Visualisation and graph-based storage of customised changes in early design phases

D. Napps[a], A. Zahedi[b], M. König[a] and F. Petzold[b]

[a]Chair of Computing in Engineering, Ruhr University Bochum, Germany
[b]Chair of Architectural Informatics, Technical University of Munich, Germany
E-mail: daniel.napps@ruhr-uni-bochum.de, ata.zahedi@tum.de, markus.koenig@ruhr-uni-bochum.de, petzold@tum.de

Abstract

Building Information Modeling (BIM) was conceived as a working method for networked planning using software assistance. Multiple stakeholders in the construction industry are involved in the design and management of digital representations of the physical and functional characteristics of a facility in the BIM model.

Design variant management coupled with documentation and recording of design knowledge are the main goals of this paper. The use of a BIM-based plugin for design rationale documentation grantees support for other architects in the early design phases, to retrieve suitable building designs and ideas, based on their needs and architectural concepts. Uniting Design Episodes with different Variant Types allows the architects to receive inspiration and freely select important parameters for their design of the building in advance. These input parameters are compared with other BIM-based design graphs, following the graph representation approach.

The relevant design options and variants corresponding to the new project are visualised and stored for both current and future users. This can lead to cost and time savings and allow the possibility for future extensibility in terms of other analysis parameters (such as the level of detail or energy efficiency).

Keywords –

Building Information Modeling (BIM); Industry Foundation Classes (IFC); Early Design Phases; Design Decisions; Graph Representation; Design Episodes; Variant Types; Plugin

1 Introduction

1.1 The problem statement

In the early design phases of buildings, the cooperation of interdisciplinary experts with architects and civil engineers is essential for considering the complex aims and requirements of a building. The architect attempts to find an efficient and aesthetic solution for the future project with the given specifications (e.g., sustainability) and evaluates a variety of designs for a building, which can result in a significant component of the project phase which is, on the one hand, only temporary but. on the other hand, can incur high expenses. These designs must be subsequently continuously coordinated and adapted with the other project participants [1]. In order to avoid the architects from designing basic concepts, such as floorplans, construction types or individual elements for each upcoming building, it is essential to provide a high-performance efficient solution to support design reuse.

Currently many architects are using Building Information Modeling (BIM) because of its benefits for exchanging models with other stakeholders. In particular, BIM includes all important information and visualisations of the building [2]. Nevertheless, for the different requirements and comparisons, several BIM models are made during the architecture process in order to evaluate the advantages and disadvantages of the various designs of the building [3]. In addition, the requirements for a new project may be vague or detailed, depending on the components of the construction, which in turn complicates the architect's work. An insufficient knowledge base for integrating previous experience is especially challenging for architects who have not been in practice for years [4], as they have either no or only limited experience with similar projects. Inadequate documentation of the early design process and an absence of transparency (in case of subjective estimations) can result in mistakes [5], when adapting the processes of previous architects to a new project, and is thus problematic even for experienced architects.

1.2 Previous research and proposed solution

During the past years of research, a Variant Type concept was developed, which allows individual elements in a BIM model to be classified. A classification is made between three Variant Types, namely structural, functional and product variants [6]. In addition, a graph representation of the variants was established. This representation is based on the structure of the Industry Foundation Classes (IFC) according to IFC4 [7].

Recently, a concept for the retrieval process of similar building designs in the early design phase of a building was presented that incorporates this diversity of variants. This process is part of the case-based reasoning approach, where problems are solved using problems that have already been solved and proven. The findings provide an example of how an architect can select the most suitable graph for the upcoming building project from a graph database [8].

Design decisions are based on both objective and subjective criteria. Proper documentation of the design process demands recording qualitative and subjective assessments and decisions. To address this, Zahedi et al. [9] introduced the concept of Explanation Tags to describe and elaborate the subjective aspects of design decisions and enhance the design documentation and its future readability and reuse. Furthermore, Zahedi et. al. [9] presented the concept of Design Episodes to store various pieces and sections of design in corresponding segmentations that enable the traceability and reuse of these design ideas and concepts to address future design tasks and problems.

Graph structures are useful in various disciplines to analyse and extract information. In the construction industry, they are used in the context of BIM because of their capacity to describe complex digital models and internal relationships [10].

1.3 Objectives and scope

The research presented in this paper aims to identify a possibility to make the architect's work more comprehensible in the early design phases of a building. In particular, the use of a plugin that can be implemented in a BIM-enabled environment which captures the knowledge of previous architects, as well as subjective design decisions related to existing buildings.

