Abstract. The paper focuses on the early stages of the design process where the architect needs assistance in finding reference projects and describes different aspects of a concept for retrieving previous design solutions with similar structural characteristics. The proposed system offers a computational approach to extracting a few characteristic and prominent features of a floor plan which are then used to generate a semantic fingerprint. We propose the use of a visual query language and a semantic structure to query and create a floor plan repository. This enables the user of the system to sketch a schematic abstraction of a floor plan and search for floor plans that are structurally similar. It is a collective effort to create a community knowledge base about past projects and the retrieval strategies of this information. We examine the use of classic mouse and keyboard and pen-and-touch-enabled interaction methods to annotate and retrieve spatial situation.

1. Introduction

The rapid development of information technology in architectural planning raises questions about the storage and use of growing amounts of data. Information about architecture is available digitally but not accessible. There is a lack of adequate retrieval methods, including suitable approaches to the storage, indexing and management of information. Traditionally planning information is represented in the form of floor plans, elevations, sections and textual
descriptions. State of the art digital representations include renderings, computer aided design (CAD) and Building Information Modeling (BIM) such as Industry Foundation Classes (IFC).

Current electronic search methods use textual information rather than graphical information. The configuration of space and the relations between physical structures are hard to represent using keywords, in fact transforming these structural configurations into verbally expressed typologies tends to result in unclear and often imprecise descriptions of architecture. Classifications are made in terms of types, morphology, similarity or patterns but the quantitative and qualitative comparison of functional as well as structural features is as yet not possible. A universal description and query language is indispensable for storing descriptive metadata independent of file type and source as well as structural, graphical or textual information. Gruber describes [1] Collaborative Knowledge Management as an approach to enabling organizational intelligence. CKM is a software environment where people work together to achieve individual and team goals and contribute to the collective knowledge, which is then made available to everyone. The key idea is to motivate participation in a collective knowledge creation process by supporting environments for collaborative work, and to harvest the value.

Ann Macintosh from the AI Applications Institute (AIAI, Edinburgh, UK) motivates the need for knowledge asset management as follows [2]: "Enterprises are realizing how important it is to "know what they know" and be able to make maximum use of the knowledge. This is their corporate knowledge asset. These knowledge assets reside in many different places such as : databases, knowledge bases, filing cabinets and peoples" heads and are distributed right across the enterprise. All too often one part of an enterprise repeats work of another part simply because it is impossible to keep track of, and make use of, knowledge in other parts. Enterprises need to know : what their corporate knowledge assets are; and how to manage and make use of these assets to get maximum return". Since the 1990s knowledge management systems have become more and more successful in several application areas [3]. With the help of ontologies or organizational memories (information models), the way of storing and retrieving knowledge has been improved.

The semantic fingerprint of spatial configurations [4] is a description and query language for creating an index of floor plans to store metadata about architecture, which can be used as signature for retrieving reference projects. Rather than using today’s keyword-based search methods, it employs geometrical search strategies. A graphical user interface supports sketch-based as well as textual retrieval strategies to search for spatial configurations, for example by drawing rooms and their relations to one another. The main goal is to extract significant features of community data available in the internet and transform these features into a spatial query language for retrieving architectural information. The fingerprint
enables the user to digitally document information about architecture by creating an index of significant features that can be retrieved and used in a normal everyday manner.

2. Structure of the paper

The section "Related work" discusses past approaches to supporting the design process using Artificial Intelligent (AI) technologies. In subsection "Knowledge management : Case-Based Design (CBD)" a brief description of applications using Case-Based reasoning methods in the area of building design is given. The subsection "Storing : raster graphics, CAD and BIM" describes state of the art methods for digitally saving geometrical 2D and 3D information. The section "Indexing : semantic fingerprint" presents the extraction process of features to create the meta-signature. Subsection "Interaction" gives a brief overview of current interaction strategies. In section "Retrieval : sketch-based" we specify the combined graphical-textual search process and finally, the section "Conclusion and future work" provides an overview of possible developments.

3. Related work

3.1. Knowledge management : Case-Based Design (CBD)

In the early stages of the architectural design process, architects are only rarely able to specify the required information. Case-based reasoning (CBR) is an area of AI and describes a knowledge management process based on conclusion by analogy. It attempts to assess similarities according to the basic premise that similar problems have similar solutions. In CBR a case consists of a problem and solution description. By entering a new problem description to obtain similar solutions the CBR system first searches for an old problem description. Figure 1 illustrates the basic concept of CBR, similar problems have similar solutions. Aamodt and Plaza [5] described this adaptation of the thinking process inside the CBR cycle with the verbs retrieve, reuse, revise and retain.
Since the middle of the 1990s the approach of applying CBR to design and architectural tasks has been known as Case-Based Design (CBD). The case-base contains information on buildings that have already been built or designed, enabling the computer to adapt solutions accordingly, on its own or with help from the architects. Table 1 provides a brief overview of some CBD systems based on two studies published by Heylighen et al. [6] and by Richter et al. [7] regarding the proposed approach. The marked fields show whether the appropriate feature was realized in the concept.

