iDesignArch: Chatbot-supported architectural requirements management for inspiring building designs

Jan H. Heinbach[1] and Daniel Napps[1]
[1]Chair of Computing in Engineering, Ruhr Universität Bochum, Universitätsstrasse 150, 44801 Bochum, Germany
E-mail(s): Jan.Heinbach@ruhr-uni-bochum.de, Daniel.Napps@ruhr-uni-bochum.de

Abstract: Describing buildings and their internal structures is a complex architectural task. Nevertheless, a precise description is essential, especially if the description should enable the search for suitable building designs among existing models. Building Information Modelling (BIM), floor or zone plans simplify the description by providing a standardized format for mainly geometric information in different levels of detail. However, creating building representations for utilization as search input is time consuming and should be carried out by experienced users only. Therefore, this paper introduces a tool that enables architects and other stakeholders to gather relevant information by answering specific project related questions posed by the tool. It generates questions based on a developed decision tree and processes the answers, to gather as much information for an upcoming building project as possible. The resulting description is subsequently used as input for identifying similar digital building models from a database. This intelligent solution, which also allows inexperienced users to use it, offers an efficient and user-friendly way of recording and modifying building requirements in order to gain inspiration from existing similar buildings for the design, conversion or modernization of a building or part of a building.

Keywords: Building Information Modeling (BIM), Decision Tree, Design Automation, Project Requirements

Erschienen in Tagungsband 35. Forum Bauinformatik 2024, Hamburg, Deutschland, DOI: 10.15480/882.13513

1 Introduction

The current building design process has evolved significantly, integrating technologies like Building Information Modeling (BIM) and Industry Foundation Classes (IFC) to improve efficiency and accuracy. This transformation is driven by the need to improve project outcomes, reduce costs, evaluate alternative designs and mitigate mistakes. Early design decisions are crucial, as they impact the whole project lifecycle. Decisions made during these early stages can have a great impact on project success, affecting everything from aesthetics, cost and schedule to sustainability and operational energy and efficiency [1]. BIM has proven to be a critical component of modern building design, facilitating collaboration by providing a shared digital model, improving visualization, coordination, and decision-making. Using BIM in the design process ensures that potential issues are identified and resolved early, reducing the likelihood of costly changes and delays during construction [2]. IFC is an open standard for a loss free data exchange of models in the architecture, engineering, and construction (AEC) industry. It enables seamless interoperability between disparate software applications, ensuring interchangeability and effective use of information. By using IFC in the design process, communications and collaboration are improved by ensuring that everyone is working with accurate and up-to-date information [3]. However, the lack of a centralized model database hampers efficiency, as each new project requires time-consuming (and often redundant) designs from scratch. This absence limits the ability to reference or utilize previous similar building designs, complicating efforts to streamline and optimize new projects [4]. In fact, in addition to a centralized database, the reuse of existing models requires a way to search systematically. Enabling a systematic search itself requires a possibility of effective and structured input of project parameters, limits and objectives to search for. This work focusses on the development and implementation of such an user-friendly and efficient method that captures the multitude of possible design decisions and their potential building parameters.

2 Background

To address the problem of starting designs from scratch and the lack of a centralized model database, previous research has developed a framework to overcome the limitations and support the architects in the early design phases with inspiration for new projects. The methodology used is based on the Case-Based Reasoning (CBR) concept, which enables problem-solving and learning by using past experiences or so called cases to understand and solve new problems. This approach allows for multiple input parameters in the comparison process, enabling specific user queries to be matched against the database [5]. Initially, the method was developed to compare BIM models. These models are exported as IFC files and transformed into a graph representation. The structural alignment of a knowledge graph with a graph from the database is performed using graph pattern matching. The graphs are analyzed for common subsets, and if a structural match is found, various similarity algorithms are then applied to compare the semantic information in the respective nodes and edges. The nodes in these graphs represent IfcEntities, such as $IfcWall$ or $IfcDoor$, and the edges represent IfcRelationships. In the first step, for instance, an $IfcWall$ is matched with another $IfcWall$, and selected semantic and geometric information of the entities are compared. This comprehensive method calculates the overall similarity, considering both structural and detailed semantic aspects [6]. Subsequent research has extended the approach to allow for multiple input parameters in the comparison process, so that specific user queries can be matched against the database and also include other analytical result in the calculation, such as cost estimation [7], the detailing of the specific elements [8] and even capture subjective criteria for design decisions [9]. The presented tool should be integrated in this framework to enable more effective search input.