The objective in using the tool is to capture design requirements for the building and to store and manage the information, changes and options using IFC. Focusing on the following two areas is of particular importance:

- Retrieval of individual Design Episodes and Variant Types that match the selected input parameters in order to search for inspiration regarding a new project.
- Ensure consistency of the entire design or individual Design Episodes through pattern matching of the individual BIM model graphs.

An efficient way to apply this graph-based method has its origins in the field of mathematics and has already been successfully used in the building sector as well [11].

The graph representation of BIM data models is generated based on the IFC data structure. Existing building graphs are compared with the design graphs described from the input parameters in terms of similarity and thus provide the architect with relevant solutions that give inspiration coming out of an existing case. As a result, the scope of this concept is particularly flexible, as the scale of view can be set in such a way that both entire buildings and individual design episodes, right down to variant types, can be compared using the plugin.

2 Background

In the following, the background information on which the methodological approach is based and which contributes to the understanding of the case study is specified. The importance of BIM and the IFC interface is briefly outlined.

The comprehensive digital representation of a building information model is widely used in the construction industry. In contrast to the outdated analogue form of building drafts on paper, the use of BIM allows, among other things, the capture of three-dimensional geometries, non-physical elements and semantic information. In particular, it enables the stakeholders involved in the project to use and exchange the same, consistent, up-to-date knowledge, and it covers the entire life cycle of a building (design, planning, construction, management, refurbishment) [2].

Industry Foundation Classes as an international standard (ISO 16739) is important for the exchange of the BIM data. It is an object-based file format developed and continuously updated by buildingSMART. IFC is described as an entity-relationship model that can represent all entities of a building that are organised and interconnected in an object-based inheritance hierarchy [12]. The IFC structure is relevant in this context because the graph representation of the BIM model is based on its internal architecture.

2.1 Graph representation

Existing graph-based publications for structured data can be divided into two principal categories. The first category covers Resource Description Framework (RDF) graph models. The second includes property graph models (PGM) [13].

The RDF approach is often used related to linked data and declares graphs as a triple (subject, predicate, object), see figure 1.

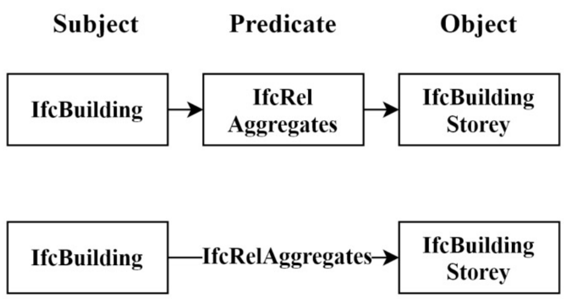

Figure 1. RDF representation of IFC Entities

Because of the Uniform Resource Identifier (URI), the graphs are unique but do not have an internal structure. As used in the construction industry when comparing entire BIM models with each other, the file size becomes problematic when exporting complex buildings [14].

In parallel the (labelled) property graph (LPG) or entity centred graph approach is used for object-oriented programming and overcomes the limitations of the triple centred approach. In each edge and node, information, attributes or properties can be stored (Figure 2).

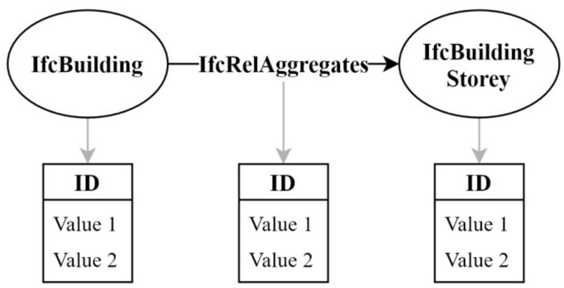

Figure 2. LPG representation of IFC Entities

Zhao et al. recently presented an approach, in which they described a method for the intact representation of relationships between IFC entities as an essential requirement for the correct classification of IFC entities [15]. Differences to the RFD approach are that LPGs are characterised by an internal, more reduced structure. Instances of relationships of the same type can also be identified so that they can be qualified or attributed [16].

Using graph pattern matching, several graphs or subgraphs can be compared with each other.

