REVIT ADD-IN FOR DOCUMENTING DESIGN DECISIONS AND RATIONALE

A BIM-based tool to capture tacit design knowledge and support its reuse

ATA ZAHEDI[1] and FRANK PETZOLD[2]
[1,2] Chair of Architectural Informatics, TUM School of Engineering and Design, Technical University of Munich.
[1]ata.zahedi@tum.de, 0000-0003-0296-9205
[2]petzold@tum.de, 0000-0001-8974-0926

Abstract. The building design is a problem-solving process in which the demands of the customer are turned into a design job. Throughout the design process, several specialists from various fields cooperate to develop the final design. The key deliverables that are exchanged and passed over are the generated designs. However, no explanation of design intentions or evidence of argumentations behind the design decisions is included in the design deliverables. Design documentation helps with communication and coordination between designers and other domain experts and stakeholders. Proper design documentation also preserves and protects corporate design knowledge and makes it available for future design problems and projects. This paper presents the implementation of an add-in for Revit that features the concepts of so-called Explanation Tags and Design Episodes to digitally explain and document design rationale and decisions, and ultimately capture the tacit design knowledge. As an extra feature, the add-in also enables the export of design episodes to a Graph-Database (Neo4j) as well as CSV files for future reference and use. Subsequently, the goal is to search and query inside the graph database for similar designs that can solve and answer the new design tasks in the future. This paper contributes to the UN Sustainable Development Goals, namely SDG 9 & SDG 11 among others.

Keywords. Design Documentation; Explanation Tags; Design Episodes; Building Information Modelling (BIM); Design Rationale; SDG 9; SDG 11.

1. Introduction

Designing a building is a problem-solving activity in which the clients' wishes and needs are translated into a design task taking into account the various regulations and building codes. The design task and its solutions co-evolve throughout the design process while all parties are trying to fulfill numerous requirements and guidelines

POST-CARBON, Proceedings of the 27th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2022, Volume 2, 557-566. © 2022 and published by the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong.

(Zeiler et al., 2007). Furthermore, during the project's life-cycle, multiple experts from multiple domains collaborate in developing the different partial models, including architectural, structural, and HVAC, among others (Cao & Protzen, 1999). Accordingly, explicitly communicating the rationale behind design decisions is crucial for developing regulatory compliant designs that also fit the owner's needs. In everyday practice, documentation of the architectural design process is not mandatory. Sometimes design variants are documented textually and graphically as an intermediate step, but usually, only the final result is documented graphically, e.g. in digital models or drawings and elevations, as well as textually. However, developing designs that meet the various requirements and regulations is a complex task, especially since building designs are composed of various elements that are connected by topological relationships and functional interdependencies. Therefore, documenting the design rationale and intent, while incorporating the individual building elements, facilitates the investigation and evaluation of the design that meets these regulations. Recording and sharing explanations of decisions made during the design process will enhance mutual understanding and collaboration among all parties and professionals involved in the design process and promote the cross-organizational distribution and reuse of shared tacit knowledge and designs for other future projects and tasks.

This paper addresses the problem of capturing tacit design knowledge and documenting design decisions and rationale. To this end, a novel approach is introduced by Zahedi et. al. (Zahedi et. al., 2021) based on BIM methodology to link customer requirements and building codes to design concepts and ideas, and to record and document design decisions, argumentation, and rationale in a way that is transparent to all stakeholders. Consequently, this paper contributes to the UN Sustainable Development Goals, in particular 'Industry, Innovation and Infrastructure' and 'Sustainable Cities and Communities'.

2. Literature Review

Storytelling is one of the methods for imparting architectural knowledge that has been studied in the literature (Heylighen, Martin, & Cavallin, 2007; Martin et al., 2003, 2005). It is recognized as a means of accelerating comprehension. Storytelling, at the same time, allows for numerous important challenges to be addressed in terms of architectural complexity. In addition to being straightforward, easy to read, and amusing, stories that recognize the deep relationships of things are unforgettable. As a result, narrative allows for a rich and compact means of conveying complexity in a short amount of time. The "Building Stories" project (Heylighen, Martin, & Cavallin, 2007; Martin et al., 2003, 2005), has demonstrated the effectiveness of storytelling in collecting and storing tacit design knowledge. Thus, the concept of design episodes introduced by Zahedi et. al. (Zahedi et. al., 2021) utilizes storytelling techniques to encapsulate and capture different bits and pieces of design in various episodes.

