

# a.SCArch - A Sketch-Based Retrieval for Architectural Floor Plans

Markus Weber\*<sup>†</sup>, Marcus Liwicki\*, Andreas Dengel\*<sup>†</sup>

\* German Research Center for AI (DFKI GmbH)  
Knowledge Management Department,  
Kaiserslautern, Germany  
{firstname.lastname}@dfki.de

<sup>†</sup> Knowledge-Based Systems Group,  
Department of Computer Science, University of Kaiserslautern,  
P.O. Box 3049, 67653 Kaiserslautern

## Abstract

*Architects' daily routine means working with drawings. They use either a pen or a computer sketching their ideas or drawing to scale. When beginning a new project they often have to search for similar projects in the past. In this paper a sketch-based approach is proposed to query the floor plan repository. The user searches for semantically similar floor plans just by drawing the new plan. An algorithm extracts the semantic structure sketched by the architect on DFKI's Touch & Write table and compares the structure of the sketch with the ones from the floor plan repository. The a.SCArch system enables the user to easily access knowledge from past projects. While in the current prototype only sketches with a predefined structure are recognized, we will extend the system to work with normal floor plans.*

## I. Motivation

During design processes architects refer to existing and already designed buildings as reference projects. These drawings are used to see how similar architectural situations were solved in the past. In order to find those similar projects, the architects have to browse through all previous projects or they apply some electronic search strategy.

However, the state of the art in electronic search is to search by textual annotations. The configuration of space and their relations to each other are hard to

be represented by keywords. Transforming these structural configurations into verbal expressed typologies produces fuzzy and often imprecise descriptions of architecture.

The concept a.vista suggested by Langenhan [7] addresses geometrical search strategies instead of today's keywords based search methods. A.vista investigates the limits of architectural spaces represented in drawings. By recording and saving the space boundaries to the database, the information about the specific project is transformed to both, the descriptive world of architecture and the computer. Every building in the database features a digital signature. The signature contains the architectural situation of the building constituted by the space boundaries and their characteristics. By sketching a situation of the demanded architectural configuration, the user creates a digital search signature for the query. The search signature will be compared with the signatures in the database.

As an abstract layer, a.vista by Langenhan [7] proposed a semantic structure to describe a signature of a floor plan within a graph representation. This formal representation is the foundation of the proposed a.SCArch system in this paper. It is a sketch-based user interface to query the signature repository by using an easy learnable visual query language. In future we will enhance the system to extract the semantic structure semi-automatically from usual queries.

The rest of this paper is organized as follows. First, Section II summarizes existing works in the area of sketch-based systems. Subsequently, Section III introduces the conceptual methods used for finding similar

architectural projects. Next, Section IV describes the concepts of the a.SCatch system and Section V evaluates the extraction of the semantic structure from the hand-drawn sketch. Finally, a summary and possible future directions can be found in Section VI.

## II. Related Work

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 [5] Feng proposed a 2D dynamic programming approach for analyzing hand-drawn electronic circuits. Sezgin et. al introduced in [13] 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 [6] 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. Finally Spatial-Query-by-Sketch proposed by Egenhofer [4] describes a visual spatial query language for geometric information systems.

In the offline domain, there exists system which automatically determines the type of the room by employing symbol recognition [14]. Furthermore, the textual information is identified by optical character recognition (OCR) [15]. To the authors' knowledge there exists no system which automatically analyzes architectural sketches to find similar projects from the past. Especially the use of a semantic layer was never investigated before.

The usage of meta data to enhance digital floor plans with additional information offers the opportunity to get intelligent files which allow the users an easier access to the planning material. One of the major approaches in architecture to enhance CAD models of buildings with semantic information is Building Information Modeling (BIM) [2], [3]. BIM is the process of generating and managing building information in an interoperable and reusable way. A BIM system is a system or a set of systems that enables users to integrate and reuse building information and domain knowledge through the life cycle of a building.

Nowadays modern architectural design is done using Computer Aided Design (CAD) software. Several vendors, such as Autodesk (AutoCAD)<sup>1</sup>, Graphisoft (ArchiCAD)<sup>2</sup> and Nemetek (Allplan)<sup>3</sup> offer software

<sup>1</sup><http://usa.autodesk.com/> : Last accessed 04/02/2010

<sup>2</sup><http://www.graphisoft.com/> : Last accessed 04/02/2010

<sup>3</sup><http://www.nemetschek.de/> : Last accessed 04/02/2010

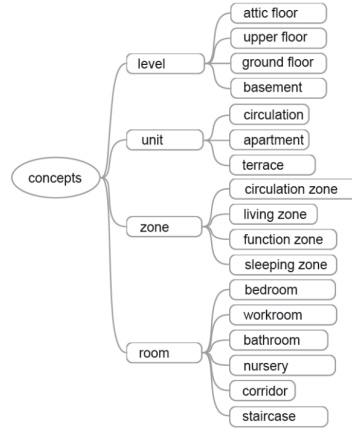


Figure 1: Taxonomy - Example for entities

packages with their own data formats to store information about the building. But according to BIM, interoperability is important to reduce costs and support all stakeholders. Furthermore, it is more natural for the architect to sketch with a pen and not with the mouse.