2.2 Variant management and pattern matching

This work is intended to contribute to the previous knowledge in this topic, where a BIM-based solution for the retrieval process in the early design phase of a building was identified [6]. Searching for similar floor plans currently operates on the basis of filtering with respect to size, type of use, number of rooms and other parameters. A combination of Variant Types and defined Design Episodes has not been made. Structure variants offer options to the structure of the building, for example the geometry of the elements. Functional variants refer to the functions, e.g. a room layout or structural engineering elements. The product variants include individual objects (e.g. windows) that can be exchanged with similar objects or other property values.

Due to a similarity calculation, the upcoming project is matched with a similar building from a database of existing designs to provide inspiration for architects in early design phases [6]. Graph pattern matching is used for the matching method. This method has been widely accepted in understanding and accessing network data and works for both graph representations. Searching for a similar graph in a graph database, graph pattern matching finds the answers A as a set of graphs from D [17].

Based on a sample query, matching graphs are selected from a database that provide an answer to the existing query. There are two approaches for this process: exact pattern matching and inexact pattern matching [8]. The first type is characterised by a one-to-one isomorphism of two examined graphs. In the second approach, similar subgraphs are examined in the overall graph.

2.3 Explanation Tags and Design Episodes

To explain and elaborate on certain aspects of design, an open-ended collection of Explanation Tags is offered within the plugin that cover many design criteria and concepts and may be used to assign to various design elements or their specific properties to justify and clarify the rationale behind those details and decisions. A particular focus has been on sustainability requirements in planning competitions, while designing the first set of Explanation Tags [18].

Certain parts of design can be documented and stored as Design Episodes. Each Design Episode in the plugin consists of multiple design elements, e.g., building components or spaces, and a name and description explaining this design chapter or situation. A

Design Episode can contain a solution for a design problem or a template to address a design case or situation [18].

3 Methodology

3.1 Definition of property graphs

A labelled property graph consists of nodes or vertices (V), relationships, properties and labels. Relationships, that can also be described as edges (E), connect nodes and structure the graph, by directing from a start point to end point. Both, nodes and relationships can have properties. Labels (L) group nodes together and specify their role within the dataset [19].

Resulting from this, a graph can be described mathematically [20].

$G = (V, E)$ with $V = (v_1, v_2, v_3, \dots, v_i)$ and $E = (e_1, e_2, e_3, \dots, e_i)$

Generated subgraphs can be either labelled or attributed.

$G = (V, E, L)$ with $L_v = (l_{v_1}, l_{v_2}, l_{v_3}, \dots, l_{v_i})$ and $L_e = (l_{e_1}, l_{e_2}, l_{e_3}, \dots, l_{e_i})$

The importance of the vertices and relations can be determined by weighting all elements of the respective set.

$W = (v) \forall v \in \mathrm{V}$ and $W = (e) \forall e$

Definitions below refer to the elements described:

- Vertices (V) of the graph represent the IFC entities of the BIM model. All entities in a digital model (e.g. IfcBuilding or IfcWall) can be considered as virtual physical objects and differ from each other. The label of the vertex, stored in the nodes, is the entity's type.
- An edge (E) of a graph is a relation, i.e. an interaction between two entities. To distinguish between the two directions a relationship may have, we define so-called directed graph edges.
- Properties represent features of entities and relationships. In a LPG this information is stored in the vertices and edges.

3.2 Plugin

Zahedi and Petzold [18] developed a plugin for design documentation, that provides two main capabilities for the users. The first is to use Explanation Tags to enhance and elaborate on the argumentation and rationale of design decisions while also providing the ability to explore and find various assigned tags in any given BIM model and to manipulate and change them if need be. Furthermore, is the ability to extend and add to the open-end collection of tags based on users' specific needs, while being able to import or export custom-designed tags as an archive. The second capability is the ability to create and manipulate Design Episodes. This plugin is designed for Autodesk Revit.

Furthermore, the plugin enables the export of Design Episodes to a graph database such as Neo4j. When exporting to Neo4j, the user has the choice to select the desired properties of various elements of a Design Episode to be exported or not. The result is a LPG that contains the Design Episode as the parent node, and Episode Elements, as child nodes connected via Episode Element edges. The Design Episode nodes have, as attributes

- a name and a description to explain the purpose of this episode using storytelling techniques,
- a model identifier for tracing the original BIM model that this episode originated from,
- and a Global Unique Identifier (GUID) to help distinguish between Design Episodes with similar names.

