BIM-based Variant Retrieval of Building Designs Using Case-based Reasoning and Pattern Matching

D. Napps, D. Pawlowski and M. König

Chair of Computing in Engineering, Ruhr University Bochum, Germany
E-mail: daniel.napps@ruhr-uni-bochum.de, dennis.pawlowski@ruhr-uni-bochum.de, markus.koenig@ruhr-uni-bochum.de

Abstract

Architects today still very often create traditional 2d plans in the early planning phase, which form the basis for discussions with investors, clients and engineers. The designs are often difficult to understand for non-experts and design variants can only be compared with great effort. The use of BIM models simplifies both communication and the actual creation through the use of parametric modeling approaches. Even though a building project always has very individual characteristics, it makes sense to take experience from other projects as a basis for the design. Therefore, it may be that suitable floor plans of similar buildings already exist, which are a good basis for variant studies in early design phases. A challenge is to be able to access the experience of other architects. Currently, this information is not available and is difficult to transfer from one project to the next. This paper outlines a solution for a BIM-based variant retrieval of a building design in early design phases using Case Based Reasoning (CBR) and Pattern Matching (PM). On the one hand, this offers the possibility to learn from old mistakes, and on the other hand, it enables the quick selection of suitable variants for a building from a diverse pool of variants.

Keywords –

Building information Modeling (BIM), case-based reasoning (CBR), Early Design Phases, Industry Foundation Classes (IFC), Pattern Matching (PM), Variant Retrieval, Similarity

1 Introduction

In the planning phase, referencing building concepts is a common practice, for example to find and evaluate similar building concepts and use them for a new project. Such manual search can take an unacceptable amount of time. An automatic generation of suitable designs or variants promises a faster workflow and at the same time serves as a transparent assessment of the building project and various analytical parameters, which is not insignificant for the consideration of the variants themselves [14]. Variant matching always allows a suitable thematic and technical view (construction costs, sustainability, etc.) due to the attributes stored in the Industry Foundation Class (IFC) interfaces, as the effects of the change in parameters and components are directly visible in the efficiency of the building. In order to find variants, the case-based reasoning method can be useful [7]. This approach starts with a problem (search query) that correlates with the specific ideas about the building project and the concepts of the planners. This paper presents a solution for finding suitable variants in the early design phases of buildings, using the case-based reasoning (CBR) approach based on the IFC standard. For this purpose, specific case studies are provided to explain the similarity calculation used, which is essential for the first phase of the CBR cycle. Building on the basic idea of CBR (similarity recognition) the method of pattern matching (graphical pattern recognition) is used to illustrate the selection of suitable variants with given framework conditions (search queries).

In previous research, a concept for modelling and managing design options was already created, which is connected to the BIM models. Three variant classes (structural, functional and product variants) were defined and implemented using the IFC [11]. This procedure avoids redundancy and offers a wide range of applications. Furthermore, a representation of variants in using graph theory was introduced [10].

The basic idea of finding suitable variants of floor plans already appeared in 1996 in a journal article by Gomez de Silva Garza and Maher [9]. At that time the technical possibilities regarding BIM models did not exist. Langenhan et al. [13], address this issue and attempt to match building floor plans using the fingerprint method and a database matching. BIM is not integrated here either. Similar to us, Althoff et al. [14] use the CBR approach with the help of exact and inexact graph matching to support the architect in finding similar 2d floor plans in the early design phases. The properties and consequences of components and structures cannot be identified. In the reviewed literature, there is no evidence of the correlation between BIM models and variant decision-making.

This paper is dealing with the topic in order to offer a different and evolved solution that integrates BIM. This paper addresses the issue to provide an evolved solution that integrates BIM models and uses IFC for the implementation. These factors are essential for interdisciplinary collaboration during the early design phases of buildings and are neglected by the authors mentioned above.

2 Background and methodology

The following chapters briefly summarize the basic theoretical knowledge and methodology on which the paper is based on.

2.1 BIM in context of the research topic

The acronym BIM stands for Building Information Modeling, which has been increasingly influencing the construction industry for several years. In a digital building model, the exact geometry of the building and information on all the built-in components are stored within the model and recorded over the entire life cycle of the building. Thus, this information can be used for (facility) management, revitalization, expansion or demolition. Additionally, a loss-free exchange of files with investors, energy and construction engineers or facility management is possible over the entire life cycle of the building [3].