In architectural building design, the usage of references is regarded as a recognized strategy (Gänshirt, 2012) for assisting design, testing ideas, defining design parameters, and demonstrating new suggestions and possibilities. The built and planned models act as an architectural knowledge repository, containing both spatial forms and solutions for specific architectural expressions. At the same time, graph structures are widely used in various disciplines for analyzing and extracting information and knowledge, as well as in the BIM context, because of their capacity to describe complicated relationships (Isaac et al., 2013). In the AEC sector, graph structures have been employed for a variety of applications, such as path planning (Hamieh et al., 2020), retrieval of comparable designs (Ayzenshtadt et al., 2018; Langenhan et al., 2013), modeling and management of BIM-based design variants (Mattern & König, 2018). That is why the Revit add-in discussed in this work, also enables the export of the design episodes to a graph structure. Abualdenien & Borrmann categorized the graph representations in the AEC industry into four groups (Abualdenien & Borrmann, 2021); space connectivity graphs (He et al., 2018; Langenhan et al., 2013), navigation graphs (Al Hattab & Hamzeh, 2018; Dubey et al., 2020), IFC model graphs (Exner et al., 2019; Ismail et al., 2018), Knowledge representation graphs (Hor et al., 2018). The graph typology presented in this paper for the export of design episodes is broadly based on the Parametric Building Graph that was introduced by Abualdenien & Borrmann (Abualdenien & Borrmann, 2021) for capturing and storing the detailing patterns.

In Case-Based Reasoning (CBR), qualitative evaluations have been proposed as a mapping mechanism for recording design scenarios. Episodic case-based design is a type of design that represents individual design circumstances using design episodes that match certain design elements' configurations (Maher et al., 1995). Accordingly, the concept of design episodes (Zahedi et. al., 2021) follows the same principle in breaking down the whole final design into various episodes that can be traced and used in the future to address similar recurring problems and tasks. Zahedi et. al. (Zahedi et. al., 2021) proposed a way for digitally recording tacit design knowledge and decisions while also taking into account their aim and subjective evaluations. Explanation tags and spatial and semantic restrictions are assigned to specific model elements and/or their attributes to express architectural concepts and subjective assessments and explanations. By applying Design Episodes and storytelling techniques, various sections and parts of a design are digitally documented and Natural Language Processing (NLP) is used to facilitate querying owner requirements and regulatory documents (Zahedi et. al., 2021). Neuckermans et al. (Neuckermans et al., 2002) and Heylighen et al. (Heylighen, Neuckermans, et al., 2007) took a graphical approach, designing and prototyping "visual keys" for visually indexing complete design cases. They also used this approach as an access mechanism in the context of their case-based design (CBD) tool called DYNAMO (Dynamic Architectural Memory Online).

3. Concept and Methodology

There are numerous implicit design considerations and domain-specific experiences hidden in every building design. Figure 1 shows the design process and its abstraction levels projected by our concept and methodology. As shown in this figure, every building project must adhere to a variety of owner specifications, rules, and boundary constraints. Next are the architectural principles and concepts, which are selected and used on the concept level to meet the clients' needs and represent them in the design. The chosen concepts are subsequently implemented by modeling and describing the individual pieces, including geometric and semantic information, topological connections, and functional dependencies (on the design level). This paper utilizes the notions of design episodes and explanation tags that play a role in documenting design decisions on the concept level. These concepts will be discussed based on another manuscript submitted for publication by Zahedi et. al. (Zahedi et. al., 2021).

