CIB World Building Congress 2019
Hong Kong SAR, China
17 – 21 June 2019

SIMULTAN as a Big-Open-Real-BIM Data Model - Proof of Concept for the Design Phase

Galina Paskaleva,
Research Unit of Building Physics,
Institute of Material Technology, Building Physics, and Building Ecology, TU Wien, Vienna
(email: galina.paskaleva@tuwien.ac.at)
Thomas Lewis,
Research Unit of Building Physics,
Institute of Material Technology, Building Physics, and Building Ecology, TU Wien, Vienna
(email: thomas.lewis@tuwien.ac.at)
Sabine Wolny,
Research Unit of Building Physics,
Institute of Material Technology, Building Physics, and Building Ecology, TU Wien, Vienna
(email: sabine.wolny@tuwien.ac.at)
Bernhard Steiner,
Research Unit of Building Physics,
Institute of Material Technology, Building Physics, and Building Ecology, TU Wien, Vienna
(email: alexander.david@tuwien.ac.at)
Thomas Bednar,
Research Unit of Building Physics,
Institute of Material Technology, Building Physics, and Building Ecology, TU Wien, Vienna
(email: thomas.bednar@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, consists of approximately 30 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 proof-of-concept we show the application of our data model in an ongoing project. We demonstrate the completeness of our data model and the fulfillment of the requirements of the design phase. 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, design phase

1. Introduction

The maintenance of digital construction models of buildings from the planning stage until demolition becomes more and more important as a dynamic overview of processes and as a tool for adapting designs and constructions more easily to upcoming requirements (Borrmann et al, 2015). The idea of the Building Information Modeling (BIM) is the consistent use of such digital building models from planning to realization for preservation and exchange of building information (Borrmann et al, 2015).

In the last years, there have been several approaches in dealing with open and closed BIM solutions to realize such digital models for, e.g., energy performance assessment, interoperability and validation of information in different planning phases, and exchanging building information models between different tools (Schlueter and Thesseling, 2009; Polit-Casillas and Howe, 2013; Steel, Drogemuller and Toth, 2012). However, there are still challenges, such as the automation of data capture, the update and maintenance of information in BIM and the handling of uncertain data, objects and relations (Volk, Stengel and Schultmann, 2014).

In this paper, we focus on the support of the building planning phase by a new data model called SIMULTAN (Paskaleva, Wolny and Bednar, 2018) as a big open BIM data model. We will demonstrate how our data model facilitates the fulfillment of planning workflow requirements in the design phase.

The remainder of the paper is structured as follows. In the next section, we present the motivating example of a current workflow in the design phase underlying the presented approach. In Section 3, we describe our data model. Section 4 shows the proof of concept for our approach. Finally, we conclude this paper by an outlook on our next steps in Section 5.

2. Motivating Example of a current workflow in the Design Phase

The construction of a building from the planning stage onwards is a continuous process involving multiple stakeholders depending on the project size:

1. Investor
2. Project developer
3. Architect
4. Building physicist
5. HVAC engineer
6. Electrical engineer
7. Energy consultant
8. Authority

All these have to collaborate in order to successfully design and construct buildings. Certain requirements to the software were derived from supporting this collaboration.

Consider the following motivating example of a typical building planning workflow with various requirements to a data model and to corresponding methods. We focus on the necessary technical software features and do not distinguish between the project phases predesign, design, and approval (e.g., as distinguished in (DIN-Normenausschuss Bauwesen, 2018; Ausschuss der Verbände und Kammern der Ingenieure und Architekten für die Honorarordnung e. V., 2013)).

The investor intends to erect three buildings comprising a shopping mall, offices and residential areas (flats). Key interest is in a high return on investment translating basically to a) high income and b) low life cycle cost. There are two crucial boundary conditions:

- An offer from the local grid operator will save an annual discount of 100.000 Euro on grid fees if planners manage to stay by guarantee below an electrical peak power demand of YY. This would avoid a costly increase of grid capacity.
- Buildings shall be of an outstanding attractiveness on the market. The goal is a BREEAM certificate.