The exchange of this (open) BIM data is made possible by the Industry Foundation Classes (IFC), an international open standard (ISO 16739) developed by buildingSMART [6]. The IFC is continuously being further developed since 2013 and is available in the latest version of the IFC standard as IFC4 [12].

BIM is compatible with our research work because the IFC interface can be used to define the information range of individual elements, directly show the changes of parameters and thus allow a comparison of parameters among themselves and thematically, regarding the whole design. Therefore, structure, function and product variants can be assigned to the respective level of detail (LOD). Three variant types have already been defined [11]. The structure variant offers options to the structure of the building, for example the geometry of the building or the number of building storeys. The functional variant can be exemplified by the load-bearing structure of the building or objects that fulfil the same use or purpose. The product variant, as the smallest level of the variant types, includes individual objects (e.g. doors) that can be exchanged with similar objects or other property values, without affecting the structure and function.

Using BIM and IFC makes it possible to capture different variants with the entire geometry and all properties. This means that they are available to everyone as explicit knowledge. For the research in this topic, a graph-based representation of the data structure is chosen [10]. In this, the BIM model is represented based on the IFC standard with the different entities, attributes and, consequently, the geometry. This continues the previous considerations in the research context and is suitable for the methodological approach of CBR and pattern match (PM). For this purpose, patterns must be defined, which in turn are linked to the similarity definitions and afterwards they can be shown as several options in a graph database.

2.2 Current problems

Currently, individual BIM models are being Currently, individual BIM models are being developed from scratch for each construction project. In this process, the planners are guided by the wishes of the client. The planner therefore designs a great many different floor plan variants for a wide variety of building types and uses during his professional career. This experience is used when a planner encounters similar boundary conditions and construction requirements. However, since this procedure is related to the implicit knowledge of the planner, the number of variants is limited. With the presented solution the knowledge of each planner is available by storing different variants of building designs in a database and generating automated solutions.

2.3 Methodology and process

2.3.1 CBR

CBR is a methodical way of problem solving, using problems that have already been solved and proven to be successful. In the context of this research, it is useful to adapt a suitable solution from a case base, as a 1:1 transfer of the solution turns out to be difficult, because buildings are mostly heterogeneous. Thus there are almost never exact matches, since the external or internal circumstances are not the same even though the appearance may be the same. That is why it is necessary to determine a degree of similarity. The usefulness of a solution is consequently an optimization phase that is based on the similarity determination of the components and adapted to their specific weighting [7].

The process of case-based reasoning can be represented as a cycle, which Aamodt and Plaza [1] divide into four steps: retrieve, reuse, revise and retain. In this paper the retrieval process is dealt with and the problem is defined as finding a suitable variant. The procedure is demonstrated by own case studies and differs from MetisCBR [14] in that it includes both, a graph structure and geometry to make the search results more accurate.

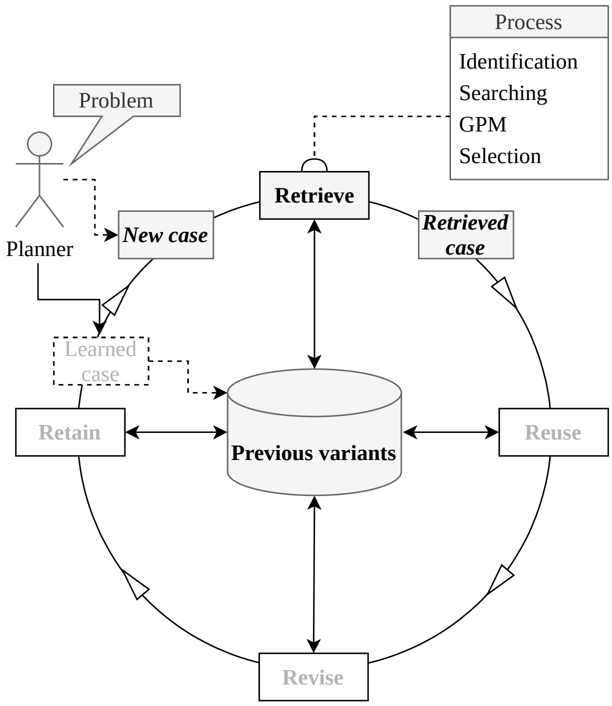

Figure 1. CBR cycle according to Aamodt & Plaza [14]