Six of the CBD prototypes (CADRE [8], FABEL [9], IDIOM [10], SEED [11], SL_CB [12] and TRACE [13]) aim to partially or completely automate the generation of building layouts by applying the retrieved solution. Two of these prototypes (CADRE and IDIOM) leave the selection of the reference project to the user. The remaining four (FABLE, SEED, SL_CB and TRACE) apply the solution to the given architectural problem automatically and generate building layouts independently with very little user input. The more state-of-the-art approaches aim on supporting users during the design process like Archie-II [14], PRECEDENTS [15], CaseBook [16], MONEO [17], CBA [18] and DYNAMO [19].
Table 1. CBD systems.

<table>
<thead>
<tr>
<th>CBD APPLICATION AND SUPPORTED FEATURE</th>
<th>DATA STORAGE</th>
<th>INPUT SYSTEM</th>
<th>OUTPUT SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Floor plans + text</td>
<td>Abstraction</td>
<td>Topology</td>
</tr>
<tr>
<td>Archie-II</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CADRE</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FABEL</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>IDIOM</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PRECEDENTS</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SEED</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SL_CB</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>TRACE</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CaseBook</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>MONEO</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CBA</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DYNAMO</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

The study by Richter et al. in 2007 identifies an acquisition bottleneck in putting complete case descriptions (problem and solution) into the case base. We assume this is due to a lack of adequate input strategies, indexing methods and knowledge management procedures. First of all, a user interface should support the graphical sketch-based workflow of architects combined with textual, schematic and tabular input strategies. Secondly, a lightweight indexing strategy such as the semantic fingerprint is needed in contrast to the overall data storage method used. Thirdly, the problem and solution descriptions need to be stored
according to the CBR paradigm. Most of the CBD prototypes do not properly implement this fundamental CBR attribute.

3.2. Storing: CAD and BIM

The use of metadata enhanced digital floor plans with additional information offers the opportunity to create smart objects that allow users to have easier access to planning material. Nowadays, semantic models, such as BIM and the semantic fingerprint, are used as data representation. Hence, enriching geometrical data with semantic information enables the applications to automatically identify rooms, doors or chairs.

Furthermore, 3D knowledge-rich parametric modeling systems are central to BIM and the life cycle of a building. As buildings are composed of geometric components, geometric information forms a substantial part of BIM. In addition, further related knowledge is added to the BIM, such as project information, lighting analyses or quantities and properties of building components and so forth.

Modern architectural design is done using Computer Aided Design (CAD) software. Several vendors, such as Autodesk (Revit Architecture), Graphisoft (ArchiCAD) and Nemetschek (Allplan) offer software packages with their own data formats to store information about the building. However, according to the BIM paradigm interoperability is vital for reducing costs and supporting all stakeholders. The buildingSMART e.V. [20] established an open specification that is not controlled by a single vendor. The file format Industry Foundation Classes (IFC) is an interoperable BIM standard for CAD applications.

3.3. Interaction

Sketches are widely used in engineering and architectural fields as they are a familiar, efficient and natural way of expressing certain kind of ideas. In [21], Juchmes proposed approach for analyzing hand-drawn architectural sketches. Sezgin et al. introduced in [22] an implemented system that combines multiple sources of knowledge to provide robust early processing for freehand sketching.

Sim-U-Sketch is a sketch-based interface for Simulink [23] where users can construct functional Simulink models simply by drawing sketches on a computer screen. To support iterative design, Sim-U-Sketch allows users to interact with their sketches in real time to modify existing objects and add new ones.

For multi-touch tables, recently a diagram editing tool has been proposed in [24]. This tool provides already an intuitive way to create and interact with the displayed components - however, there are still more interactions required than in usual paper-work. For example, a note has to be created (either with a pen or hand gesture) before a handwritten text can be written down.

The COMIC system [25] is a large European project that studies multi-modal interactions in design applications using pen and speech. In multi-modal systems,
methods like mode detection [26] are already mature enough to be applied to improve usability. As for the retrieval of similar sketches, there exist also approaches in the literature. Yaner [27] examines the retrieval and mapping tasks of visual analogy. Finally, Spatial-Query-by-Sketch proposed by Egenhofer [28] describes a visual spatial query language for geometric information systems.