The design element nodes have, as attributes, an ID, a Unique ID, and an Object Name as the basis, as well as every other property from Revit that the user has decided to be exported to Neo4j. When the elements in Revit are directly connected to each other, the resulting design element nodes are also connected with an IsConnected edge. When an element is contained in another one in Revit, the corresponding design element nodes in Neo4j are connected via a ContainedIn edge. When the elements in Revit are adjacent to each other, the IsAdjacent edge is used to connect the exported element nodes in Neo4j.

3.3 Matching

Graph pattern matching and pattern construction are important tasks in the comparison of two graph representations. Based on the definition of property graphs, graphs assume characteristics from each other. Comparing graphs requires that the graphs are connected. Thus, a graph always consists of nodes that are connected by edges. (Figure 3).

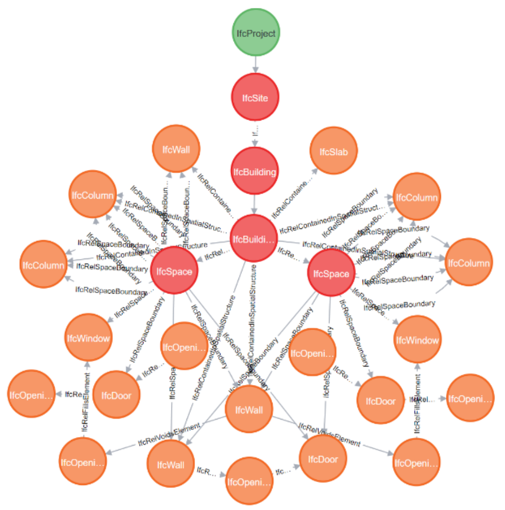

Figure 3. Graph representation of the IFC structure

The inexact pattern matching, mentioned at the beginning, accommodates the heterogeneity of the building industry, because two buildings are never exactly similar.

Using a pattern query, the substructures of a graph G can be analysed for matches using a query function. An inexact graph pattern match $I_{gpm}(G)$ of a pattern query I in a data graph G is a subgraph G* of G. Both vertices V and edges E of the graph G are analysed during the matching process 18].

The subgraphs that are searched for can be simplistically found as the result of the following command:

Find all matches in $G = (V, E, L)$ with a given query $I = (V_I, E_I, L_I)$.

Combining Design Episodes and Variant Types enables querying a subgraph matching of different building graphs in various ways. This approach enables searching directly for defined Design Episodes or for individual optional Variant Types for specific rooms or storeys.

4 Case study

In the following, the two approaches of Zahedi et al. [9] and Napps et al. [8] are combined. In terms of further research, this results in a potential filtering for finding building designs. Furthermore, architects will be provided with a tool to define Variant Types and Design Episodes in a BIM model to save them transparently and comprehensible for other users. In order to illustrate this process, a sample Revit project file is used provided by Autodesk [21]. Three BIM models are created, each with different functional and structural variants that belong to a defined Design Episode. Focussing only on the two Variant Types mentioned is due to the fact that the process for the product variant is comparable to that of the functional variant.

4.1 Define Variant Types and Design Episodes

The introduced plugin is used to define an exemplary Design Episode in all three BIM models. For this purpose, the Explanation Tags are used to define specific attributes of a Design Episode. The assignment is based on the subjective decisions of the architect.

Components within this episode receive additional Explanation Tags, defined by the architect in order to create a design concept. To add the variants to the model, the plugin is first extended by the three new Explanation Tags, each representing a Variant Type (Figure 4).

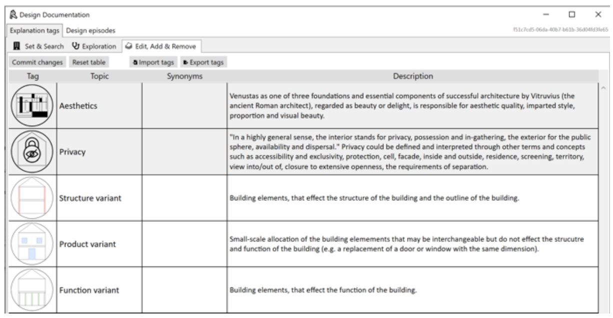

Figure 4. Creating new Explanation Tags for Variant Types

During the next phase, different variants are implemented in the three models and identified with new tags. These identifications are saved, can be retrieved for further tasks and are available for other users.

