PAPER • **OPEN ACCESS**

Development of Plugins for seamless Integration of the SIMULTAN Meta Data Model with IDA-ICE and RFEM 6

To cite this article: Bernhard Steiner *et al* 2023 *J. Phys.: Conf. Ser.* **2654** 012049

Development of Plugins for seamless Integration of the SIMULTAN Meta Data Model with IDA-ICE and RFEM 6

Bernhard Steiner[1], Andreas Sarkany[1], Zsombor Járosi[2], Galina Paskaleva[1], Thomas Bednar[1] and Christoph Bauer[3]

TU Wien, Karlsplatz 13, 1040 Vienna[1]
CouchFerret Kft., 1065 Budapest, Révay köz 4.[2]
Woschitzgroup, Karlsplatz 2/Top 19, 1010 Vienna[3]

E-mail: bernhard.steiner@tuwien.ac.at, andreas.sarkany@tuwien.ac.at, zsombor.jarosi@couchferret.com, galina.paskaleva@tuwien.ac.at, thomas.bednar@tuwien.ac.at, cb@woschitzgroup.com

Abstract. SIMULTAN ComponentBuilder plugins enable individualized development of connections to simulation tools, easing data management of the simulation-driven interdisciplinary planning processes in civil engineering. The Software Development Kit (SDK) is an integral part of ongoing research projects, which aim to exhibit and expand the capability of the SIMULTAN Meta Data Model. The goal is to store digital information necessary for the planning processes in civil engineering and operation and operational performance analysis of buildings via digital twins. A theoretical outline of the plugin structure is given, and the usability of the SDK is shown by presenting two example plugins. Two plugins were developed to enable seamless integration with the proprietary simulation software IDA ICE and RFEM 6 within the same data model. Furthermore, it is shown that the existing data in the SIMULTAN Data Model can be reused by utilizing already existing information and avoiding duplication. Limitations of the SDK will be explained, and concepts for future data model extensions will be discussed. The development and deployment of the two plugins show the benefits of using a Big-open-real-BIM Data Model for civil engineering planning processes, enabling individualization and workflow-enhancing expansion.

1. Introduction

Due to their complexity and intrinsic demand for flexibility, the interdisciplinary planning processes in civil engineering pose a challenge for the digital modelling and proper storage of all the information needed in different stages of the building process. [1] A proof of concept for solving this problem with the SIMULTAN Meta Data Model (MDM) for the evolution of a virtual building from design through construction into operation has been shown in [1, 2] and [3]. While an example of the seamless integration of specialised tools developed by domain experts has been given in [4], the coupling of proprietary software in the design phase is yet to be explored.

Open BIM standards like the Industry Foundation Classes (IFC) aim to include all the information needed for different stages of the planning process. However, each extension proposal has to undergo a lengthy development cycle before being incorporated into the data model. This

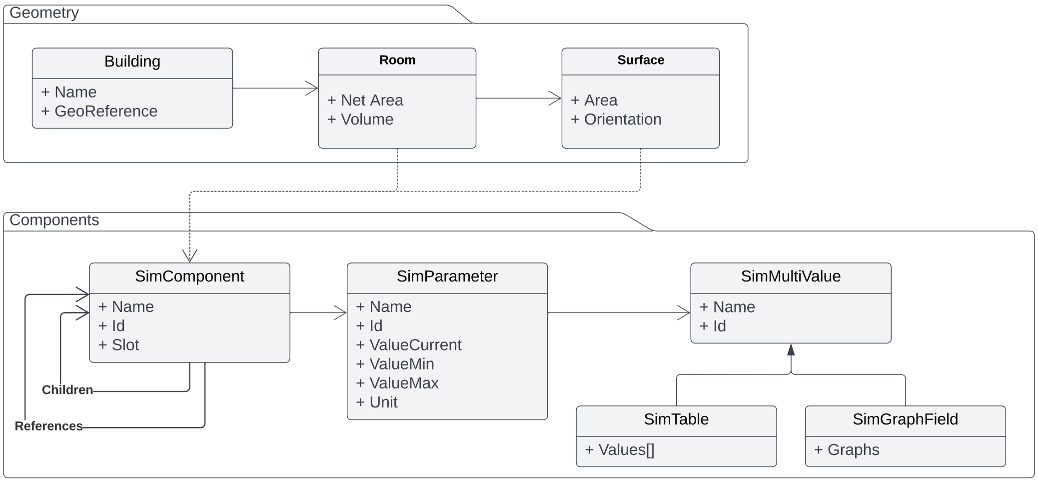

Figure 1. Core SIMULTAN classes as described in [2] and used by [4]. Components serve as hierarchical containers which contain Parameters and other Components. Parameters have a name and a value, which may be entered by the user or sourced from a MultiValue (Table or Graph). Components can reference other components and may be referenced from room geometry or surface geometry.