Thus, data on multiple buildings has to be held by the model - not via a workaround such as a "multiplication factor" that would allow for several "building clones" but the data of each building has to be stored separately. **Req. 1 Multibuilding administration**

Grids and networks describe the relations between the buildings and the system boundary (e.g., connection to higher-level grid). The electric grid as a component contains subcomponents such as cables, transformers etc. Parameters of these subcomponents, e.g., are cable cross sections and sizes (capacities) of the transformers. **Req. 2 Manage data on grids/networks**

The project developer translates the investor's requirements into technical specifications for the planner:

- Thermal comfort in all offices and residential areas has to be of category I (best) according to EN 7730. In all other areas foreseen for a regular presence of people, we will have at least category II.
- The maximum electrical peak load will be partly ensured by intelligently managing and shifting loads between the three buildings, their zones and single loads.

Based on these framing conditions the architect establishes a (preliminary) design. All building zones are visualized and edited in a Computer Aided Design (CAD) tool. Faces and volumes should not be "loose by themselves" but each geometric element is strictly tied to its respective component. Raw geometry may be directly created from scratch in the CAD tool, imported, or established as a combination of importing and reediting. Any component, such as a ventilation plant, a switch cabinet and even tiny elements, such as sensors, are attributed a precise location in 3D space and thus may be visualized in the CAD viewer. **Req. 3 View on components in three dimensions and in time**

On the basis of the predesign received from the architect (layered walls, floors, roofs, windows etc.), the building physicist develops safe constructions, taking into account the following factors:

1. Thermal and hygric performance and durability of each construction
2. Air tightness, in particular junctions between constructions and penetrations, such as via pipes
3. Sound protection and acoustic performance
4. Fire safety

These specifications have to be assigned as attributes to geometric components. This assignment is one of the goals of BIM. **Req. 4 Attaching any kind of alphanumeric attributes to geometric objects**

The architect wants to prove the compliance with the required elevated thermal comfort. For instance, in the CAD viewer all constructions (e.g., walls) with U-value better than 30% than the legal requirement (i.e. below it) should be highlighted in a special color. **Req. 5 Visualization of component attributes in 3D space**

Based on the specifications and information provided by the different stakeholders, the investor wants to know the life cycle costs (LCC) of the project, which result from looping across many components adding the according (forecast) costs of, for example, fuel, maintenance and repair. Thus, to calculate the LCC, it needs to reference all components that accrue cost. **Req. 6 Components may reference each other**

Now the building performance has to be assessed. HVAC and electrical engineer enter power ratings as parameters of single units such as ventilation plants and heaters, as well as of entire rooms (e.g., based on the number of publicly accessible electrical plugs). For each of these load components, we need a load profile (time series). Some profiles result from a dynamic building simulation and some are directly specified by, e.g., the energy consultant. In addition, it has to be defined how many persons are present when and where, how (metabolically) active they are and how warm they are dressed. These time series (load profiles, presence of people and also climate data) are records with time stamps that are imported, held and used for further processing such as analysis and calculations in the software. **Req. 7 Holding and visualizing time series information**

Various actors provide their data in different formats and structures. In order to use them for, e.g., calculations the data has to be reordered and regrouped. Foremost, this requires a number of matrix operations, which should be supported by the used software. **Req. 8 Recalculating (large) data**

Another important part in the design phase is the definition of the control and operation system. The HVAC engineer delivers a textual description how the various physical components of the HVAC system are controlled. This human readable description has to be transformed into machine code for allowing a fully dynamic thermal simulation to identify potential gaps in the control logic.
Req. 9 Model control systems and logics between components

All information about components, networks, relations between components is needed for various computations via specific software tools. To exchange information and results matching interfaces between the tools are required. For example, the energy consultant runs a thermal simulation in one tool including a daylight simulation, which results in load profiles for the ventilation plants, the heaters, coolers and for lighting. **Req. 10 Providing interfaces to external (simulation) software**

In the building project, there are different responsibilities. For example, the HVAC engineer wants to reduce the pressure loss in the air ducts that requires the suspended ceiling to be lowered by 1 cm. While the HVAC engineer can see the geometric collision in the CAD model, only the architect should be responsible for lowering the ceiling accordingly. Otherwise, the stakeholders end up with different versions of the project. **Req. 11 Managing roles and rights**