4. Methods

4.1. Indexing: Semantic fingerprint

The proposed semantic structure of our previous work is a selective semantic and is used to formalize the structure of a floor plan. Four main concepts to describe spaces of housing constructions and their relations were introduced:

- Room - the most atomic structure in a formal representation;
- Zone - consists of several rooms, for instance a sleeping zone;
- Unit - groups zones, such as apartment or terrace;
- Level - the current floor level of the building.

A building and the corresponding floor plan is separated into levels. Each level is divided into multiple units, which could be for instance an apartment or a terrace. Units can be further divide into zones. Examples for zones are living zones, sleeping zones and function zones. A zone groups different rooms which are the most atomic part of the structure.

We assume every extracted floor plan inherits a finite number of these features. Furthermore, these structural and functional features can be found in every project. The fingerprint contains features that serve as a meta-signature for the reasoning process. In addition to the usual dataset providing information about a project, every project is also given a semantic fingerprint, which inherits the spatial situation of the building in terms of its spatial boundaries and their characteristics. By providing a sketch of the required spatial configuration, the user creates a digital search fingerprint for the query. This search fingerprint can then be compared with the fingerprints in the database.

As each single floor plan contains a level of a building, the root node always is a level node. The level will be hierarchically divided into units, zones and rooms via part-of relations. The resulting structure would form a tree. Since the structural concepts of a layer can be connected either with a direct connection or an adjacent connection, we also add vertexes for these connections, resulting in a graph.

The semantic fingerprint structure proposed by Langenhan [4] represents the spatial, functional structure and is used to formalize the structure of a floor plan.
Four main concepts (Level, Unit, Zone and Room) are used to describe the spaces in a housing construction and their relations. Table 2 illustrates some entities corresponding to the concepts.

**Table 2. Taxonomy – Example of entities.**

<table>
<thead>
<tr>
<th>CONCEPTS</th>
<th>LEVEL</th>
<th>UNIT</th>
<th>ZONE</th>
<th>ROOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entities</td>
<td>Attic Floor</td>
<td>Circulation</td>
<td>Circulation</td>
<td>Bedroom</td>
</tr>
<tr>
<td>Upper Floor</td>
<td>Apartment</td>
<td>Living</td>
<td>Workroom</td>
<td></td>
</tr>
<tr>
<td>Ground Floor</td>
<td>Terrace</td>
<td>Function</td>
<td>Bathroom</td>
<td></td>
</tr>
<tr>
<td>Basement</td>
<td>Balcony</td>
<td>Sleeping</td>
<td>Kitchen</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Corridor</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Staircase</td>
<td></td>
</tr>
</tbody>
</table>

The instances of a concept with the same type can have either directly, adjacent or no relation. If two spaces have a shared wall and a door which links the spaces, we are talking about a connection. An adjacent relation is indicated by a shared wall without an alley.

To create a semantic fingerprint we propose a semi-automatic means of extracting floor plan features [29] by applying image recognition techniques [30, 32] along with machine learning methods that classify the structural information and present the results to the architect who then approves or modifies the fingerprint. With the support of semi-automatic extraction, an architect is able to formalize knowledge about past projects.

In order to separate textual and graphical information (see Figure 2), one of the first steps in the analysis of the floor plans is to extract the textual information from the graphical information. Tombre et al. [33] adapted the well known method of Fletcher & Kasturi [34] for separating text and graphics. As the method produces good results, it is used for the text/graphics separation. The main idea is to separate the image into two layers, a layer containing text - characters and annotations - and a layer containing graphical objects. The textual layer contains valuable information for the room classification and the graphical layer symbols and the walls. Thus optical character recognition (OCR) will be applied on the textual layer to extract this information.

At the current state, the document analysis system is focused on a limited type of floor plan (see Figure 2), where walls are symbolized by thick lines. Based on morphological filters [35] walls are detected and the physical limitations of rooms are automatically detected. For detecting the connection between rooms, doors symbol are very important. There is no visual standard for door symbols, but they
are usually depicted with an arc, hence detecting arcs is a good approach to detect doors.

Arcs are represented with thin lines the analysis is performed on thin layer. So the basic idea is to extract features from an arc and find these features in the image of the floor plan. As the arcs can be rotated or scaled in the image, the extracted features had to be invariant to rotation and scaling. SURF (Speeded Up Robust Features) is a robust scale- and rotation-invariant interest point detector and descriptor, introduced by Bay et al. [36] and it fulfills the requirements.