results in one of the main challenges IFC is facing: "keeping up with the updates of building regulations or with the ever-expanding state of the art in simulation tools." [1] The SIMULTAN MDM solves this issue by offering the domain experts domain-independent building blocks for the construction of custom domain-specific models. Since each model is built out of the same building blocks, communication within or across domains is as intuitive as communicating ideas in the same language. The primary building blocks of SIMULTAN, as shown in Figure 1, are components acting as meta-data containers for Parameters. Components also contain child Components and can reference Components from other top-level Components. Parameters contain a name and a value, either entered by a user or based on the data in a Table or Graph (implemented as MultiValues).

When focusing on practical applicability, all potentially available information is not always needed. In many cases, ensuring interoperability between in-house disciplines (e.g., building physics and structural analysis) would be a helpful first step. The domain-specific data model discussed in this work showcases the flexibility that allows SIMULTAN to be used as an underlying and connecting MDM. The construction of the domain-specific data model will be demonstrated by the development and implementation of a digital connection between the domains of building physics and structural analysis. This will enable us to run simulations (IDA ICE and RFEM 6) for both domains within the same data model (see Figure 2).

2. Domain Specific Data Model

The domain-specific data model enabling the export via the plugins is based on an already existing meta data model described in [2] to store information needed for building physics-related simulations. During the development of a new domain specific data model, the plugin developers together with the domain experts (civil engineers) have to specify the domain-specific data model based on the needs of the specific task. The definition is a formal agreement stating how data has to be laid out in the MDM to be supported by the plugins. The data model consists of geometrical information linked to physical properties and information that is purely

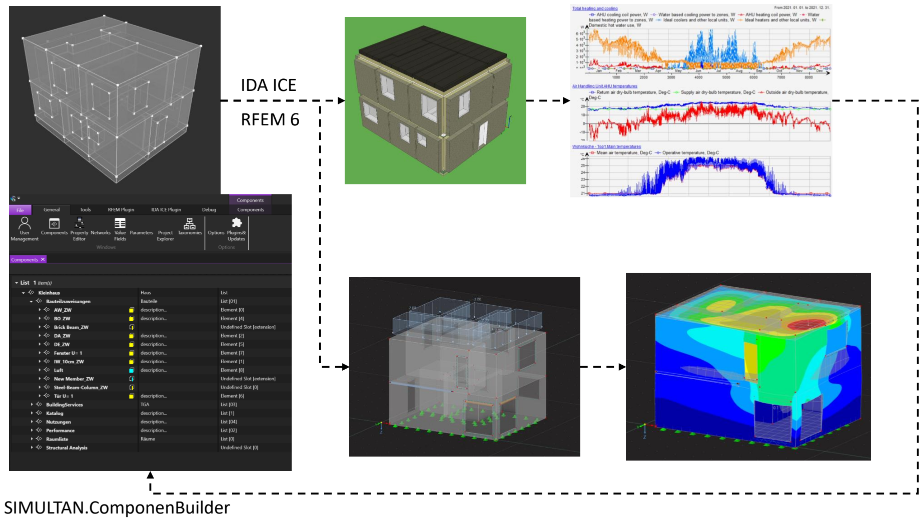

Figure 2. Depiction of the connections and the direction of data exchange between the SIMULTAN data model and simulation software (IDA ICE, RFEM 6)

used as input to describe technical or physical properties, e.g., parameters for HVAC systems (see Figure 3).

For now, entering data into the data model is done by using the *SIMULTAN.ComponentBuilder* which is a prototypical implementation of a User Interface to interact with the SIMULTAN data model. The SIMULTAN MDM reference implementation has already been made open source and may in future be integrated into other commercial CAD/BIM tools.

IDA ICE and RFEM 6 support various input file formats, but these import options proved insufficient for our purposes. Therefore, exporting the SIMULTAN data model to a common file format (such as IFC) and importing it into IDA ICE or RFEM 6 was not a feasible option. Instead of relying on the existing applications' importers, the plugins were developed to export the data stored in SIMULTAN directly to the IDA ICE or RFEM 6 native file format, respectively. This workflow ensures that all the necessary details of the model are retained and that the subsequent analysis is performed on the correct data.