2.3.2 GPM

The method of graph pattern matching (GPM) is a possibility to find (suitable) variants from a graph database. Based on a pattern query, matching graphs are selected from a case base that provide an answer to the existing request (problem). This approach can also be extended to subgraphs if more specific requirements are being searched for. Similarity definition and weights are of great importance for the pattern matching in order to limit the results and have them target-oriented [15]. Furthermore, a categorization is made between two types of pattern matching (PM), which are called exact PM and inexact PM [14].

An exact PM is characterized by one-to-one isomorphism, i.e. the same number of nodes are contained in both graphs and both graphs have the same manner of connections (number of edges). If the graphs are not isomorph, subgraphs can be examined for isomorphism in the next step [14].

The inexact PM is a different approach that is more suitable to the heterogeneous field of the building sector, because building structures and room uses differ from each other. The search query is decomposed into different IFC data or building components and these are evaluated and normalized with a similarity score [14].

In addition, specific weights are added, which allows the similarities to be determined and the relevance for the comparison of variants to be specified. A variant is a match if the graph of the variant is a subgraph of the database entry. In contrast to the exact pattern match, so-called replacement rules are applicable to the inexact PM [14]. These replacement rules are important, because they extend the inexact pattern match similarity. For example, instead of a load-bearing wall, the user is able to choose a load-bearing column. or instead of a connection between two rooms with a door, a breakthrough is possible. These replacement rules must be individually adapted by the user.

2.3.3 Similarity

In the following, the similarity calculation according to Richard Hemming is determined, which is often used in the retrieval process of the CBR. The concept of generalized similarity determines the similarity of two cases for any number of attribute values, which can be weighted and normalized (1). According to Hemming, a qualitative and quantitative similarity determination is possible. Generalized similarity allows a comparison of normalized real attributes, as well as a comparison based on classifications [2].

$$\text{sim}\ (p, v) = \frac{\sum_{i=1}^{n} \omega_i sim_i(p_i, v_i)}{\sum_{i=1}^{n} \omega_i} \tag{1}$$

Using this formula, the similarity between a problem *p* and a possible variant *v*, with non-negative weights, is a value between zero and one. The more similar the variants, the higher the value of sim [12]. According to this approach, the similarities are used for all entities stored in the IFC data structure. Depending on the type of entities, similarity measures can be defined for both IFC quantities and property definitions.

2.3.4 Retrieval

In a database (case base), *n* variant graphs are defined, which differ from the simple consideration of the building graphs in that they contain option points. These variant graphs were defined as part of other projects and include structural, functional and product variants. Following the top-down design, the new user specifies the various parameters for the current project (problem) for which he wants to find a variant (e.g. use of the building floor, sizes, number and functions of the rooms). The process also allows searching for required connections (e.g. between the rooms). It is up to the user to decide whether this is necessary for the problem or not.

After that, the individual entities are weighted by the user, which results in priorities being selected depending on individual preferences and the use cases, generating different solutions in the end. The database is searched for graphs that are either an exact match or an inexact match of the graph of the problem. These matches are presented to the user and sorted by the magnitude of the calculated similarity. If the result still has a need for optimization, fine-tuning can be done. This is helpful to show possible further structure, function or product variants for a selected area if a match is identified. This might be the case if a room contains windows or doors as a result, which in turn can be replaced by other product variants. Another case could be chosen other wall structures, for example, to improve the energy properties of the building (structure variant) (Fig. 2).