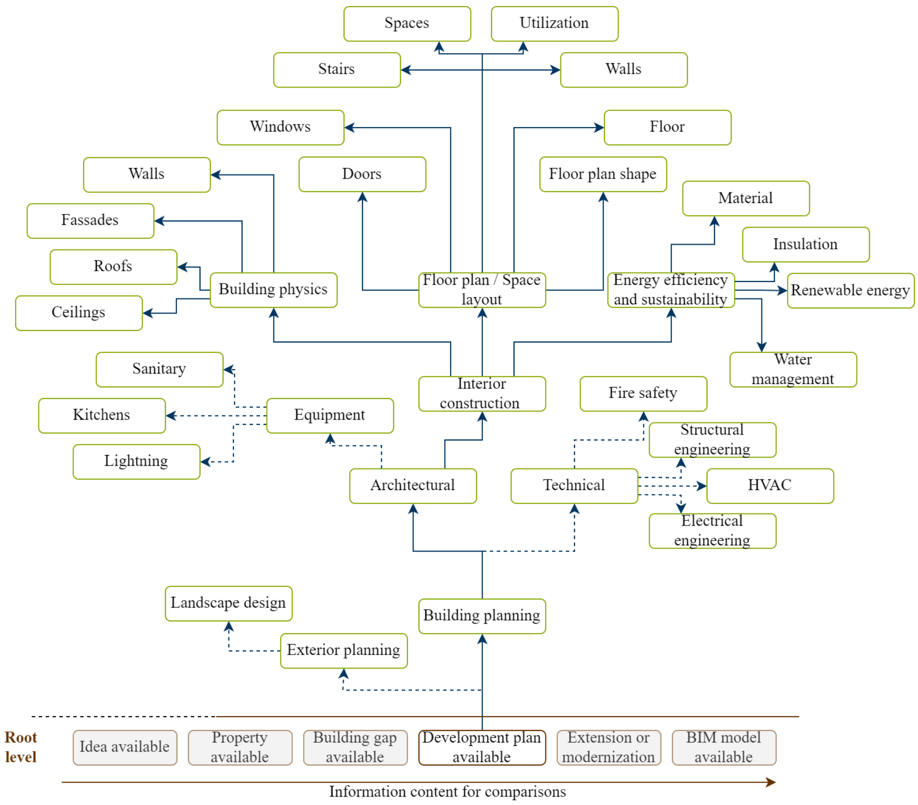

Figure 1: The developed iDesignArch decision tree for input with available development plan.

3 Methodology

The methodology of this work aims to provide architects with an structured alternative to the currently difficult and unstructured method of building descriptions making it easier to subsequently use it for searching in a database.

3.1 Requirements

Matching information from BIM models, represented by IFC, allows for a very structured process and captures detailed information requirements. The goal of getting away from required BIM models as search input, while preserving the possible information amount, is challenging. Especially specific parameters for new projects — such as plot size, usage type, or the number of certain units or rooms — are significantly increasing the complexity of the retrieval process. This abundance of potential information makes it nearly impossible to manage effectively.

For this reason, potential levels for possible input accuracy are differentiated on root level of the developed decision tree. This distinction considers whether the architect has only a rough idea for a new project (Idea available) or if there is already an existing plot of land, a development plan, or even a BIM model. The decision tree in Figure 1 illustrates an example in case of the existence of a specific development plan for a new project and the possible requirements that an architect can capture to match them with a suitable BIM building from the database (searching for models). This work focuses on the solid line, describing the possibilities of architectural design for a building.

Nevertheless, capturing information from the other branches of the tree (dashed line) is also feasible. Thus, the architectural decisions focus on the internal structure, building design, floor plan, and energy efficiency. The information becomes increasingly detailed down to the leaves of the tree (e.g., Walls and Windows). It should be noted that these leaf nodes do not represent the end of the tree, as further details such as materiality or other properties are stored for each of them. The tree is structured to cover the most crucial decisive information in architecture that have a crucial impact on the building's lifecycle and are available during design phase.

3.2 iDesignArch

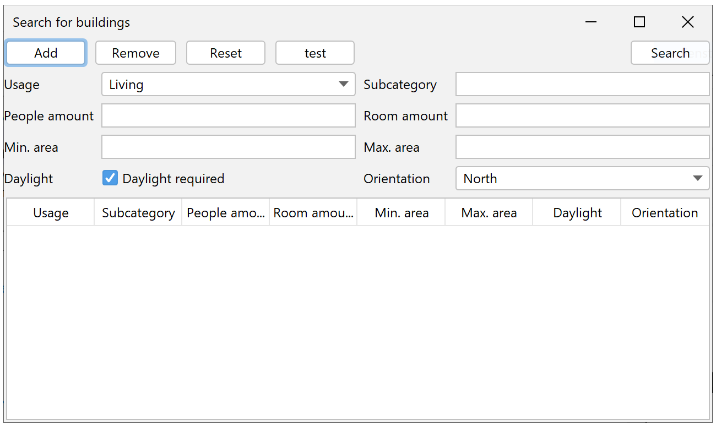

Figure 2: The sub-GUI for room details from the first approach.