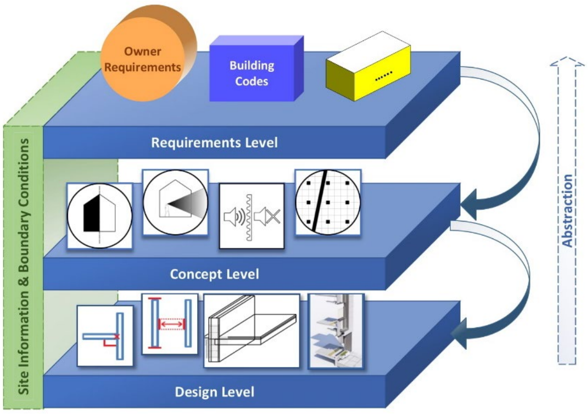


Figure 1. The design process is depicted using abstraction layers (Abualdenien & Borrmann, 2021;Zahedi et. al., 2021)

3.1. EXPLANATION TAGS

Rejecting, selecting, and further detailing architectural design decisions and variants depends not only on objective (quantitative) criteria but also on subjective (qualitative) criteria. In addition to recording the building model and quantitative criteria, qualitative and descriptive explanations and evaluations are also necessary to be captured, for the documentation of variant selection to make the decisions made and their justifications, e.g. design quality or aesthetics, comprehensible and to support the correct interpretation of the designers' choices (Zahedi et. al., 2021).

The goal is to store and document design decisions and variant selections without significantly interrupting the design process. For this purpose, a collection of identified keywords and visual icons (from the vocabulary of architectural design and other domain experts) with descriptions and examples of usage, are offered to the architect as so-called Explanation Tags, in the interest of clarifying and documenting his design decisions (Zahedi et. al., 2021). In designing the explanation tags, specific attention is paid to sustainability requirements introduced by the Federal Ministry of Transport and Digital Infrastructure in Germany (Fuchs et. al., 2013). However, this is just an open-ended collection of explanation tags, and the add-in provides the possibility to add custom-designed tags to this collection. Furthermore, the users can export or import their individual collections of tags as an archive via the add-in. During design in a BIM authoring tool, (Autodesk Revit in the case of this paper) the architect is given a collection of structured pictogram explanatory tags that cover most design criteria and concepts in the architects' vocabulary so that one can argue and justify the design decisions by adding and tagging these explanatory tags to building components or spaces and their respective attributes (Zahedi et. al., 2021).

3.2. DESIGN EPISODES

One of the solutions discussed in the literature for conveying tacit architectural knowledge is storytelling (Martin et al., 2003). Although this does not transfer large amounts of information, it is a means of promoting a better understanding (Heylighen et al. 2007). The outcome is more about the ideas, processes, decisions, and implications of the interactions presented in the story (Zahedi et. al., 2021). To this end, the design episode method, developed by Zahedi et. al. is used to divide the final design into smaller chapters and sections where each of these episodes relates to a design situation and offers a solution for it using storytelling techniques (Zahedi et. al., 2021). Design Episode is a method for summarizing and documenting a certain part of the design, using textual descriptions, coupled with the selection of the associated building elements and spaces (Zahedi et. al., 2021). Both concepts of explanation tags and design episodes and their utilization via our Revit add-in will be further explained using an example in the implementation section. Another associated goal is to export the recorded tacit design knowledge to a graph database and prepare this captured knowledge for future inquiries by other architects in search of inspirations and answers to similar design tasks and problems. Subsequently, the add-in that is developed for Autodesk Revit will be further explained in the next chapter.