Often it is necessary to change settings after the first draft to comply with certain requirements, e.g., the budget limit. For this purpose, specific components are adapted to find out which version works best. However, the different versions should not be overwritten, but kept available. For instance, the architect modifies the glass type of a high façade and triggers a rough thermal simulation for each glass variant. The consultant/physicist runs a CFD simulation on such pre-selected glass products in order to see for specific spatial areas (subzones) if the thermal comfort is met. **Req. 12 Manage variants**

Finally, the authority requires the compliance with a number of conventional legal requirements such as statics, fire safety, acoustic performance, minimum daylighting and energy efficiency.
Req. 13 Keeping an overview on fulfilling requirements

The software delivers technical reports as demanded from the point of view of each actor.
Req. 14 Reports reflecting different views of various stakeholders

3. Completeness of the data model SIMULTAN for the Design Phase

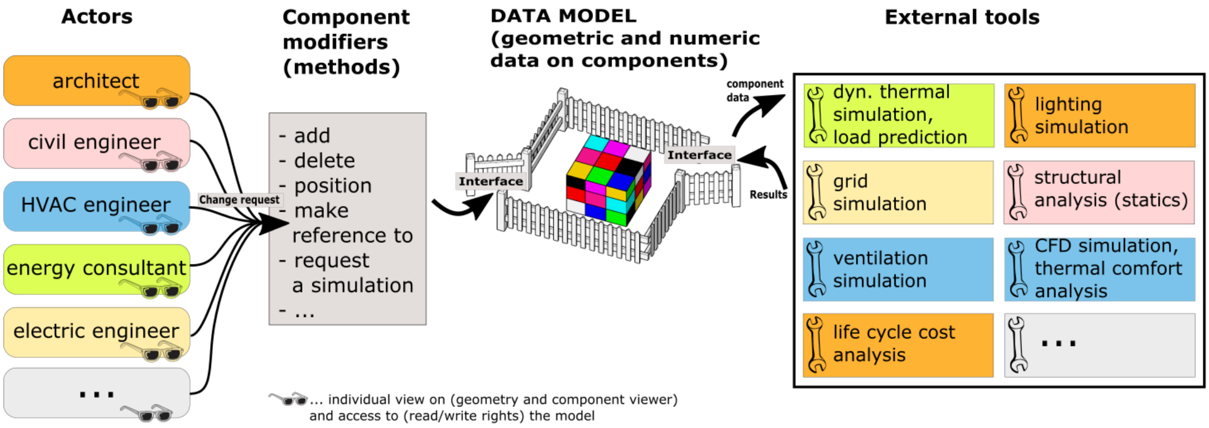

Figure 1: Scheme of the SIMULTAN workflow involving actors, the data model and external calculation tools.

In this section, we present our data model SIMULTAN and its suitability for supporting the design phase. Figure 1 shows the overview of the data model and the involved actors. Each actor has an individual view onto the data model, including read and write rights for each component according to standardized roles. Currently there are 13 roles, such as architect and building physicist. Actors make change requests to the data model exclusively via pre-defined methods. This ensures the model consistency. There are two fundamental views onto the data model. The 3D model view as implemented in any CAD program provides information on the geometric relations (lines, faces, volumes). A second list-style backend view shows all model components and their parameters including many components that have no immediate representation in the 3D model such as wall constructions, materials, networks of any kind, e.g. an electric grid, or mere data series.

The typical workflow of a planning team uses the geometry as a means of communicating requirements, conditions, conflicts, updates etc. A wall, for example, is geometrically a surface with a length and a width. However, a wall also needs to hold information about material, thickness, thermal, and hygric properties (see Figure 2). Thus, a component may have a reference (a pointer) to any other component (excluding cyclic references) addressing it as subcomponent (Paskaleva, Wolny and Bednar, 2018).