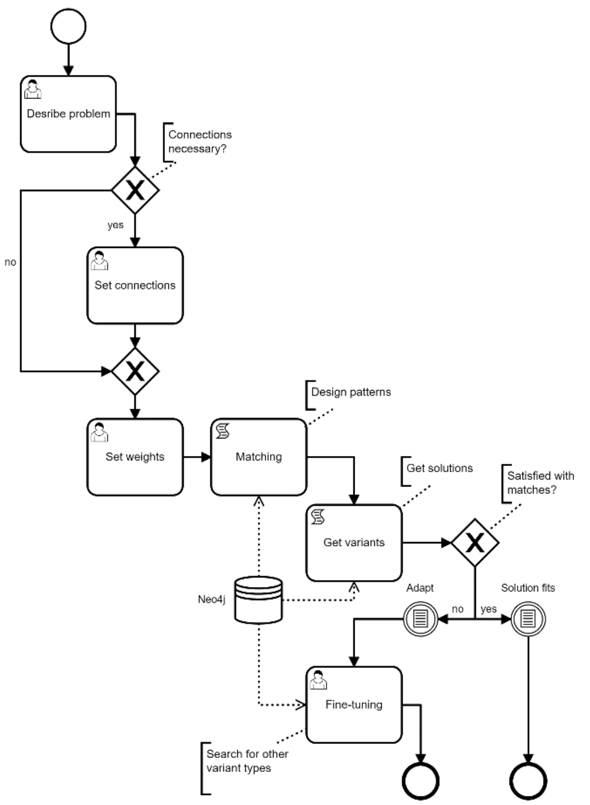

Figure 2. BPMN (Retrieve)

3 Case study

With the help of a simplified case study, the BIM-based variant retrieval is presented. In determining the weights, planners and architects usually act subjectively, stick to their experience, are oriented to the sustainability or adapt the wishes of the clients [4]. In this paper the authors assume the role of the user, which is why the problem description and weights are determined by them.

A variant is to be found for a ground floor with office use, an area of 290 m^2 and with at least two offices. Other criteria set by the user are, offices with windows, a connection of the offices with a corridor, the wall thicknesses and other attributes. According to this problem description, the floor plan could possibly look like the following which captures the problem visually, but is not part of the data base (Fig. 3).

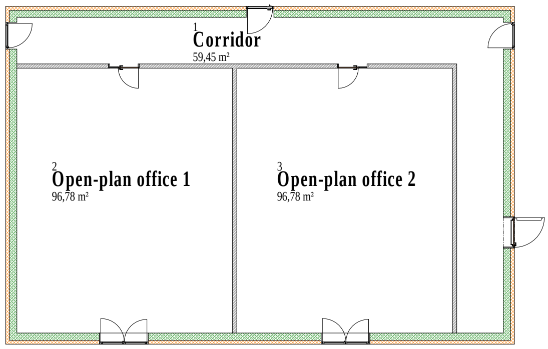

Figure 3. Exemplary floor plan according to problem description of the use

The comparison of the problem with different variants from a case base is based on the different attributes, stored in the BIM model. Table 1 lists the attributes essential for the case study, together with their defined specifications. Based on these attributes, the most similar problems in the current case base are determined for the defined issue.

Table 1. Attributes and specifications

Entities	IfcObject-Quantities	Property definitions
IfcBuilding-Storey	GrossFloorArea GrossVolume	-
IfcColumn	AverageHeight GrossFloorArea GrossVolume	LoadBearing
IfcDoor	OverallHeight OverallWidth	IsExternal FireBearing
IfcRelConnects	-	-
IfcSpace	GrossFloorArea GrossVolume	Category
IfcWall	NormalWidth	LoadBearing
IfcWindow	OverallHeight OverallWidth	IsExternal

The calculation of the similarity measure must be defined for each property. For this purpose, the individual properties are mapped to integers in order to be able to calculate normalized differences. The properties are given values to make strings and deviations measurable and to be able to measure the differences. At the end of the table there is a normalization factor (NF) to normalize the differences and receive a number between zero and one. A difference of zero corresponds to the similarity measure one, that means, that the attributes are equal to each other. A difference of one equates to the similarity measure zero, that means that the properties are not similar to each other.

Table 2-6. Values for similarity determination

Boolean	Value
True	0
False	1

Deviation area & volume	Value
0%	0
< 5%	1
5% up to < 10%	2
10% up to < 15%	3
15% up to < 20%	4
20% and above	5
Normalization factor (NF)	1/6

Deviation height & width	Value
0cm	0
< 5cm	1
5cm up to < 10cm	2
10cm up to < 20cm	3
20cm up to < 30cm	4
30cm and above	5
NF	1/6

Deviation NominalWidth	Value
0cm	0
< 1cm	1
1cm up to < 2cm	2
2cm up to < 3cm	3
3cm up to < 5cm	4
5cm up to < 7cm	5
7cm up to < 10cm	6
10cm up to < 20cm	7
20cm and above	8
NF	1/9

Category (DIN 277)	Value
NUF 1	0
NUF 2	1
NUF 3	2
NUF 4	3
NUF 5	4
NUF 6	5
NF	1/6

Weights must be added to the selected properties to capture the relevance of the properties and solve the problem. The list of weights and the calculation of the corresponding values are subjective estimations of the designer. These weights are selected in this example as follows (Tab. 7).