Each Variant Type is identified by its colour (Figure 5-7). Included in the Design Episode are the dining room and the kitchen, bathroom and mechanical room. Individual elements that are located in this Design Episode are provided with tags, so that many options are generated.

The function variant changes from a wall with an open doorway separating the kitchen from the dining area (model one) to a glass wall including a door (model two) and in case of model three, to a load-bearing column without a separating effect. Equally, the structure variant is changed and thus varies between a glass curtain wall, an energy-efficient wall and an extension of the room structure.

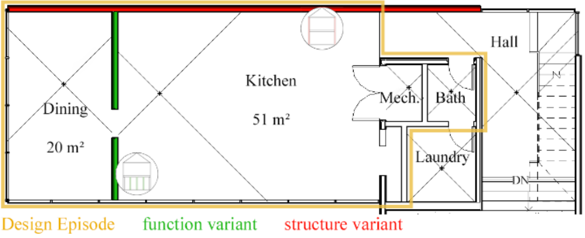

Figure 5. Defining Variant Types in the first BIM model

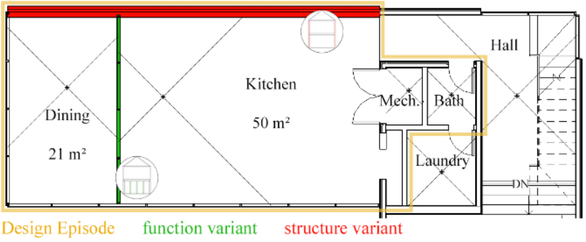

Figure 6. Defining Variant Types in the second BIM model

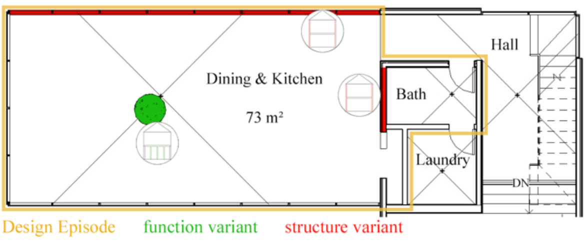

Figure 7. Defining Variant Types in the third BIM model

The arguments and justification for each variant case, whether it be functional or structural in the examples above, could be discussed and saved as an episode description in the Revit plugin.

Choosing a glass wall over a regular wall, for example, will provide openness and transparency, whereas the other choice will provide privacy.

4.2 Transfer BIM model to graphical representation

Neo4j is used to realise the graph-based representation of the BIM model. Exporting the entire BIM model as an IFC4 file is possible in order to import the data into Neo4j, but not necessary, as the plugin offers a more efficient way. Using this interface, the Design Episodes are displayed correctly. The user is able to directly export Design Episodes to Neo4j, since this feature is implemented as part of the plugin.

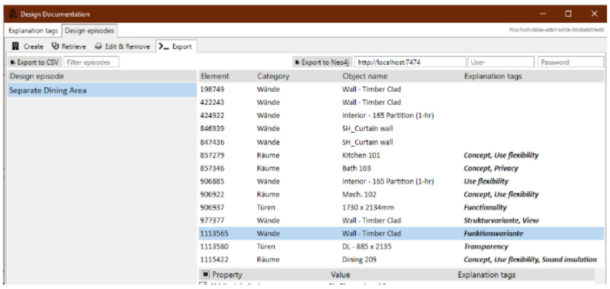

Figure 8. Set and export Design Episodes

Converting the model into a graph network allows an analysis of the relationships between the individual properties of a complex model. This is an effective feature for exporting subgraphs from the entire BIM model because the Design Episodes are part of the entire building. For a graphical comparison of the Design Episode, it is exported from all three BIM models, which ensures different Variant Types in the subgraphs. By the end of this process, three different subgraphs will have been loaded into Neo4j.

4.3 Visualisation and application

After a Design Episode with different Explanation Tags has been created and exported (Figure 8), the architect is able to choose possible matching variants, stored in a database, based on different requirements. Continuing, the Design Episode can be visualised by its graphical structure.

Users of the plugin are either able to create variants themselves, that contain other functions in the Design Episode or they can retrieve variants for this type of Design Episode that have already been stored in the database.

Within this case study, the basic assumption is that there are three different options stored in the database for a similar Design Episode.

In the case the architect requires a separation of the kitchen and the diner in terms of noise and smell and additionally a panoramic view to the outside, the Variant Types of model one are the most suitable alternative for the project.