To support such geometric objects by a data model, dedicated structures and semantics are required. The geometry in itself contains spatial relationship semantics. In three dimensions an architectural space can be represented by a volume (see type *Volume* in Figure 3), the enclosing walls, floor and ceiling or roof can be represented by the surfaces defining the volume's boundary (see type *Face* in Figure 3). The types *EdgeLoop*, *Edge*, and *Vertex* complete the geometric data model by allowing the representation of 1-dimensional surface boundaries, and 1- and 0-dimensional elements, respectively. The types *PFace* and *PEdge* enable the efficient traversal of the geometry model by encoding containment and neighborhood information. This realization is based on the Partial Entity Structure described in (Lee and Lee, 2001). In order to utilize the geometric semantics as an interface for other design information we have developed a method of connecting the geometric data model to a linguistic data model that is capable of carrying any information representable by textual or numeric data. That linguistic data model consists of four major elements: *Component*, *Parameter*, *Calculation* and *Geometric Relationship* (see Figure 3) that fulfill the role of syntax, or grammar, of all models conforming to it (such as the model in Figure 2). Components act as containers – they can contain other components as well as parameters and calculations based on these parameters – see relationships '*Subcomponents*', '*Contained Parameters*' and '*Contained Calculations*' respectively. Components can also reference other components via the relationship '*ref. Components*'. All these features enable the construction of arbitrarily complex data structures, or models. Using our data model, components can be represented, alphanumeric attributes can be assigned to geometric objects, and components can reference each other. Thus, the SIMULTAN data model is able to fulfill **Req. 3** to **Req. 6**.

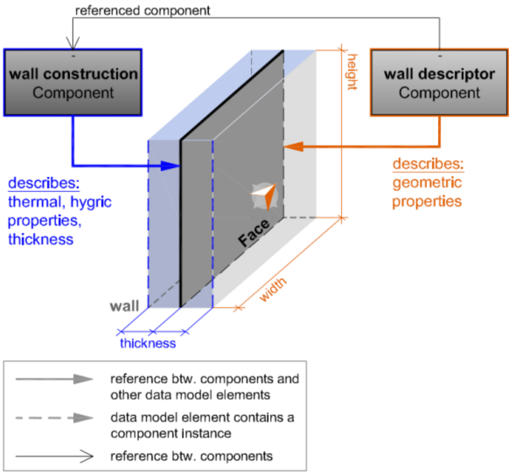

Figure 2: SIMULTAN data model applied to a specific surface (Face). A Component instance, the wall descriptor (in orange), references the surface and extracts geometric information (e.g. area, size, orientation). Another Component instance, the wall construction (in blue), references the same surface. It supplies information to the surface – the material layers, the total thickness, etc. The surface acts as an interface between the two component instances and enables them to reference each other (see the relationship 'referenced component') and to exchange further information.