The room detection is based on the results of the arc detection and the wall detection, as an arc indicates a door symbol which symbolizes a direct connection between two rooms. The detection of the room itself is done by analyzing the wall polygons.

![Original floor plan, input for analysis.](image)
Basically, the idea of the room detection is to find openings in the walls, such as windows or doors. After finding these openings, they will be closed (see Figure 3). Now, by inverting the image, rooms are easy to detect by applying the contour finding algorithm used for wall detection (see Figure 4). In order to classified the connections between the rooms the openings in the walls are compared to the door candidates. If there is an opening and a candidate for a door
symbol the connection between the adjacent room polygons is marked as direct connection, otherwise as adjacent connection.

Currently, the detection of single rooms and their interconnections already gives reasonable results. Figure 4 illustrates first results of the automatic extraction algorithm. Current work involves classifying the type of room by using symbol recognition [30, 37] and OCR for each piece of textual information. Our system [38] is evaluated using a data set containing original floor plan images. This data set was introduced in [39] and contains 90 floor plan images from the period of more than ten years. The detection rate for the rooms in the floor plan is 89%. Furthermore, a rule-based system could be applied in order to group rooms into zones and zones into units.

The semi-automatic extraction of features from raster graphics shown above can likewise be used to analyze and index mass data available on the internet. This provides access to collective knowledge about architecture. Another approach is to use standardized intelligent data formats, already containing semantic information, such as the IFC, to extract a semantic fingerprint.

5. Retrieval: Sketch-based

Usability is the main aim of Human-Computer Interaction (HCI) research. It is essential for designing interfaces that allow users to intuitively work and interact with applications. Appropriate metaphors and devices have to be used to allow easy and fast interaction. Architects prefer to sketch in their initial design phase to create an image of the building in their mind.

We introduce a sketch-based visual-query-language based on the proposed semantic. The user interface focuses on specifying space and their relations by drawing them. This query style supports the spatial thinking approach that architects use, which often have a visual representation in mind without being able to provide an accurate description of the spatial configuration.

Figure 5 illustrates a web-based retrieval interface that can be used with standard input devices such as a mouse or keyboard. To create a schematic query the architect sketches a space using layers for the level, rooms, zones and units. The mock-up shows menu options for editing the schematic sketch in layout mode (1), space mode (2) and relation mode (3). Navigation within the layers and the actual spatial configuration (5) as well as specifying the semantic structure is done using the space navigator (4). The results (6) of the visual query show up on the right-hand side.
Several prototypes were implemented on a multi-touch table with combined touch and pen inputs to support semi-automatic extraction and intuitive sketch-based retrieval strategies. For the purposes of searching the repository and assessing the similarities between the sketch from the architect’s hand drawing and the repository, a similarity measure must be calculated. In graph theory this can be interpreted as subgraph matching. For the retrieval, the query graph $G$ has be compared with the set of graphs from the database. In order to calculate the similarity of a query graph and a database graph the edit cost could be applied.

As the pen and touch paradigm is more intuitive for the work of an architect, the prototype is being implemented for the Touch & Write table, which has been introduced in [40]. The Touch & Write table, illustrated in Figure 6, combines the paradigm of multi-touch input devices and a multi-pen input device. The table is a novel rear-projection tabletop which detects touching by using frustrated total internal reflection (FTIR) [41] and a high resolution pen technology offered by Anoto Group AB [42].

As already discussed in Section "Retrieval : Sketch-based" a visual query language is used to formalize the structural, functional query. The entities listed in Table 2 are represented by rectangles. Enclosing rectangles are interpreted as a part-of relationship. For instance, if a rectangle $R_1$ encloses other rectangles $R_2$ and $R_3$, it indicates that $R_2$ and $R_3$ are part of $R_1$, such as a sleeping zone which contains two bedrooms. How the units, zones and rooms are connected with each other is indicated by lines connecting the rectangles.
Two different connection types have to be considered, because two entities are either adjacent or directly connected. In the schematic view this is indicated by two parallel lines if the entities are directly connected or one line if they are only adjacent to each other. To summarize the semantics of the visual query language:

- Rectangles represent structural entities.
- Enclosings imply part-of relation.
- Single lines indicate adjacent connections.
- Two parallel lines indicate direct connections.

The retrieval interface provides a sketch-based approach to formalize the query. In analogy to the real world the user interface can be considered as a blank piece of paper were the architect sketches his initial ideas for the building. For instance he or she is looking for reference projects where several apartments are connected with a central corridor and at least one apartment is composed of a sleeping zone and a living zone. Now, after sketching his ideas, the architect searches for similar reference projects.