The developed structured, innovative, and user-friendly tool is introduced as iDesignArch. This abbreviation stands for Intelligence Design-Decision-Making for Architects. The basis of iDesignArch is the discussed and carefully crafted decision tree for decision-making in early design phases.

3.2.1 Previous Approach

In a first approach, a Graphical User Interface (GUI) for gathering the information described in Section 3.1 was developed. However, due to the complexity of the input data the GUI is unintuitive, complex and confusing. It consist of a lot of different sub GUIs, of which an example for room details can be seen in Figure 2. In contrast to creating a simple schematic BIM model, this approach involves numerous detailed steps. Additionally this approach is less flexible.

3.2.2 Decision Tree

The search input needs to be structured according to the required parameters for the building search. Some of the information, which is very complex to enter, is not even required for the actual search. Therefore the input possibilities can be filtered and reduced according to their importance for good search results.

A closer look on the relevant information reveals dependencies between the different search parameters. These dependencies are sometimes obvious (e.g. details about the rooms are required before wall details can be added) from the context of building physics, but sometimes they purely arise from the search design itself (e.g. requiring floor plan shape before adding room details). These dependencies result in the fact, that some information, which was entered in a complicated way, could not be used for a search due to missing additional information it depends on. Combining these dependencies and the insight, that some parameters are more important than others, a decision tree was developed. In fact there are multiple decision trees, as the information basis (e.g. only idea available, property known, development plan available or BIM model available) highly effects the required information and their importance. This paper sets the focus on the availability of an existing development plan. The resulting decision tree for that case is visible in Figure 3.

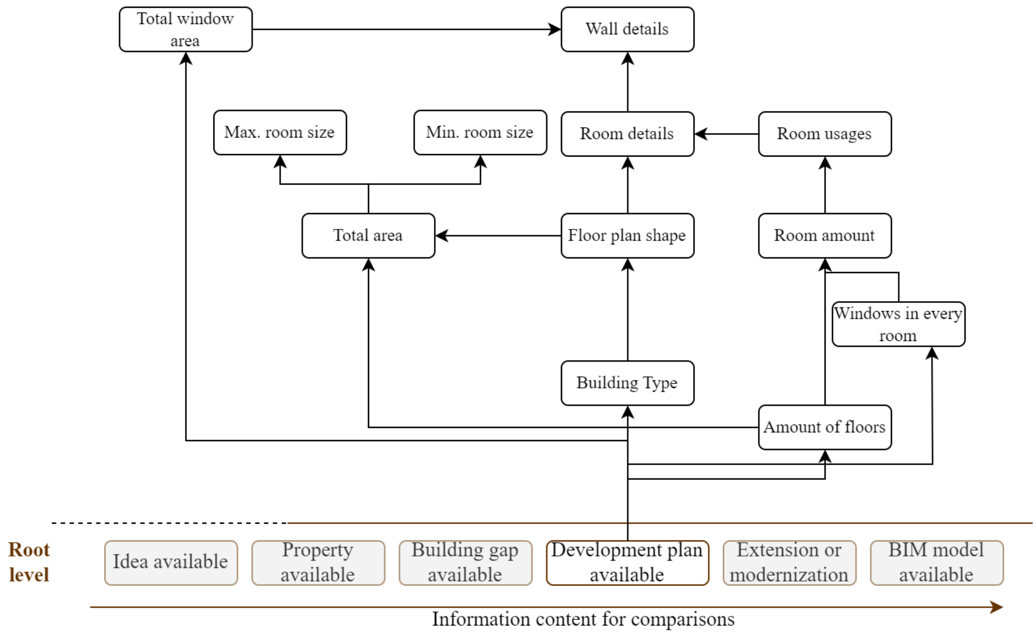

Figure 3: The developed iDesignArch decision tree for specific input data with available development plan.

The decision tree represents a structured way to gather the required information. A node is considered as *fulfilled*, if the information, represented by a node is available and all parent nodes are *fulfilled* as well. Based on Figure 3, the *total area* node is *fulfilled*, if the total area of the building is provided (available information), as well as the *Floor plan shape* and *Amount of floors* nodes are *fulfilled* (*fulfilled* parents). An example input will be discussed in Section 5.2. The root node of each decision tree is the initial information basis, which in this case is the available development plan. As long as a node is not *fulfilled*, the corresponding information can not be used for a search until additional information is provided. The decision tree is used to identify these required information.