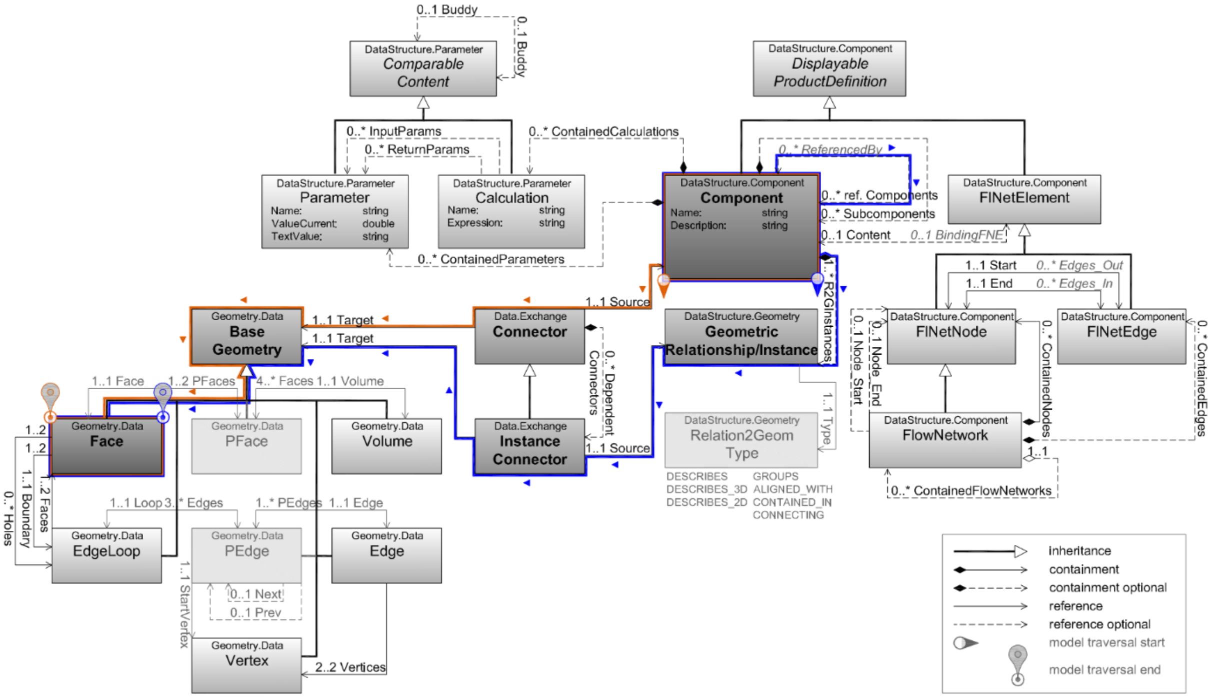

Figure 3: Data model excerpt illustrating the interaction between the geometry model and the generic component model that can carry any user-defined semantics (see type Component). The orange path traverses the model elements involved in associating a wall representation with a geometric surface. The blue path shows the model elements that assign a wall construction representation to a geometric surface.

For the fulfillment of **Req. 2** the data model must support the management of networks and their components, which is shown in Figure 4 and Figure 5. The structure of a component – e.g., the definition of a ventilation duct – is independent of the structure of a system, e.g., the entire ventilation network. The ventilation duct component may have a sub-component containing its properties influencing the airflow, including the shape of its cross-section, and another sub-component containing information about materials and mounting, possibly referencing other components carrying material properties. The duct definition itself does not carry any connectivity information, as the same duct definition is used multiple times in a different context (see the thick lines in Figure 4). In this case the component (of type Component, see Figure 5) acts as a template, whereas the instances (of type Geometric Relationship / Instance, also in Figure 5) act as the specific copies, e.g., a specific section of the duct with a specific size, placed in a network element.

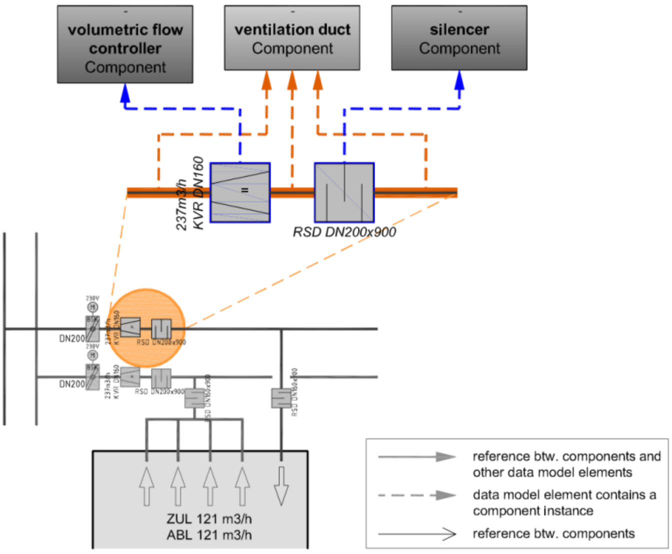

Figure 4: An excerpt of the model of a ventilation system. The network nodes highlighted in blue contain an instance of the volumetric flow control and the silencer components, respectively (see the dashed arrows). Each of the three duct sections highlighted in orange contains its own instance of the ventilation duct component – each with different length and, possibly, cross section size. The cross section shape, the material of the duct wall, whether it is insulated or not, is defined in the component itself.