![Fig. 6. Retrieval interface on the Touch&Write table.](image)

In Figure 6 (a) a rectangle is drawn, which is automatically detected as an entity in Figure 6 (b) and the type is annotated by writing the correct term from the taxonomy in Figure 6 (c). Our initial approach is based on a simplified visual query language to investigate different levels of abstraction to focus on certain spatial aspects like in this case topologies and proportions. Thus the user is able to perform a query without knowing the exact position of the rooms. This approach is located between functional schemata like bubble diagrams and an illustration of a fixed spatial configuration like a floor plan. However, future work will be to extract the semantic structure from initial floor plan sketches of an architect and support other annotation inputs apart of functional descriptions. Thus an architect
does not have to learn a special query language, instead just use his natural way of sketching ideas.

The Touch&Write table - with its multi-pen and multi-touch support - enables a team to work together using touch- and pen-interaction. Each team member can simultaneously modify or extend the visual query using his/her own pen and the team can discuss the results of the retrieval. Future work will be to offer the team members possibility to annotate or rate the retrieval results. Either by using pen gestures, such as drawing rating stars, can be used to rate the result or after modifying a retrieved floor plan could be used as an initial concept. Reasonable modifications could be stroking out a part of the floor plan, adding additional rooms or enlarging respectively shrinking rooms.

5.2. Retrieval : Graph theory

Basically, the visual query will be internally represented as a graph structure. Now, the application searches for graphs in the repository which contains this substructure or a similar substructure. In graph theory a similar problem is known as Maximum Common Subgraph-isomorphism (MCS) which is known to be a complex computational problem. Thus the number of matched vertexes is a potential measure of similarity. The actual system searches for exact matches of this substructure in the graph repository. In order to perform this search the whole database in real-time, an index structure is determined a priori. We adapted Messmer’s method [43] for exact subgraph matching by modifying the generation process of the index structure. Using a well-founded total order on the graph structure extended the limitation of 19 nodes per graph [43], which is now depending on the number of nodes with the same label. In our case, labels are equivalent with the types in the semantic structure, such as AT for attic floor, AP for apartment or CI for circulation.

![Fig. 7. Retrieval process - Sample query graph.](image)
Furthermore, we adapted the retrieval process which uses a decision tree as an index. Figure 8 displays a decision tree for the graph database with two graphs containing the semantics of a floor plan using the proposed structure and a sample query graph in Figure 7. With this specific query graph, an architect would search for two directly connected apartments in an attic floor. Each leaf of the decision tree (Figure 8) is a possible alignment of the adjacency matrix representing a graph.

Thus all leaves beneath the last matching node of the search path, respectively the associated graphs, are added to the result set $R$. The number of exact row column vector matches $N$ for the query graph divided through the number of nodes of the query graph $|V_q|$ is used as a simple similarity measure.

$$\text{Sim}(q, R) = \frac{N}{|V_q|}$$

In our example the similarity would be 66.6% as only two of three comparisons match. Our current work focuses on researching further standard approaches for solving the MCS problem and the subgraph isomorphism respectively.

*Fig. 8. Retrieval process - Decision tree.*
6. Conclusions and future work

The retrieval is based on spatial configurations and semantic descriptions. The representation of architectural data in terms of semantic models is helpful to architects to enable them to use such information at a later date. Technologies such as semantic-based querying or even image-based querying will help to augment the design process.

The obvious outcome of the proposed semantic fingerprint is the ability to access valuable information. Furthermore, automatically analyzed fingerprints can put forward both commonly used as well as best practice projects. It will be possible to rate architecture according to the fingerprint of a building. Besides traditional 2D and 3D visualization techniques, other strategies such as diagrams, mental maps, visual thesauri or conceptual maps [44, 45] represent a useful extension to knowledge-based design in the early stages of the process.

The prototypical system based on the pen and touch metaphor realizes a collaborative knowledge management system which enables a team to work together in the initial planning phase. Each team member can participate in the discussion and modify the query with its own pen device. As the retrieval approach performs the search in real-time and presents the best matching results, the team can continuously refine their query by performing vague conceptual freehand sketches in a digital environment.

Future work will be to improve the interaction with the retrieved results and integrate a feedback system. Additionally the use of semantic-rich building information modeling (BIM) as knowledge container will optimize the reuse of the information. Furthermore the extension of the regarded spatial context will be a key feature of future developments. Apart of the topological and functional inner structure, the orientation of spaces as well as the urban integration, and the relation of buildings to each other will be focused by combining Building information modeling (BIM) and Geographic information system (GIS).

References

SKETCH-BASED METHODS FOR RESEARCHING BUILDING LAYOUTS …


101