2.1. Data model adaptations for IDA ICE

The existing domain-specific data model, shown in Figure 3, already contains provisions for data relevant to building physics simulations. Therefore, our goal was to preserve as many elements from this model as possible, and to only add elements for data that was not accounted for yet. As shown in Figure 4, existing data that could be used as input for the IDA ICE plugin was made available either via key parameters and/or via the component structure that was used to model the data. The information required by IDA ICE is shown in green.

For any type of IDA ICE-specific data, the flexibility of SIMULTAN makes it possible to add elements to the existing data model with the exact structure necessary to hold that data. Elements added through the course of this project are denoted by an asterisk ($*$) after their name.

Since most of the pre-existing elements represent building physics concepts, many material properties were already present. Missing parameters for the IDA ICE simulation, such as the solar transmittance coefficient τ_e for transparent materials, were easily added. Additionally,

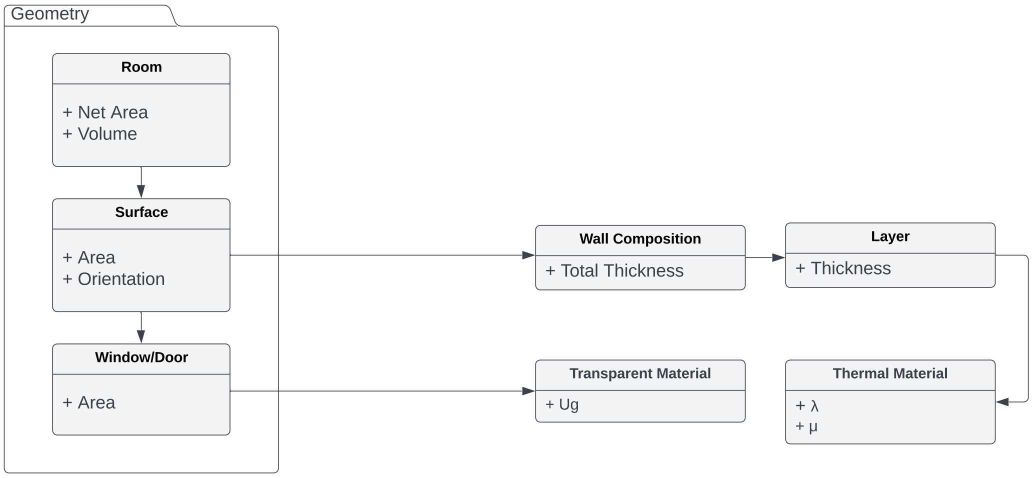

Figure 3. Excerpt of the SIMULTAN data model as agreed upon in [2] before the development of the plugins. This data model was expanded and modified to enable the implementation of the plugins. Geometric Objects (on the left) are linked to components (on the right) which represent the wall composition. Wall compositions store a sequence of material layers, each referencing a material with thermal, hygric and other physical properties.

occupant behaviour and relevant equipment were added to the rooms as components with parameters which can refer to an optional schedule. Information added to the data model for the export can either be numerical (numbers, graphs, matrices, etc.) or text-based. This is sufficient for most numerical simulation tools and therefore is only limited by the information the connected tools accept. The export of the HVAC system has been implemented in a simplified manner taking advantage of the ESBO-plant interface in IDA-ICE. A more complex export of the control-system could be handled by the data model since it adheres to above described limitations of the MDM and will be part of our future work.

The results created by the entire building hygrothermal simulation can be recorded into the data model. For each Room, the Operative Temperature, the Heating Load and the Cooling Load are imported as graphs. The graphs belonging to each room are grouped and linked to it via a Room Results component. In the same way, results for the entire Building (total heating load and total cooling load) are imported and linked to it via a Building Results component. A subset of the resulting data model is shown in the top-left part of Figure 4. The imported results also support numerical and text-based data.

2.2. Data model adaptations for RFEM 6

Based on the data model adapted for IDA ICE, the data model was additionally enriched by the elements required to hold the data for structural analysis in RFEM 6. Already existing data was again reused, avoiding duplication and the unnecessary cluttering of the data model.