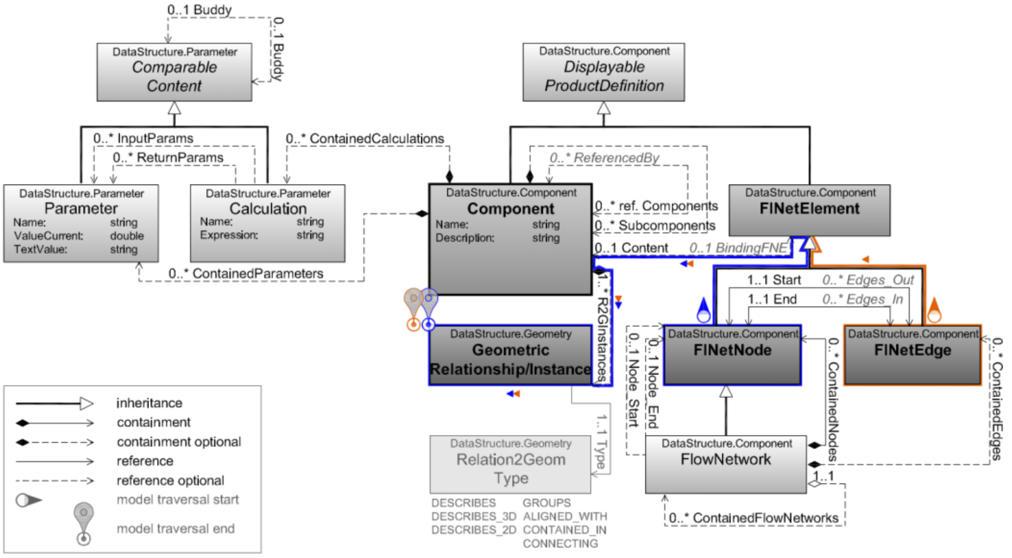

Figure 5: The data model elements allowing the definition of building service networks. A Flow Network consists of nodes (FlNetNode), edges (FlNetEdge) connecting them, and of sub-networks (see relationship 'ContainedFlowNetworks'). The orange path traverses the model elements enabling the placement of the instance of a component in a network edge, the blue path traverses those enabling the placement of the instance of a component in a network node.

Figure 5 shows the elements of the data model enabling the construction of the flow network. Those elements (*FlNetNode*, *FlNetEdge* and *FlowNetwork*) create a hierarchical cyclical directed multigraph (where multiple edges connecting the same two nodes are admissible). Each node allows the definition of rules (sum, maximum, etc.) for the propagation of the parameter values of the component instances contained in it and in its incident edges (e.g., flow speed or pressure loss). The separation of concerns – in components and networks – contributes to the simplicity of the data models of both. Those data models are, consequently, easier to manage and maintain. In addition, the network components can also be located in the geometric view according to the same principle as shown for the wall.

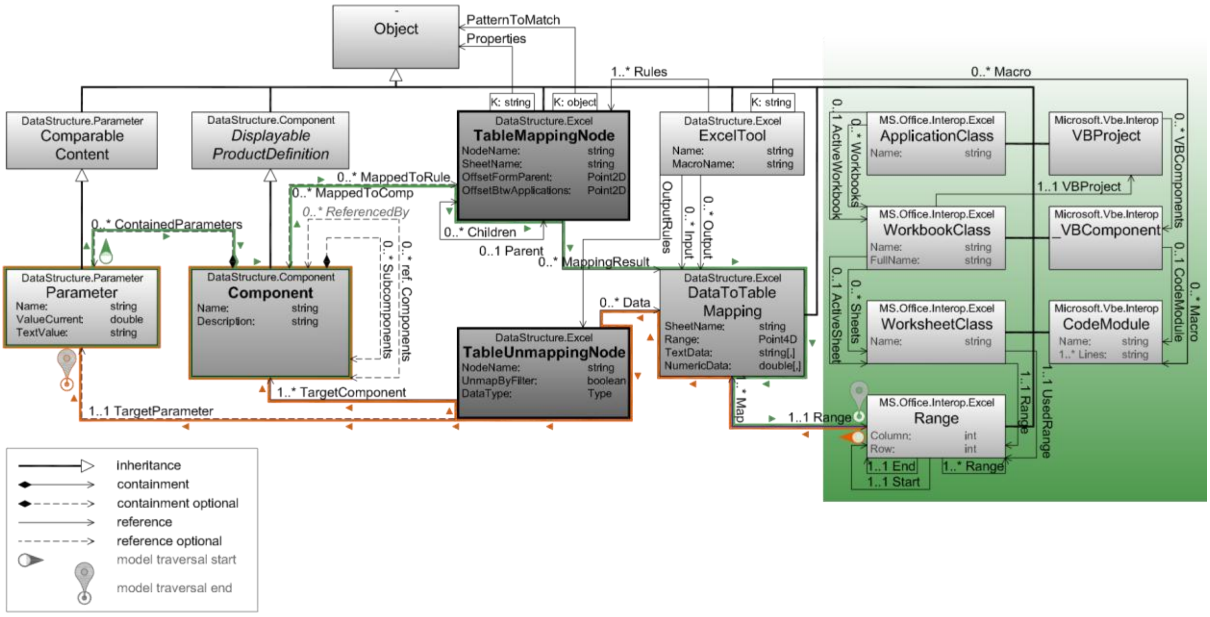

Figure 6: The interface to Microsoft Excel ™. An excerpt of our data model is on the left, the Excel API data model parts we communicate with – in the green box on the right. The green path traverses the elements of the data model that supply input from a component's parameters to a cell within an Excel worksheet. The orange path traverses the elements that supply the output of a cell in Excel to a component or its parameters. The mapping of in- and output is enabled by the types Table Mapping Node, Table Unmapping Node and Data To Table Mapping