Table 7. Entity weighting

IfcData	Weights
IfcBuildingStorey	1,0
IfcColumn	0,2
IfcDoor	0,3
IfcRelConnects	0,5
IfcSpace	0,6
IfcWall	0,7
IfcWindow	0,3

After these steps, the similarity measure between the problem and two possible variants can be calculated. The example mentioned at the beginning is the problem (p) for which a solution is to be found that corresponds to the weights (ω) with the specific attributes (a) of the variants (v). The values of the attributes are calculated according to the following formula [2]:

$$\mathrm{sim}(p(a_1), v(a_1)) = \omega_1(1 - (NF_1(p(a_1) - v\,(a_1)))) \quad (2)$$

$$\mathrm{sim}(p(a_1), v_1(a_1)) = 1(1 - (\frac{1}{6}(0 - 0)))$$

$$\mathrm{sim}(p(a_1), v_1(a_1)) = 1$$

In this example (2), the similarity of the IfcBuildingStorey between the problem and variant one is shown. The calculation is made for each quantity and property of the entities (Tab. 1).

The similarities of the entities are calculated, with regards to the problem, and compared in the following table (Tab. 8), with the characteristics of the individual attributes. In this process, the inputs requested by the user are compared on the basis of their IFC data with already solved problems from the case base.

Table 8. Similarity

Entities	Variant 1	Variant 2
IfcBuildingStorey	1	1
IfcColumn	0,2	0
IfcDoor	0,1	0,3
IfcRelConnects	0,5	0,4
IfcSpace	0,3	0,6
IfcWall	0,25	0,5
IfcWindow	0,15	0,3

The floor plans of the two variants are compared pictorially in order to show the differences to the problem

(Fig. 4 and 5).

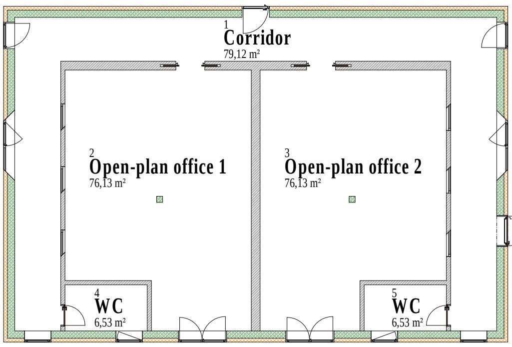

Figure 4. Building floor plan representation variant one

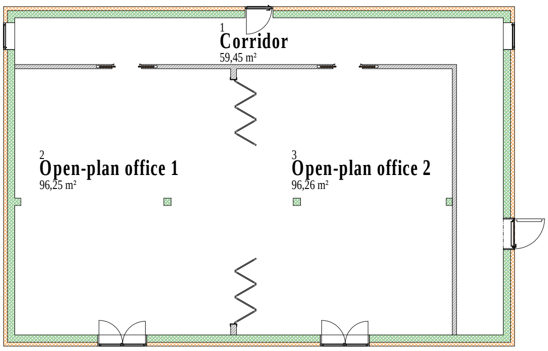

Figure 5. Building floor plan representation variant two

Compared to figure 3, the two possible solutions have differences that are more or less significant.

However, instead of the image representation, the actual process matches the IFC data structure of different projects from the case base with the existing problem. The graph of the variants one and two are inexact pattern matches out of the IFC structure of the problem definition. The representation of the floor plans is thus only representative in order to have a better look at the differences, which is more difficult to see in the graphical representation.

In figure 6, a simplified representation has been chosen to show the graphical structure of the problem. The individual properties of the data are stored in the nodes and edges.