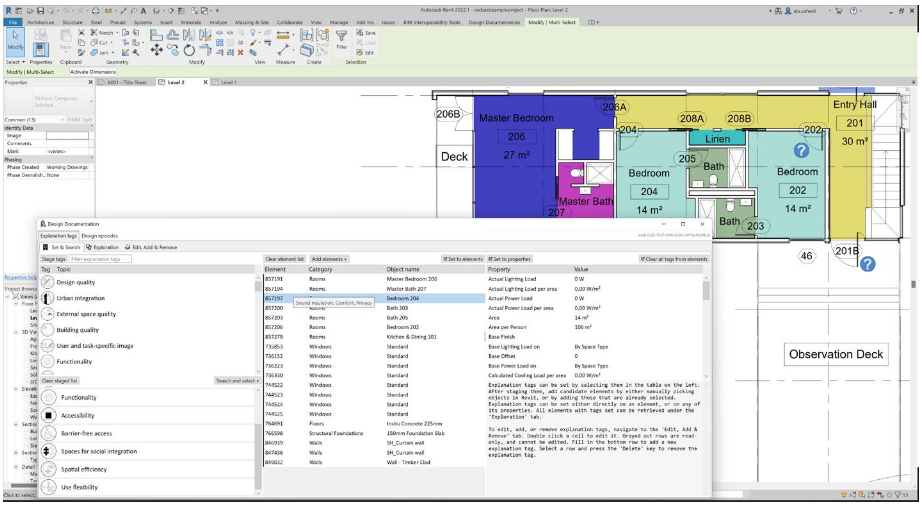
4. Implementation

In summary, the implemented add-in in Revit, allows the architects to accomplish the following:

- They can use design episodes to explain their motivations and intentions for different sections and aspects of the design via storytelling and in the form of textual descriptions together with the inclusion of corresponding model elements.
- They can use explanation tags to highlight and clarify the reasons and goals for different design decisions in a more graphical and explanatory way.
- They can export the documented tacit design knowledge in the form of various design episodes to a graph database (Neo4j) for later use and retrieval.

In general, the add-in simply adds a tool to Revit's toolkit, which collectively implies adding a tool to the Revit-ribbon. The add-in was written in C# (Microsoft programming language) for the code-behind and backend, whereas XAML (Extensible Application Markup Language) was used for the User Interface (UI). The add-in interacts with the Revit API, allowing users to communicate with the Revit environment and, as a result, read and write to any open Revit models. After being installed, the add-in is accessible under the Design Documentation tab in the Revit toolbar.

The add-in serves two main purposes, each of which is featured via its main tab in the tool. The first is to provide users with capabilities related to explanation tags. Namely, to allow the users to set and search explanation tags in the BIM model. To do so, the users can stage their desired tags and elements, and then assign various tags to either elements or their attributes. In other words, the user can associate a set of explanation tags to an element as a whole (*element-global explanation tag*), or assign them to a specific parameter or attribute of an element (*parameter-local explanation* *tag*) within the Revit context. An element in the Revit context is understood as a Revit element with an ID which could be different building components or spaces. Figure 2 shows the staging of several building elements and explanation tags for assigning. Moreover, an info log explains and reports on different actions that are taken in the add-in. In addition to setting explanation tags to elements, the user can also explore and search for certain elements that have specific explanation tags assigned to them, together with various sorting and filtering capabilities. Furthermore, the users can edit and explore the list of the already available explanation tags. At the same time, they can also create their own new tags and add them to the collection of existing explanation tags. Within the scope of this add-in, an explanation tag is an object that has a name, an image, a description, and zero or more synonyms. Synonyms are different explanation tags that have overlapping meanings or are synonymous with each other. The add-in is developed in a way so that the explanation tags are model-independent, which means they are kept globally and under the user's application data. This implies that explanation tags may be imported and exported to archives and used across various models.



Tag	Topic
	Design quality
	Urban integration
	External space quality
	Building quality
	User and task-specific image
	Functionality

	Functionality
	Accessibility
	Barrier-free access
	Spaces for social integration
	Spatial efficiency
	Use flexibility

Element	Category	Object name
857191	Rooms	Master Bedroom 206
857194	Rooms	Master Bath 207
857197		Bedroom 204
857200	Rooms	Bath 203
857203	Rooms	Bath 205
857206	Rooms	Bedroom 202
857279	Rooms	Kitchen & Dining 101
735853	Windows	Standard
736152	Windows	Standard
736223	Windows	Standard
736330	Windows	Standard
744522	Windows	Standard
744523	Windows	Standard
744524	Windows	Standard
744525	Windows	Standard
764691	Floors	Insitu Concrete 225mm
766598	Structural Foundations	150mm Foundation Slab
846939	Walls	SH_Curtain wall
847436	Walls	SH_Curtain wall
849032	Walls	Wall - Timber Clad