The vertices of the graph that are a function variant are marked with a green circle. The nodes that are a structural variant are circled in red. The Explanation Tags can also be listed in the names of the nodes, but in this case it was decided to represent the entities in order to reflect the structure of the model.

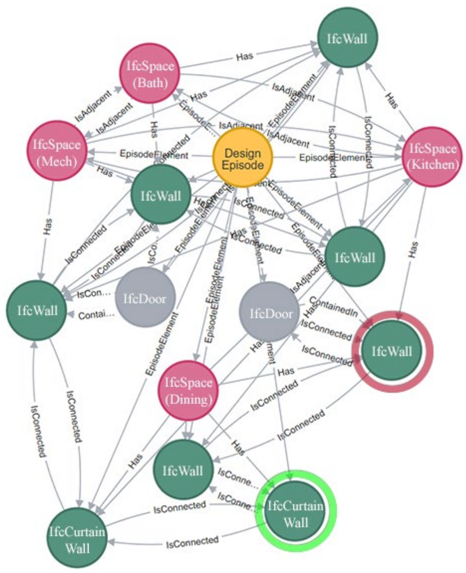

Figure 9. Graph representation of BIM model 1

If it is important for the client to have a separation between the kitchen and the diner, but desires a visual connection between the rooms and would like to build as energy-efficiently as possible, then the Variant Types from model two correspond to the ideas.

Figure 10. Graph representation of BIM model 2

A request for a shared kitchen and dining area can be supplemented by static adjustments (e.g. a column). Other room modifications that affect the structure can also be found, for example if a larger bathroom is desired and another room is only optional (Figure 7).

Adjustments made via the variant management can have large or small effects on the graph structure (Figure 9-11).

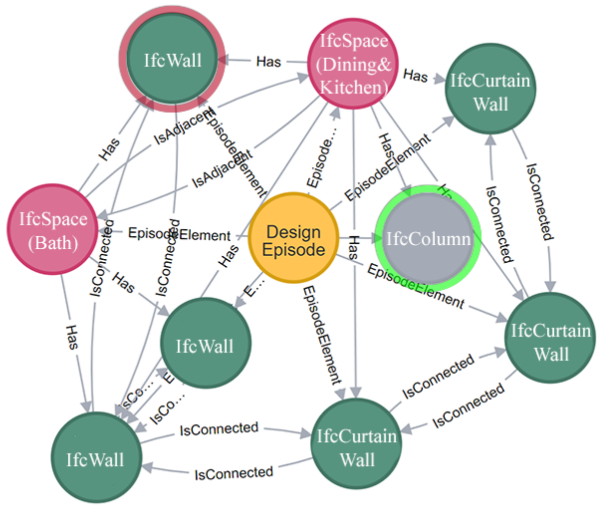

Figure 11. Graph representation of BIM model 3

There is a difference between the three graphical representations of the digital models. In a design selection process by a user, only the most relevant Design Episodes are selected according to the input conditions for the project or Design Episode.

Several Variant Types for an entity can be grouped into option nodes, which makes it easier for the user, as he or she is directly shown possible options on the basis of the Explanation Tags made for a Design Episode.

5 Conclusion and future work

In this paper a way for documenting design decisions by using the announced Revit plugin in the early design phases was introduced to retrieve suitable building designs and ideas, based on architectural needs and concepts. This workflow is based on the traceability of earlier design decisions, leading to time and cost savings for future similar projects.

Users of the plugin may specify a Design Episode, which can be assigned with various Explanation Tags. This results in important design decisions for the architect, which can be entered and stored directly in a BIM model. Based on these design decisions, architects are able to define Variant Types in the model. It has been demonstrated that depending on the Explanation Tags, different Design Episodes can be shown to the architect. This inspiration can be refined by different variations without changing the basic settings of the Design Episode, if these are stored in an option node.

Pattern matching allows similar design variants to be retrieved from a database that match the client's requirements. In addition, options from a variant pool are displayed, allowing the architect to make decisions based on the design decisions at the beginning of the project. Thus, the user is inspired by different options, which fit the requirements of the new building or the defined Design Episode.

It is to be investigated whether the graphically simplified relationship descriptions are sufficient for more complex applications in the construction industry. Alternatively, the relationship notation of the IFC structure can be used. This is important for the similarity calculations in the retrieval process of the CBR approach.