Once the component model is created, it allows the efficient examination of multiple variations by calling external simulation tools. For instance, one such tool is Microsoft Excel ™. We use its ability to apply formulas and complex calculations within VBA modules to the contents of multiple spreadsheets. This means that we do not need to maintain those complex calculations within the data model, but can, instead, handle them separately and, most importantly, update them without having to change the data model itself. Microsoft Excel ™'s API enables the communication with our data model via mapping (see Figure 6). The user can define rules for mapping the contents of a component to ranges of cells in a worksheet (see type *Table Mapping Node* in Figure 6). The collecting of the calculation result is accomplished via the definition of un-mapping rules from ranges of cells in a worksheet to components determined via a filter or to specific components and parameters (see type *Table Unmapping Node* in Figure 6). This part of the data model fulfillments requirements **Req. 7**, **8**, and **10**.

To manage roles and rights (**Req. 11**) and in order to avoid liability issues, resulting from a model's complexity, each component is supplied with an access profile (see *Component Access Profile* in Figure 77). This profile grants rights and monitors the activities of a number of typical stakeholders (e.g., architect, building developer, building services engineer, etc. – see *Component Manager Type*). There is only one stakeholder (aside from the administrator) with writing access, and arbitrarily many with reading, supervising and release access. The history of user actions allows the calculation of the *Component Validity*. For example, changing the component after it has been approved by another stakeholder produces an invalid state (WRITE_AFTER_SUPERVIZE).

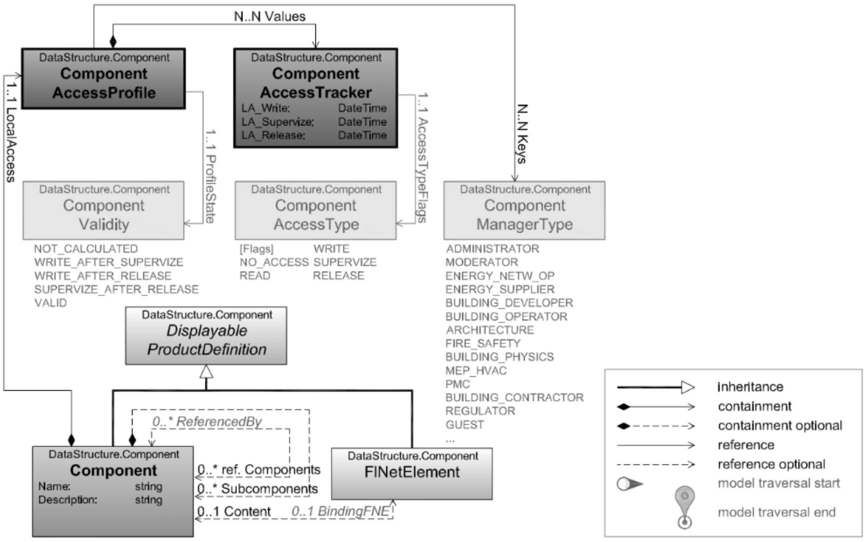

Figure 7: Access management of the data model. Each component contains an access profile (see Component Access Profile) which provides rights and manages the activity of a potentially unlimited number of users (see Component Management Type). Only the administrator and one other user can have writing access, others can have reading, supervising or release rights to the same component.

4. Proof of Concept

The presented SIMULTAN data model is applied to the real scenario of four large building projects. The Viennese airport (FWAG) is currently developing two refurbishing projects and two new buildings (office building and a kind of shopping mall). During the projects the 3D geometry of the four buildings is drawn in the CAD viewer of SIMULTAN, based on the imported .dwg files received from the planner. Material layered constructions are established based on the planners' specifications and assigned to walls, ceilings, floors and windows. Components, such as air ducts, ventilation plants and electrical transformers, are positioned for testing purposes in the CAD viewer. The thermal performance of all buildings is assessed in a dynamic thermal building simulation. This is the basis for estimating the impact on additional electricity consumption, which is of great interest to the airport. Figure 8 shows a multi building administration view where different object information is presented based on the data held in the data model.