Data model elements relevant to structural simulations are denoted in blue in Figure 4. Again, the added parameters and components are marked with an asterisk symbol ($*$) after the name. As can be seen, the general model structure of walls and wall layers could be reused, while other elements, such as the RFEM custom material identifiers and cross sections as well as the loads, had to be added. Since the SIMULTAN MDM only supported attaching components to Rooms (3D) and Surfaces (2D), it had to be adapted to allow for the specification of Beams (1D) in the geometry and for components that can be attached to those beams.

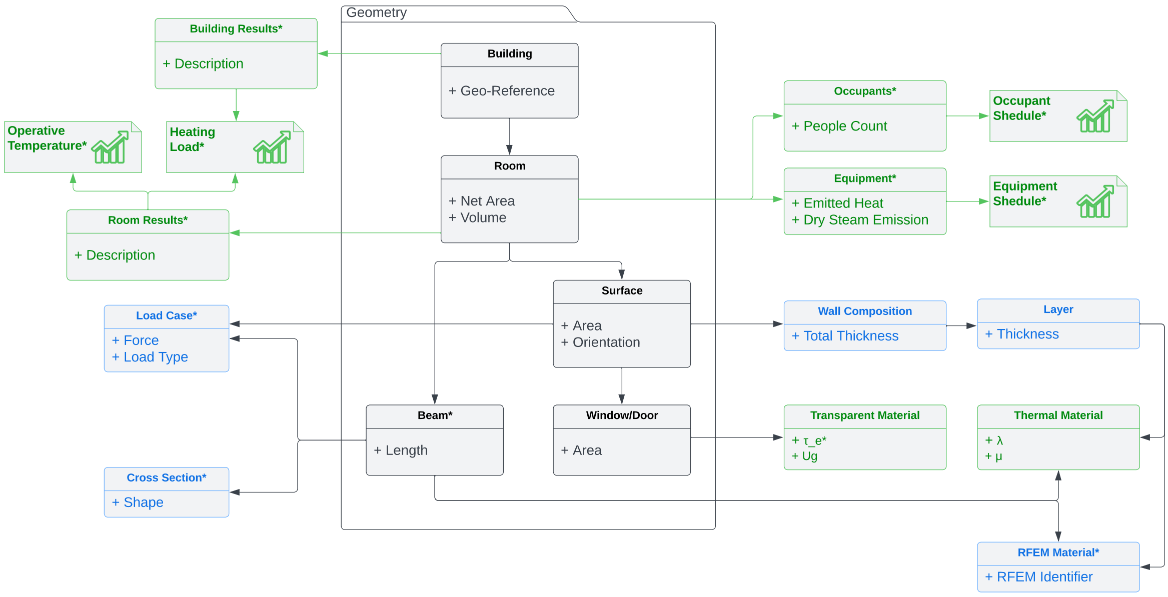

Figure 4. Excerpt of the domain-specific data model as defined in this project. Blocks denote components (except in package Geometry), while the content of blocks denotes parameters. MultiValues (Tables and Graphs) are shown as documents. Extensions to the previous data model are marked with an asterisk ($*$) after the name. IDA ICE specific parts are shown in green, while RFEM-related data is coloured in blue.

Due to the flexibility of the MDM, which in turn enabled the adaptations to the domain-specific data model, both the data for the structural analysis in RFEM and for the building physics simulation in IDA ICE can be stored in the same instance model simultaneously without conflicts.

3. Implementation

The prototypical implementation of the domain-specific data model described in Section 2 is specified and enforced by plugins for the SIMULTAN editing software *SIMULTAN.ComponentBuilder*. Plugins allow developers to connect their tools with the authoring software, providing seamless integration with the data model and user interfaces. Using the SIMULTAN **S**oftware **D**evelopment **K**it (SDK), a set of libraries and templates, developers can write their own code which interacts with the data model, while a user is editing the project data in a familiar environment.

3.1. Data transfer to IDA ICE

The goal was to provide a way to run IDA ICE simulations on existing data models built on top of the SIMULTAN MDM. This was achieved through the conversion of those models into C# representations of the IDA ICE data model and by serializing the converted data into the native IDA ICE *idm* file format.

The IDA ICE text-based file format, *idm*, is application specific and has LISP-like structure. In order to read and write this format, a custom parser was necessary. With the help of an ANTLR-generated lexer and parser[5][1], a C# class representation was created, which mirrors

[1] ANother Tool for Language Recognition

the data in the *idm* file format. The resulting *idm* file can then be opened in IDA ICE, just like any model saved from within the IDA ICE application itself. Subsequently, the user can run simulations on the data, and simulation results can be saved to that same *idm* file.

