a universal translator. However, with a greater flexibility comes also a greater responsibility for the domain experts defining new data structures. Therefore, SIMULTAN incorporates an access tracking system (see *ComponentAccessProfile* in Fig. 2) as the mandatory equivalent of IFC4's *IfcActorResource*, where each component has an owner solely responsible for its development. Other actors (or stakeholders) can have supervision or publication rights. Thus, our data structure enables the stakeholders to view and modify all necessary information within their workflow. Applications using this data structure adapt to the user's workflow instead of forcing the user to adapt to the application. In a future development step, this system can be coupled with secure transaction technologies, such as BlockChains (Puthal et al. 2018), to provide a solid foundation for reliable and effective interactive work in real time.

ACKNOWLEDGEMENT

The Austrian Ministry for Transport, Innovation and Technology (bmvit) supported this work.

REFERENCES

Borrmann A., König M., Koch C. and Beetz J. 2015. Building Information Modeling. Technologische Grundlagen und industrielle Praxis. Wiesbaden: Springer Vieweg.

buildingSMART 2018. http://www.buildingsmart-tech.org/specifications/ifc-overview

buildingSMART 2013. http://www.buildingsmart-tech.org/specifications/ifc-releases/ifc4-release

Eastmann C. 1975. The Use of Computers Instead of Drawings. AIA Journal, Volume 63, Number 3, pp 46-50.

Jernigan F. E. 2008. BIG BIM little bim - The practical approach to Building Information Modeling - Integrated practice done the right way! 2nd ed., 4 Site Press, Salisbury (MD), USA.

Laakso M. and Kiviniemi A. 2012. The IFC Standard - A Review of History, Development, and Standardization. Information Technology in Construction, Volume 17, pp 134-161.

Liu J. N. K., He Y. L., Lim E. H. Y. and Wang X. Z. 2013. A New Method for Knowledge and Information Management Domain Ontology Graph Model. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Volume 43, Number 1, pp 115-127.

OMG 2017. https://www.omg.org/spec/UML

Polit-Casillas R. and Howe A.S. 2013. Virtual Construction of Space Habitats: Connecting Building Information Models (BIM) and SysML. In: *Proceedings of the AIAA Space 2013 Conference and Exposition*, San Diego, (AIAA 2013-5508).

Puthal D., Malik N., Mohanty S. P., Kougianos E. and Yang C. 2018. The Blockchain as a Decentralized Security Framework. IEEE Consumer Electronics Magazine, Volume 7, Number 2, pp 18-21.

Steel J., Drogemuller R. and Toth B. 2012. Model Interoperability in Building Information Modelling. *Software and Systems Modeling,* 11(1), .99-109.

van Nederveen G. and Tolman F.P. 1992. Modelling multiple views on buildings. Automation in Construction, Volume 1, Number 3, pp 215-224.