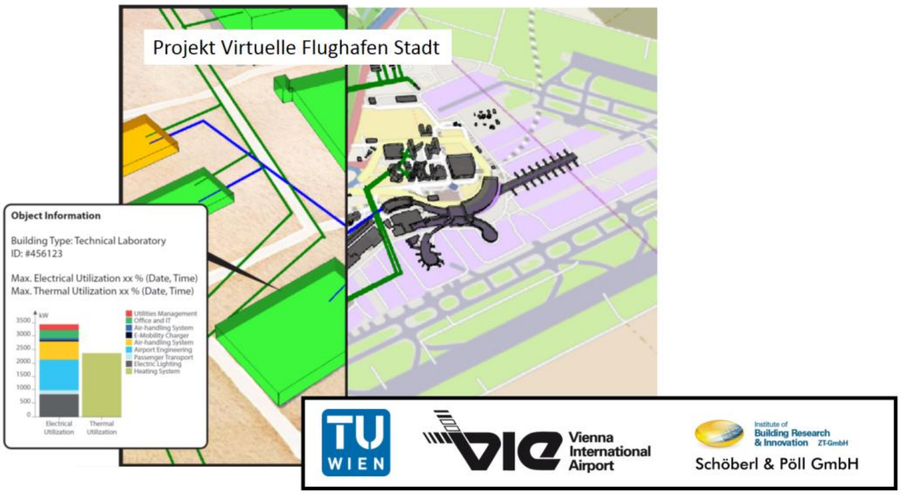

Figure 8: Overview of the Virtual City Airport with multiple buildings and object information

5. Conclusion and Outlook

Design complexity does not require data model complexity. Just as only a few arithmetic operators allow the formulation of complex mathematical relationships, so does a simple data model allow the maintenance of arbitrarily complex designs by providing the appropriate set of data elements and relationships between them. SIMULTAN with its 30 basic types fulfills the promise of IFC with its over one thousand types and entities. Based on the current development status of our data model, there are several steps for future work of which some are listed:

1. Interface to a more sophisticated thermal simulation tool featuring as well CFD simulation
2. Automatic assessment of pressure losses of the air duct network as specified by 3D data
3. Allowing for building sophisticated control systems for the HVAC systems including lighting

Acknowledgements

Wiener Stadtwerke, TU Wien URBEM (https://urbem.tuwien.ac.at/home/EN/), Austrian Ministry for Transport, Innovation and Technology (bmvit) and Vienna International Airport have supported this work.

References

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

DIN-Normenausschuss Bauwesen (NABau). (2018) *DIN 276 Kosten im Bauwesen.* Beuth Verlag GmbH.

Ausschuss der Verbände und Kammern der Ingenieure und Architekten für die Honorarordnung e. V. (Commission on fees of the German Chamber of Engineers and Architects). (2013) *Honorarordnung für Architekten und Ingenieure (HOAI)*. Wolters Kluwer Deutschland GmbH.

Lee, S. H. and Lee, K. (2001) *Partial Entity Structure: A Compact Non-manifold Boundary Representation Based on Partial Topological Entities*. In Proceedings of the 6th ACM Symposium on Solid Modeling and Applications, Ann Arbor, Michigan, USA, pp. 159-170.

Paskaleva, G., Wolny, S. and Bednar T. (2018) *Big-open-real-BIM Data Model - Proof of Concept*. In Proceedings of the 7th International Building Physics Conference, IBPC2018, Syracuse, NY, USA, pp. 1083 - 1088.

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).

Schlueter, A. and Thesseling, F. (2009) *Building information model based energy/exergy performance assessment in early design stages*. Automation in Construction, Volume 18 (2), pp. 153-163.

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

Volk, R., Stengel, J. and Schultmann, F. (2014) *Building Information Modeling (BIM) for existing buildings — Literature review and future needs*. Automation in Construction, Volume 38, pp. 109-127.