After the simulation is complete, the plugin reads the text-based *prn* files containing the results and imports these results into the SIMULTAN data model as graphs. The specialised components in the domain-specific data model (see Room Result and Building Result in Figure 4) are created by the plugin and are populated with data from the *prn* files. Through these steps, the plugin successfully provides a way to run IDA ICE simulations on an existing SIMULTAN data model and transfer the results back into SIMULTAN.

3.2. Data transfer to RFEM 6

Similarly to IDA ICE, existing SIMULTAN data models had to be transferred to RFEM 6 in order to run structural analysis on them. The process for transferring the SIMULTAN data model to RFEM 6 is similar to that for IDA ICE, except that parsing the RFEM data format did not require a custom parser implementation since RFEM 6 supports *xml* as its text-based data format.

The structure of the *xml* file is parsed into C# classes. This step is necessary since the RFEM data structure might change between versions, and dynamic parsing into classes allows for easier coupling. The RFEM classes are then populated with the data from the SIMULTAN data model, converting the data into the format expected by RFEM 6. The resulting objects are then serialized back into the *xml* file format and can be opened and run in RFEM 6.

Currently, no data is read back from the RFEM simulation since any further analysis does not require the results, but adding this functionality could be achieved similarly to that for IDA ICE (see Section 3.1).

3.3. Data format conversion

When transferring data between data formats, certain conversions will be necessary due to the differences in how those formats represent information. During the development of our exporters, we encountered multiple instances in which the conversion between data formats required not only the restructuring of information but also the conversion of the information itself. For example, there are significant differences in the representation of openings in IDA ICE and in the SIMULTAN geometry model. Specifically, IDA ICE stores the vertices of each opening relative to the 2-dimensional coordinate system of the surface that contains it, as shown in Figure 5 (on the right). Conversely, SIMULTAN stores the vertices of openings consistently in world coordinates, which necessitates the use of geometric transformations.

During these conversions, special care has to be taken in edge cases. We identified one such scenario during testing involving horizontal roof surfaces. In IDA ICE, the openings in such surfaces are defined relative to the world coordinate system, as seen in Figure 5 (on the left). This scenario had to be handled separately to ensure the accurate export of the openings.

4. Findings and Discussion

To show the capability of the *SIMULTAN.ComponentBuilder* plugins, a structural analysis and a whole building hygrothermal analysis were conducted. The test case is a simple residential building, shown in Figure 2, and it is investigated if the Meta Data Model (MDM) allows the storage of all the information needed for the simulations. Furthermore, it is investigated whether plugins can transfer data to the target applications without loss or distortion.

For the purpose of this paper, the whole data is retransferred to the simulation tool for every iteration to ensure that the most up-to-date data is used. The intermediate data in the simulation software may be stored for documentation purposes, but is not reused in a new simulation. In future, and especially with simulations that require additional input and settings

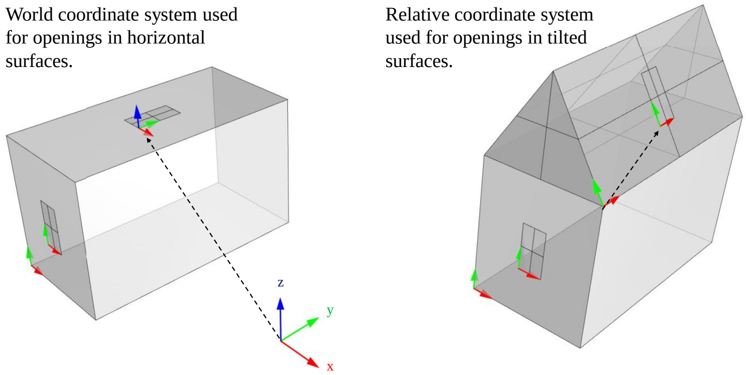

Figure 5. In IDA ICE, openings in a horizontal surface are defined relative to the world coordinate system, while openings in a tilted surface, relative to the 2D coordinate system of the surface itself.

in the simulation tool, it would be interesting to explore how existing simulation data could be merged and update with changes of the data in the shared data model.