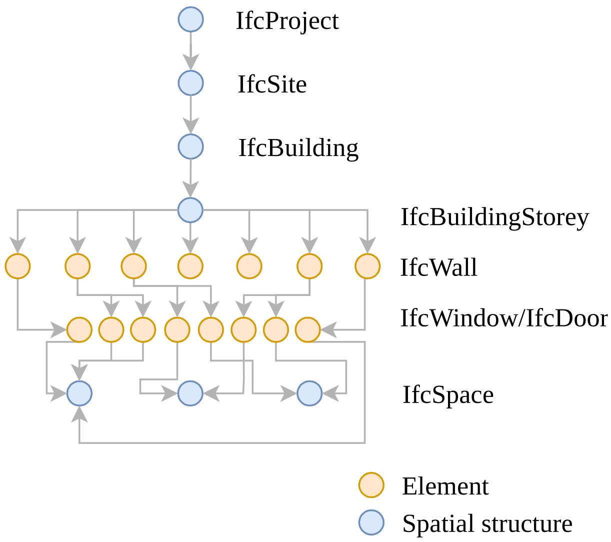

Figure 6. IFC structure of the problem definition

According to the similarity calculation, a similarity of variant one (x) to the problem (p) is calculated as follows:

$$\mathrm{sim}(p,x) = \frac{1+0{,}2+0{,}1+0{,}5+0{,}3+0{,}25+{,}015}{1+0{,}2+0{,}3+0{,}5+0{,}6+0{,}7+0{,}3} \qquad (3)$$
$$\mathrm{sim}(p,x) = 0{,}694$$

For variant two (y), the similarity is calculated in the same way.

$$\mathrm{sim}(p,y) = \frac{1+0+0{,}3+0{,}4+0{,}6+0{,}5+0{,}3}{1+0{,}2+0{,}3+0{,}5+0{,}6+0{,}7+0{,}3} \qquad (4)$$
$$\mathrm{sim}(p,x) = 0{,}86$$

In this case, it indicates that variant two has a greater similarity to the problem and thus results as the more suitable solution regarding this issue.

If the user is not completely satisfied with this solution, he or she can search for possible alternatives to the variant types in the fine-tuning section. An example of a procedure in this area is illustrated in figure 7.

```
MATCH (c:IfcElement)-->(d:IfcElement)<--(e:IfcSpatialStructureElement), (d:IfcElement)<--(f:IfcSpatialStructureElement)
WHERE c.Entity="IfcOpeningElement" AND 0.19<=c.Depth<=0.21 AND 0.8<=c.Height<=0.9 AND 1.8<=c.Width<=2.3
    AND d.Entity="IfcDoor" AND d.`Operation Type`="SINGLE_SWING_RIGHT" AND d.IsExternal=false
    AND e.Entity="IfcSpace" AND e.`Room key Usage type`="1 Living"
    AND  f.Entity="IfcSpace" AND f.`Room key Usage type`="2 Office work"
    AND size( (d)-[:IfcRelSpaceBoundary]-() ) >= 2
RETURN c,d,e,f,ID(d)
```

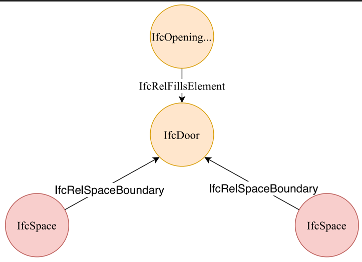

Figure 7. Optional product variant (PM)

In the given case the user is able to find an optional product variant (door) out of Neo4j based on specific deviations. By using the given cypher query, the database is searched for a door that fits into a given wall opening (depth, height, width). The object is an interior swinging door that opens to the right and connects an office use with a living use. Exact or inexact pattern matches are searched in the database that match the selected variant for which there is an option for the variant type.

Matching alternative variants are returned and the user can choose a suitable option or expand the search. This process can be adapted for each product variant and extended to functional and structural variants if required.

4 Conclusion and future work

With the help of CBR, existing knowledge can be used to find solutions for similar problems. The similarity measures can be flexibly adapted to the individual requirements of the user.

In this paper, a BIM-based solution for the retrieval process in the early design phase of a building is identified. This makes it possible to include all the information stored in a BIM model in the similarity measurement using IFC. The paper points out optimization tendencies for the current research, which will be identified and dealt with in the current research. Based on this, the other processes of the CBR move into the focus of the upcoming research.