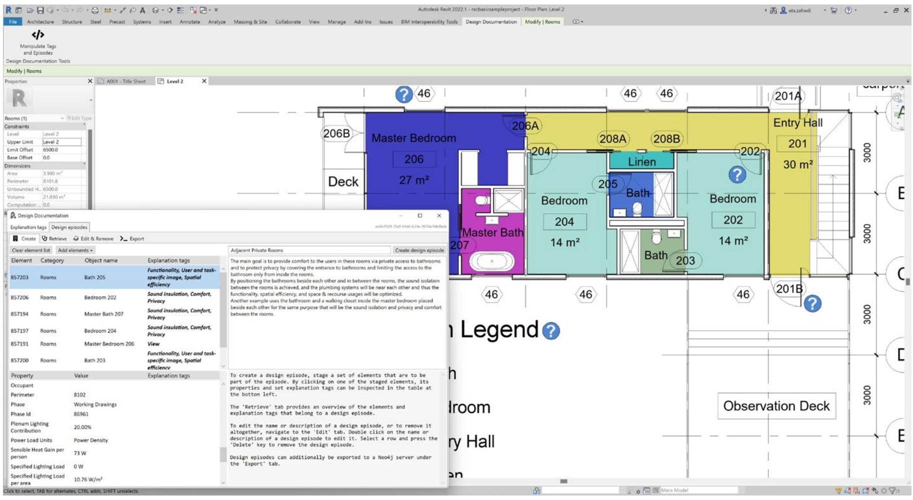
Property	Value
Actual Lighting Load	0 W
Actual Lighting Load per area	0.00 W/m²
Actual Power Load	0 W
Actual Power Load per area	0.00 W/m²
Area	14 m²
Area per Person	106 m²
Base Finish	
Base Lighting Load on	By Space Type
Base Offset	0
Base Power Load on	By Space Type
Calculated Cooling Load per area	0.00 W/m²

```
Explanation tags can be set by selecting them in the table on the left.
After staging them, add candidate elements by either manually picking
objects in Revit, or by adding those that are already selected.
Explanation tags can be set either directly on an element, or on any of
its properties. All elements with tags set can be retrieved under the
'Exploration' tab.

To edit, add, or remove explanation tags, navigate to the 'Edit, Add &
Remove' tab. Double click a cell to edit it. Grayed out rows are read-
only, and cannot be edited. Fill in the bottom row to add a new
explanation tag. Select a row and press the 'Delete' key to remove the
explanation tag.
```

Common (13)	
Identity Data	
Image	
Comments	
Mark	<varies>
Phasing	
Phase Created	Working Drawings
Phase Demolish...	None

Figure 2. Revit Add-in Tab for manupulating Explanation Tags showcasing the staging of some building elements and tags for assigning.

The second purpose of the add-in is to create and retrieve design episodes. Within the scope of the plugin, a design episode is an object with a name and description that contains a set of elements from the Revit model. In other words, upon creating a design episode, the user selects the set of elements (together with their attributes and possible explanation tags) that express this episode or chapter of design and then provides a name and description, using storytelling manner, for this design situation while explaining in more details the design motives, rationale, and argumentation. Figure 3 illustrates an example for creating design episodes. Unlike the explanation tags, the design episodes are model-bound rather than model-agnostic. This means that the design episodes are stored within a Revit document and can be passed on together with the BIM model. Just like explanation tags, design episodes can be also explored and retrieved, while different search and filtering possibilities are available to facilitate this feature. Additionally, design episodes can be modified or deleted, and if need be, collectively imported or exported to an archive.



Adjacent Private Rooms

Element	Category	Object name	Explanation tags
857203	Rooms	Bath 205	*Functionality, User and task-specific image, Spatial efficiency*
857206	Rooms	Bedroom 202	*Sound insulation, Comfort, Privacy*
857194	Rooms	Master Bath 207	*Sound insulation, Comfort, Privacy*
857197	Rooms	Bedroom 204	*Sound insulation, Comfort, Privacy*
857191	Rooms	Master Bedroom 206	*View*
857200	Rooms	Bath 203	*Functionality, User and task-specific image, Spatial efficiency*

The main goal is to provide comfort to the users in these rooms via private access to bathrooms and to protect privacy by covering the entrance to bathrooms and limiting the access to the bathroom only from inside the rooms.
By positioning the bathrooms beside each other and in between the rooms, the sound isolation between the rooms is achieved, and the plumbing systems will be near each other and thus the functionality, spatial efficiency, and space & recourse usages will be optimized.
Another example uses the bathroom and a walking closet inside the master bedroom placed beside each other for the same purpose that will be the sound isolation and privacy and comfort between the rooms.