The benefit of the tool for architects is limited by the voluntary use and implementation of the tool in Revit. A large overall benefit will only arise if many architects use the plugin regularly and save the designs in a database for other users. In addition, future work might take an approach that permits an automatic assignment of Variant Types, so that the user's effort is limited. Following on from this, the integration of additional area-specific topics such as structural design, detailing and energy efficiency is to be evaluated in order to optimise the model, as well as an appropriate visualisation of the design changes.

References

[1] Darwich, A. K., Means, J. K. and Lawrence, T., ASHRAE GreenGuide. Design, construction and operation of sustainable buildings, Eds. 2018. ASHRAE, Atlanta, GA, 2018.

[2] Borrmann, A., König, M., Koch C. and Beetz, J., Building Information Modeling. Technology Foundations and Industry Practice, Eds. 2018. Springer, Cham, Switzerland, 2018.

[3] MITRE, System Engineering Guide. 2014.

[4] Demian, P. and Fruchter, R. An ethnographic study of design knowledge reuse in the architecture, engineering, and construction industry. Research in Engineering Design, 16: 184-195, 2006.

[5] Zima, K., Plebankiewicz, E., Wieczorek, D. A SWOT Analysis of the Use of BIM Technology in the Polish Construction Industry. Buildings, 10: 16-28, 2020.

[6] Mattern, H. and König M. Concepts for formal modeling and management of building design options. In Proceedings of the ICCET, Seattle, Washington, 2017.

[7] Mattern, H. and König, M. BIM-based modeling and management of design options at early planning phases. Advanced Engineering Informatics, 38: 316-329, 2018.

[8] Napps, D., Pawlowski, D. and König M. BIM-based variant retrieval of building designs using case-based reasoning and pattern matching. In Proceedings of the ISARC, pages 435-442, Dubai, United Arab Emirates, 2021.

[9] Zahedi, A., Abualdenien, J., Petzold, F., Borrmann, A. Documenting Design Decisions using Design Episodes, Explanation Tags and Constraints. Manuscript submitted for publication, 2021.

[10] Isaac, S., Sadeghpour, F. and Navon, R. Analyzing Building Information Using Graph Theory. In Proceedings of the ISARC, pages 1013-1020, Montreal, Canada, 2013.

[11] Wu, J. and Zhang, J. New Automated BIM Object Classification Method to Support BIM Interoperability. Journal of Computing in Civil Engineering, 33(5): 1-55, 2019.

[12] ISO. ISO 16739-1: 2018. Online: https://www.iso.org/standard/70303.html. Accessed: 24.01.2022.

[13] Hor, A. H., Jadidi, A. and Sohn G. BIM-GIS integrated geospatial information model using semantic web and RDF graphs. In Proceedings of the ISPRS, pages 73-79, Prague, Czech Republic, 2016.

[14] Cao, H. and Connolly, D. Delta: an ontology for the distribution of differences between RDF graphs. In Proceedings of the W3, New York City, USA, 2004.

[15] Zhao, Q., Li, Y., Hei, X. and Yang, M. A graph-based method for IFC data merging, Advances in Civil Engineering, 2020: 1-15, 2020.

[16] Barrasa, J. and Howard, R. RDF triple stores vs. labeled property graphs: What's the difference? Online: https://neo4j.com/blog/rdf-triple-store-vs-labeled-property-graph-difference/. Accessed: 10.01.2020.

[17] Wu, Y. and Khan, A. A graph pattern matching. In Sakr, S., Zomaya, A. (Publisher) Enzyclopedia of big data technologies, 1-5, 2020.

[18] Zahedi, A., Petzold, F. Revit Add-In For Documenting Design Decisions and Rationale. A BIM-based tool to capture tacit design knowledge and support its reuse. In Proceedings of the CAADRIA 2022, pages 557-566, Sydney, Australia, 2022.

[19] Robinson, I. Graph databases. New opportunities for connected data, Eds. 2015. O'Reilly, Beijing, 2015.

[20] Kriege, N. and Mutzel, P. Subgraph matching kernels for attributed graphs. In Proceedings of the ICML, pages 291-298, Edinburgh, Scotland, 2012.

[21] Autodesk. Revit sample project files. Architecture. Online:https://knowledge.autodesk.com/support/revit/getting-started/caas/CloudHelp/cloudhelp/2022/ENU/Revit-GetStarted/files/GUID-61EF2F22-3A1F-4317-B925-1E85F138BE88-htm.html. Accessed: 03.01.2022.