CBR is based on the fact that there are enough variants in the graph data pool. In the presented example the number of variants is quite limited, but sufficient to show the retrieval process of finding a suitable variant for a simplified given problem. Machine learning can be used in the future research to fill up the variant pool and generate a higher number of possible matches.

Following on this the increase in complexity of buildings and their structures will be addressed in the next phase. Studies indicate that that a high number of nodes, relationships and properties does not cause any difficulties for the graph database and thus the utilisation of the CPU and query speeds are only dependent on the implemented hardware [5]. Therefore, CPU utilisation must be tested under various complex conditions. Complementary Grossniklaus et al. [8] highlight the advantages of a graph database (e.g. Neo4j) in relation to a large amount of data, as it allows high data correlation between entities, an efficient query performance and powerful graph-based algorithm.

The exemplary weights of the entities can be extended to the individual quantities and properties in further studies in order to be able to provide even more targeted results. In addition to that, extensions can be made according to the value, based on other parameters, such as fire protection, u-value or energy efficiency.

In addition to this, the similarity calculation can also be extended to provide more accurate results and needs to be implemented so that an automated similarity matching algorithm can be used. The shown calculation can possibly be supplemented by other established similarity calculations.

5 References

[1] Aamodt, A. and Plaza, E. 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7, 1, 39–59.

[2] Beierle, C. and Kern-Isberner, G. 2019. *Methoden wissensbasierter Systeme. Grundlagen, Algorithmen, Anwendungen*. Lehrbuch. Springer Vieweg, Wiesbaden, Heidelberg.

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

[4] Brown, J., Azhar, S., and Farooqui, R. 2009. BIM-based Sustainability Analysis: An Evaluation of Building Performance Analysis Software. *International Journal of Construction Education and Research* 2009.

[5] H. Huang and Z. Dong. 2013. Research on architecture and query performance based on distributed graph database Neo4j. In *2013 3rd International Conference on Consumer Electronics, Communications and Networks*, 533–536. DOI=10.1109/CECNet.2013.6703387.

[6] ISO. 2018. *ISO 16739-1:2018. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries - Part 1: Data schema*. https://www.iso.org/standard/70303.html. Accessed 19 July 2021.

[7] Jiang, X., Wang, S., Wang, J., Lyu, S., and Skitmore, M. 2020. A Decision Method for Construction Safety Risk Management Based on

Ontology and Improved CBR: Example of a Subway Project. *International journal of environmental research and public health* 17, 11.

[8] Jürgen Hölsch, Tobias Schmidt, and M. Grossniklaus. 2017. On the Performance of Analytical and Pattern Matching Graph Queries in Neo4j and a Relational Database. *EDBT/ICDT*.

[9] Maher, M. L. and Gomez de Silva Garza, A. 1997. Case-based reasoning in design. *IEEE Expert* 12, 2, 34–41.

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

[11] Mattern, H. and König, M. 06222017. Concepts for Formal Modeling and Management of Building Design Options. In *Computing in Civil Engineering 2017*. American Society of Civil Engineers, Reston, VA, 59–66. DOI=10.1061/9780784480823.008.

[12] Muneer, A. IFC4 - the new buildingSMART Standard. What's new in IFC4? 2013.

[13] Roith, J., Langenhan, C., and Petzold, F. 2017. Supporting the building design process with graph-based methods using centrally coordinated federated databases. *Vis. in Eng.* 5, 1.

[14] Sabri, Q. U., Bayer, J., Ayzenshtadt, V., Saqib Bukhari, S., Althoff, K.-D., and Dengel, A. 2/24/2017 - 2/26/2017. Semantic Pattern-based Retrieval of Architectural Floor Plans with Case-based and Graph-based Searching Techniques and their Evaluation and Visualization. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*. SCITEPRESS - Science and Technology Publications, 50–60. DOI=10.5220/0006112800500060.

[15] Wu, Y. and Khan, A. 2020. Graph Pattern Matching. In *Encyclopedia of Big Data Technologies*, S. Sakr and A. Zomaya, Eds. Springer International Publishing, Cham, 1–5. DOI=10.1007/978-3-319-63962-8_74-1.