Property	Value	Explanation tags
Occupant		
Perimeter	8102	
Phase	Working Drawings	
Phase Id	86961	
Plenum Lighting Contribution	20.00%	
Power Load Units	Power Density	
Sensible Heat Gain per person	73 W	
Specified Lighting Load	0 W	
Specified Lighting Load per area	10.76 W/m²	

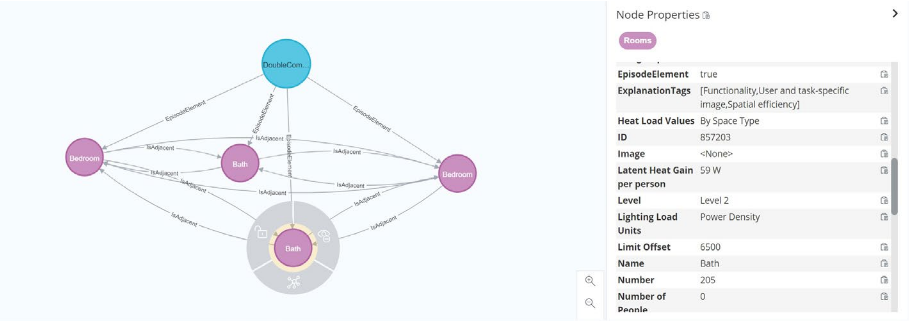
```
To create a design episode, stage a set of elements that are to be
part of the episode. By clicking on one of the staged elements, its
properties and set explanation tags can be inspected in the table at
the bottom left.

The 'Retrieve' tab provides an overview of the elements and
explanation tags that belong to a design episode.

To edit the name or description of a design episode, or to remove it
altogether, navigate to the 'Edit' tab. Double click on the name or
description of a design episode to edit it. Select a row and press the
'Delete' key to remove the design episode.

Design episodes can additionally be exported to a Neo4j server under
the 'Export' tab.
```

Rooms (1)	
Constraints	
Level	Level 2
Upper Limit	Level 2
Limit Offset	6500.0
Base Offset	0.0
Dimensions	
Area	3.980 m²
Perimeter	8101.8
Unbounded H...	6500.0
Volume	21.830 m³
Computation ...	0.0

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 3. Revit Add-in Tab for manupulating Design Episodes showcasing the "Adjacent private rooms" as an example for creating a design episode.

Individual design episodes can also be exported as a CSV file or a Labeled Property Graph (LPG) in Neo4j. In order to start with the exported graph typology, we need to have a better understanding of what this Neo4j export implies. Neo4j enables us to model objects and relationships between them as a lebeled property graph. As mentioned before, the graph typology provided in this research for the export of design episodes is loosely based on Abualdenien & Borrmann's Parametric Building Graph for capturing and storing detailed patterns (Abualdenien & Borrmann, 2021). Figure 4 demonstrates an example graph visualization of a design episode in the Neo4j database. Two sorts of nodes will be present in our network. The *Design Episode* node, which, as the name suggests, is an exported design episode, and the *Element* node, which is an exported episode element, also depicted via an *EpisodeElement* flag that is set to true on these nodes. There are several relationships among the elements and design episodes. The first relationship is the *EpisodeElement* relationship, which indicates that an element belongs to a design episode. Some other relationships between the elements are, *ContainedIn*, *IsAdjacent*, *IsConnected*, and *Has* relationships. *ContainedIn* relationship specifies that an element is contained in another element. An *isAdjacent* relationship between two nodes implies that the corresponding elements in the Revit model are adjacent to each other, whereas, *IsConnected* suggests that the originating elements are directly connected to each other in the Revit model. The *Has* relationship indicates that one element belongs to another element. In addition, each node has several attributes associated with it. *Model Identifier* and *Export DateTime* as strings are assigned to design episode nodes in order to link them to their originating Revit model and make it easier to trace them. The GUID attribute helps us in distinguishing between different design episodes with similar names. Finally, there's the design episode's Name and Description, which respectively represent and describe the corresponding design situation and the motives behind this episode of design, using storytelling techniques. Furthermore, as the basis for the element nodes, we have the EpisodeElement boolean flag, and some properties that are directly taken from Revit, such as ID, Unique ID, and Object Name, which are in that order; element's Revit-provided id and unique id, and name. The *ExplanationTags* property holds the element's *element-global explanation tags*. Moreover, when exporting a design episode to Neo4j, the user can select and choose the desired parameters of each element in the design episode to be exported or not. For each of these parameters that get exported, there is also a property (assigned to the element node) that contains the corresponding parameter along with its value from Revit. Similarly, there is an *ET[PropertyName]* attribute that maps the set of *parameter-local explanation tags* to the element node for each of the exported parameters that has parameter-local explanation tags assigned to them.