4.1. Discussion - Testcase

The test case was set up as an instance of the domain-specific data model adapted to the target applications, as shown in Section 2. For each of the analyses, a separate geometry model was created within that same domain-specific model. This separation of the geometrical data according to discipline provides domain-specific focus and avoids cluttered and hard to edit federated geometry models. When working with either plugin, the data necessary to run the simulations was exported successfully. The structural analysis and the whole building hygrothermal simulation were carried out as expected and the results of the hygrothermal simulations were successfully imported back into the model for each simulation zone, enabling further analyses based on that data.

As described in Section 3.3, several special cases could be identified and solved. Throughout the development process, an effort has been made to identify and handle as many edge cases as possible. However, as the plugins are at a prototypical stage, some unexpected behaviour in not yet tested scenarios may occur.

4.2. Findings and Future Work

The flexibility of the SIMULTAN MDM allows the user to react easily to updates in any connected simulation software and to incorporate simulations early in the planning process. However, as of now, there is no automatic system in place that can translate between domain-specific data models. In consequence, disobeying constraints imposed by the data model of the newest version of a proprietary simulation tool may result in unexpected errors. Furthermore, the creation of the adapted domain-specific data model relies on the precision of the user.

This was partially solved by adding specialized user interfaces which allow the editing of components and parameters without changing critical naming or structure. Overall, this resulted in significantly fewer errors during the modelling process. On the other hand, the specialized user interfaces reduce the flexibility to store any data required without relying on software developers to implement it.

A more general approach would be the usage of taxonomies and templates. Taxonomies are a set of pre-defined terms the user can pick from and which could be provided by the plugins. Templates would allow to relax constraints about the component/parameter structure of the domain-specific data model, especially when several plugins require the same data but a different data structure. A plugin could provide a template which describes the required data elements for a domain-specific data model. By defining a mapping between the data structure and the template, the data structure can be decoupled from the data semantics.

5. Conclusions and Further Research

Two plugins for the *SIMULTAN.ComponentBuilder* to enable simulations with RFEM 6 and IDA ICE were developed. Both plugins rely on a common domain-specific data model conforming to the SIMULTAN MDM. The plugins allow interdisciplinary collaboration, whole-building hygrothermal simulations and structural analysis, within the same data model as a proof-of-concept. Furthermore, it could be shown that the SIMULTAN MDM provides a rich enough language for the definition of domain-specific models capable of storing all data required to conduct those simulations. The results of the whole building hygrothermal simulation could also be stored in the resulting data model, ensuring that no loss of information occurs between analyses. For now, the structural analysis results are not stored in the data model. The development of templates to avoid the manual generation of domain-specific models, as well as the incorporation of a bidirectional connection to the structural analysis, will be part of our future work. The SIMULTAN MDM is available as open source [2].

Acknowledgements

The results presented in this paper are part of the FFG research project "Ganzheitliche Gebäudesimulation". The authors want to thank the financial support provided by the project funding that made this research possible.

Author Credit Statement

All authors declare that they have no conflicts of interest.
Bernhard Steiner. Conceptualization, Writing, Visualization, Reviewing, Editing, Data modelling, Plugin development Supervision, Expertise Computer Science.
Andreas Sarkany. Conceptualization, Writing, Visualization, Reviewing, Editing, Data Modelling, Plugin development Supervision, Expertise Civil Engineering.
Zsombor Járosi Writing, Plugin development, Expertise Software Engineering.
Galina Paskaleva. Writing, Reviewing, Expertise Data Modelling.
Thomas Bednar. Reviewing, Data Modelling and Supervision.
Christoph Bauer. Reviewing.

References

[1] Paskaleva G, Lewis T, Wolny S, Steiner B and Bednar T 2019 SIMULTAN as a big-open-real-BIM data model - proof of concept for the design phase *21st CIB World Building Congress. Constructing Smart Cities* p 10
[2] Paskaleva G, Wolny S and Bednar T 2018 Big-open-real-BIM data model - proof of concept *IBPC2018 - Healthy, Intelligent and Resilient Buildings and Urban Environments* pp 1083–1088
[3] Paskaleva G, Lewis T, Wolny S and Bednar T 2019 SIMULTAN as a big-open-real-BIM data model - evolution of virtual building from design through construction into operation phase *21st CIB World Building Congress. Constructing Smart Cities* p 11
[4] Bühler M, Steiner B and Bednar T 2022 **1101** 082015 ISSN 1755-1315
[5] Parr T 2013 *The Definitive ANTLR 4 Reference* (The Pragmatic Bookshelf)

[2] https://github.com/bph-tuwien/SIMULTAN