4 Implementation

The decision tree approach developed in Section 3 is used to develop an interactive input method which enables user friendly and fast input of information without requiring a schematic BIM model. An initial approach uses sequential input (Subsection 4.1), while the final approach uses a Large Language Model (LLM) to enable free text input and increase usability (Subsection 4.2).

4.1 Sequential Input

To use the decision tree in an actual input, the tree need to be available in a machine readable way. In addition to that, every node is associated with a string, containing a human understandable question for the corresponding parameter. The input program will process the decision tree sequentially. Starting at the root node asking the user the corresponding question level by level. As visible in Figure 4, the user than enters the relevant information. However the total amount of project information gathered by the tool is drastically reduced, as the input is limited to search relevant questions.

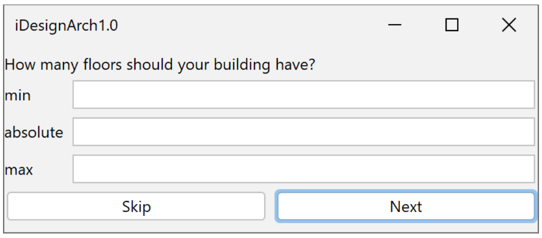

Figure 4: The sequential input window from iDesignArch. Input for *Amount of floors* node.

4.2 Large Language Model Input

To combine the advantages of the GUI and the sequential approach, as well as to increase usability, a LLM — in our case customized ChatGPT [10] — is used to process user input. The customized ChatGPT enables the user to freely describe the searched building and then subsequently answer queries, raised according to the decision tree, which are crucial to the search quality. More detailed, the user need to answer questions until all nodes, for which information was provided, are *fulfilled*. Afterwards the user is asked to optionally provide information about nodes on the next level in the decision tree.

The LLM is used only for processing user input and generating questions. The actual search or data processing happens outside of ChatGPT. Therefore iDesignArch does not have to use ChatGPT. The implementation need to be understood as a proof of concept. Information matching one of the nodes in the decision tree is passed to the search tool, and all other information is captured as project information.

5 Demonstration

This section describes an exemplarily usage of iDesignArch with LLM (Subsection 4.2).

5.1 Setup

For demonstration purposes, an office worker with no background in building or construction industry, should search for a new building for his department. We briefly explained the information available from the development plan (e.g. property dimensions and height regulations) but did not prescribe any other properties. Due to the scope of this paper, we manually summarized the input in Table 1. The summary was created by a human, reading the input provided by the office worker and extracting the properties. The original input started with : "Hello. I want to build a new office building. I need 3 to 5 stories with at least 35 offices. I need 5 conference rooms..."

Table 1: Manual summary of input requirements.

office Building	3 to 5 floors	min. 35 offices
5 conference rooms	2 bathrooms per floor	5 additional rooms
large room on ground floor	smart home functions	air conditioning