EpisodeElement	true
ExplanationTags	[Functionality,User and task-specific image,Spatial efficiency]
Heat Load Values	By Space Type
ID	857203
Image	<None>
Latent Heat Gain per person	59 W
Level	Level 2
Lighting Load Units	Power Density
Limit Offset	6500
Name	Bath
Number	205
Number of People	0

Figure 4. an Screenshot of the Neo4j database showcasing the graph Typology of the same "Adjusent private rooms" design episode when exported into Neo4j database.

5. Conclusion

A crucial step in communicating with owners and domain experts is to document design knowledge, objectives, and conclusions. This also makes future assessment and reuse of finished projects easier. Building industry and especially the architectural design process lack proper documentation and preservation of tacit design knowledge, which could be enormously useful to address and solve similar design tasks and problems in the future. This precious knowledge often gets lost and forgotten due to the absence of appropriate explanation and documentation. Proper design documentation, we believe, may lead to improved reuse of design knowledge and expertise.

Existing BIM models contain only plain geometry and semantics, with no reasoning or explanation for design choices. The development and demonstration of an add-in for Revit that utilizes the ideas of so-called Explanation Tags and Design Episodes introduced in another manuscript submitted for publication by the authors

(Zahedi et. al., 2021) to digitally explain and document design reasoning and choices is described in this work. Furthermore, the add-in also allows the user to export design episodes as LPG to a Graph-Database (Neo4j) or CSV file for future reference. Following that, the next step is to explore and query the graph database for similar designs that can address and resolve future design challenges. By explicitly documenting the design decisions and rationale, resulting in capturing the tacit architectural design knowledge, this work adds to the United Nations Sustainable Development Goals, particularly those related to 'Industry, Innovation, and Infrastructure" and "Sustainable Cities and Communities'. Special attention is made to 'Sustainability Requirements in Planning Competitions' (SNAP-Systematik für Nachhaltigkeitsanforderungen in Planungswettbewerben) upon identifying appropriate explanation tags for design decisions' documentation in the developed Revit add-in.

Acknowledgements

We gratefully acknowledge the support of the German Research Foundation (DFG) (DFG, last updated: 2021) for funding the project under the grant FOR-2363. We also thank Max van Deuzen for helping us to implement this add-in as part of his interdisciplinary student project.

References

Abualdenien, J., & Borrmann, A. (2021). PBG: A parametric building graph capturing and transferring detailing patterns of building models. In *Proc. of the Conference CIB W78.*

Al Hattab, M., & Hamzeh, F. (2018). Simulating the dynamics of social agents and information flows in BIM-based design. *Automation in Construction*, 92, 1–22.

Zahedi, A., Abualdenien, J., Petzold, F. & Borrmann, A. (2021). Documenting Design Decisions using Design Episodes, Explanation Tags and Constraints: Manuscript submitted for publication.

Ayzenshtadt, V., Espinoza-Stapelfeld, C., Langenhan, C., & Althoff, K.-D. (2018). Multi-Agent-Based Generation of Explanations for Retrieval Results Within a Case-Based Support Framework for Architectural Design. In *ICAART* (1).

Cao, Q., & Protzen, J.-P. (1999). Managing design information: Issue-based information systems and fuzzy reasoning system. *Design Studies*, 20(4), 343–362.