5.2 Results

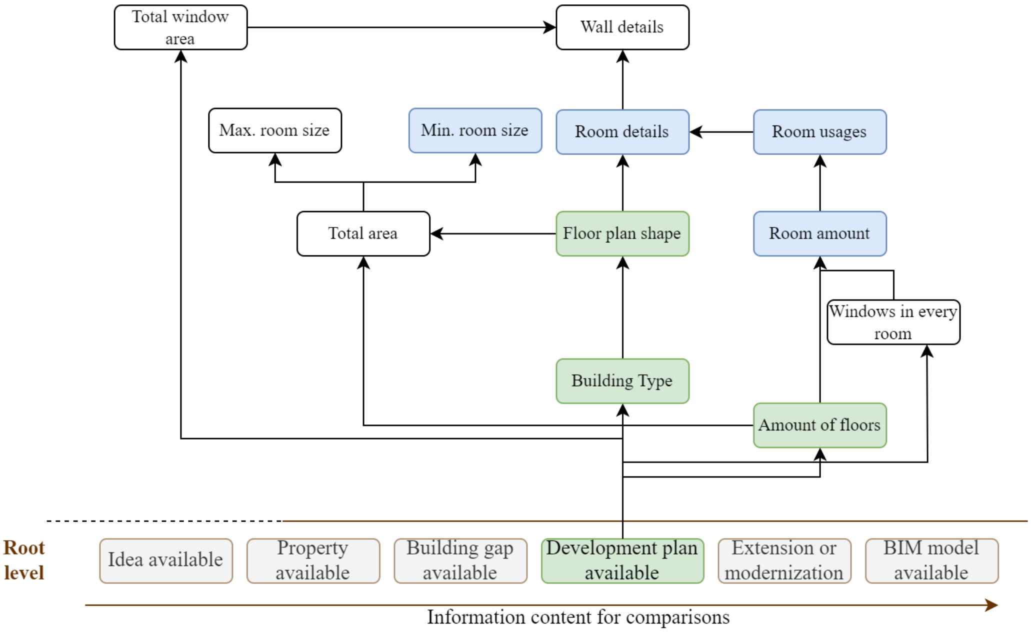

Figure 5: The iDesignArch decision tree after initial input. Nodes highlighted in blue if information is provided, green if *fulfilled*. For white nodes there is no information provided.

As visualized in Figure 5, some nodes have provided information, without being fulfilled (blue). For example the room amount was specified. The node *Room amount* on the top right has two parents: *Amount of floors* and *Windows in every room*. While one of them is *fulfilled* (green), there is no information about *Windows in every room*. Therefore, *Room amount* and all children can not be *fulfilled*.

Accordingly, ChatGPT asks for the missing information to *fulfil* all blue nodes. In our example the prompt is "Would you like all the rooms in your office building, including the offices, conference rooms, and additional rooms, to have windows?" [1].

Subsequently to the input phase the data to is extracted from ChatGPT as JSON and passed on to the search tool. This is where current limits of the demonstration application become clear, as ChatGPT did not always use the correct JSON identifiers required for the search input. In the same way there is no guarantee that ChatGPT (or any other LLM) interprets user input correctly. Additionally there is no guarantee how wrong user input is handled. Therefore, future work, especially reducing these issues, is required.

[1] Generated by ChatGPT [10]

6 Conclusion

The presented iDesignArch approach uses current technologies — more precisely LLMs — to enhance and simplify the user input for capturing project data. Together with search algorithms from previous research, it enables the search for similar buildings, even with input from inexperienced users. To ensure the transparency, reproducibility and modularity of the approach, the LLM is used only to handle the user input. Despite promising performance, there are still some challenges regarding the LLM to be overcome. Actual search and information processing is handled without them. This ensures that if not available or requested, the LLM can be replaced by iDesignArch GUIs, another LLM or other input possibilities.

References

[1] Y.-W. Song, M. Herzog, and B. Bender, “Understanding the initial requirements definition in early design phases”, en, *Proceedings of the Design Society: International Conference on Engineering Design*, vol. 1, no. 1, pp. 3751–3760, Jul. 2019. DOI: 10.1017/dsi.2019.382.

[2] M. K. Najjar, K. Figueiredo, A. C. J. Evangelista, A. W. A. Hammad, V. W. Y. Tam, and A. Haddad, “Life cycle assessment methodology integrated with bim as a decision-making tool at early-stages of building design”, *International Journal of Construction Management*, vol. 22, no. 4, pp. 541–555, Mar. 2022. DOI: 10.1080/15623599.2019.1637098.

[3] 14:00-17:00, *Iso 16739-1:2024*, en. [Online]. Available: https://www.iso.org/standard/84123.html.

[4] C. J. Hopfe and R. S. McLeod, “Enhancing resilient community decision-making using building performance simulation”, *Building and Environment*, vol. 188, p. 107 398, Jan. 2021. DOI: 10.1016/j.buildenv.2020.107398.

[5] A. Aamodt and E. Plaza, “Case-based reasoning: Foundational issues, methodological variations, and system approaches”, en, *AI Communications*, vol. 7, no. 1, pp. 39–59, 1994. DOI: 10.3233/AIC-1994-7104.

[6] D. Napps, D. Pawlowski, and M. König, “Bim-based variant retrieval of building designs using case-based reasoning and pattern matching”, en-US, *ISARC Proceedings*, pp. 435–442, Nov. 2021.

[7] D. Napps, J. Maibaum, and M. König, “Identifying cost-efficient alternatives to building designs based on building regulations”, en,

[8] D. Napps and M. König, “Consideration of detailing in the graph-based retrieval of design variants”, in *ECPPM 2022 - eWork and eBusiness in Architecture, Engineering and Construction 2022*. CRC Press, 2023.

[9] D. Napps, A. Zahedi, M. König, and F. Petzold, “Visualisation and graph-based storage of customised changes in early design phases”, en-US, *ISARC Proceedings*, pp. 191–198, Jul. 2022.

[10] OpenAI, *Chatgpt: A large language model based on gpt-4 architecture*, 2024. [Online]. Available: https://www.openai.com/research/chatgpt.