DFG. (last updated: 2021). *Deutsche Forschungsgemeinschaft e*.V. https://www.dfg.de/

Dubey, R. K., Khoo, W. P., Morad, M. G., Hölscher, C., & Kapadia, M. (2020). AUTOSIGN: A multi-criteria optimization approach to computer aided design of signage layouts in complex buildings. *Computers & Graphics*, 88, 13–23.

Exner, H., Abualdenien, J., König, M., & Borrmann, A. (2019). Managing Building Design Variants at Multiple Development Levels. In *Proc. of the 36th International Council for Research and Innovation in Building and Construction (CIB W78).*

Gänshirt, C. (2012). *Werkzeuge für Ideen: Einführung ins architektonische Entwerfen*. Walter de Gruyter.

Hamieh, A., Makhlouf, A. B., Louhichi, B., & Deneux, D. (2020). A BIM-based method to plan indoor paths. *Automation in Construction*, 113, 103120.

He, T., Zhang, J., Lin, J., & Li, Y. (2018). Multiaspect similarity evaluation of BIM-based standard dwelling units for residential design. *Journal of Computing in Civil Engineering*, 32(5), 4018032.

Heylighen, A., Martin, W. M., & Cavallin, H. (2007). Building stories revisited: unlocking the knowledge capital of architectural practice. *Architectural Engineering and Design Management*, 3(1), 65–74.

Heylighen, A., Neuckermans, H., Casaer, M., & Dewulf, G. P. M. (2007). Building memories. *Building Research & Information*, 35(1), 90–100.

Hor, A. E., Gunho, S., Claudio, P., Jadidi, M., & Afnan, A. (2018). A semantic graph database for BIM- GIS intergtated information model for an intelligent urban mobility web application. *Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(4).

Isaac, S., Sadeghpour, F., & Navon, R. (2013). Analyzing building information using graph theory. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction.* Symposium conducted at the meeting of IAARC Publications.

Ismail, A., Strug, B., & Ślusarczyk, G. (2018). Building knowledge extraction from BIM/IFC data for analysis in graph databases. In *International Conference on Artificial Intelligence and Soft Computing*. Symposium conducted at the meeting of Springer.

Kolltveit, B. J., & Grønhaug, K. (2004). The importance of the early phase: the case of construction and building projects. *International Journal of Project Management*, 22(7), 545–551.

Langenhan, C., Weber, M., Liwicki, M., Petzold, F., & Dengel, A. (2013). Graph-based retrieval of building information models for supporting the early design stages. *Advanced Engineering Informatics*, 27(4), 413–426.

Maher, M. L., Balachandran, M., & Zhang, D. M. (1995). *Case-based reasoning in design*. Psychology Press.

Martin, W. M., Heylighen, A., & Cavallin, H. (2003). Building^2 Stories. A hermeneutic approach to studying design practice. In *Proceedings of the 5thEuropean Academy of Design Conference*.

Martin, W. M., Heylighen, A., & Cavallin, H. (2005). The right story at the right time. *AI & SOCIETY*, 19(1), 34–47.

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

Meng, Z., Zahedi, A., & Petzold, F. (2020). Web-Based Communication Platform for Decision Making in Early Design Phases. In *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC)*.

Neuckermans, H., Heylighen, A., & Morisse, P. (2002). Visual Keys to Architectural Design.

Fuchs, M., Hartmann, F., Henrich, J., Wagner, C., & Zeumer, M. (2013). SNAP Systematik für Nachhaltigkeitsanforderungen in Planungswettbewerben-Endbericht. Bonn, Berlin.

Steinmann, F. (1997). *Modellbildung und computergestütztes Modellieren in frühen Phasen des architektonischen Entwurfs*, [Bauhaus University, Weimar, Germany].

Zahedi, A., Abualdenien, J., Petzold, F., & Borrmann, A. (2019) Minimized Communication Protocol Based on a Multi-LOD Meta-Model for Adaptive Detailing of BIM Models. In *EG-ICE 2019*.

Zahedi, A., & Petzold, F. (2019). Adaptive Minimized Communication Protocol based on BIM. In *2019 European Conference on Computing in Construction*.

Zeiler, W., Savanovic, P., & Quanjel, E. (2007). Design decision support for the conceptual phase of the design process.