## III. Methods

The proposed semantic structure of Langenhan is a selective semantic and will be used to formalize the structure of a floor plan. Langenhan introduces four main concepts to describe spaces of housing constructions and their relations following the paradigm of the incremental space:

- 1) **Room** - the most atomic structure in a formal representation;
- 2) **Zone** - consists of several rooms and describes the functionality of the grouped rooms, for instance a sleeping zone;
- 3) **Unit** - groups zones and has also a functional meaning, such as apartment or terrace;
- 4) **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.

But today there is not always a strict division of the function that spaces can serve. Thus a single physical space can have several functions and thus be a combination of multiple functional spaces, such as a living room combined with a kitchen. Examples for structural entities are given in Fig. 1.

Between these concepts, there exist connections. 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.

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 (see Section IV-C).

#### IV. Concept a.SCatch

Based on the results of Langenhan, the aim of a.SCatch is to implement a system which takes advantage of the semantical information. A semantic search will be realized by sketching a concept of an architectural problem and triggering a search for similar projects of the past.

Therefore several subtasks have to be solved:

- 1) Semi-automatic extraction of the semantic structure of older projects,
- 2) Extraction of the semantic structure from the sketch of the architect,
- 3) Retrieval of similar floor plans from the repository,
- 4) Visualization of the results and the interaction with the user interface.

The following sections will discuss the further details of the a.SCatch system.

##### A. Semi-automatic Extraction from Existing Projects

A subtask of the a.SCatch project is to research techniques to extract the graph representation either from graphic representation by applying state of the art image understanding techniques [1], [9], [10], [12] or extract the information from known architectural model file formats, such as *Industrial Foundation Classes*.

In this context, semi-automatic extraction is defined as a process where the system tries to automatically extract features from a floor plan by using image understanding techniques and apply machine learning methods that classify the structural information. The extracted structure will be proposed to the user and he needs to approve the result or modify them via a pen- and touch-enabled interface.

With the support of semi-automatic extraction an architect is able to formalize knowledge about past

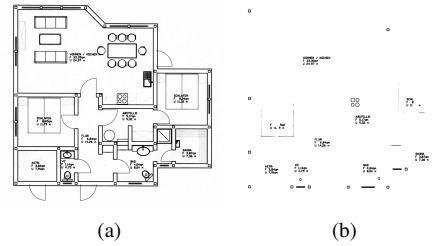


Figure 2: Example for text/graphics separation

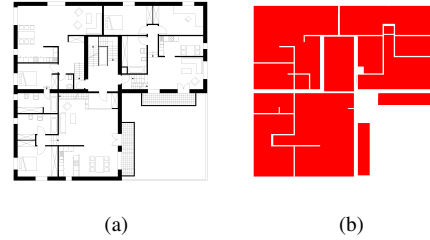


Figure 3: Example for room detection

projects. This formalization process comprises two phases. The automatic analysis is the first part of the analysis and is mainly based on the techniques discussed in [1]. Currently the focus is on the detection of single rooms and their interconnections. Future work will be to classify the type of the room by employing symbol recognition [14] and optical character recognition (OCR) on the separate textual information [15]. Furthermore a rule based system could be applied in order to group rooms in zones and zones into units. The floor plan with its extracted semantics will be stored in a repository and a user interface is provided to manually annotate the floor plan. Results of the room detection and the text/graphics separation are illustrated in Fig. 2 and Fig. 3.

##### B. Sketch-based Retrieval

As we are dealing with visual information and an exclusive textual description of floor plan is too fuzzy, we propose a visual query. Whenever the architect is searching the repository, he formalizes his query as a sketch, similar to the fundamental concepts of Spatial-Query-by-Sketch proposed by Egenhofer [4]. Initially the architect sketches a floor plan with the associated rooms, zones and units. The respective online data of the pen device used to detect the geometrical shapes representing the concepts and gestures, which indicate the connection type.

The schematic abstraction in Fig. 4a shows a floor plan translated into the proposed semantic structure by Langenhan [7]. Enclosing rectangles are interpreted as a part-of relations. For instance, if a rectangle  $R_1$

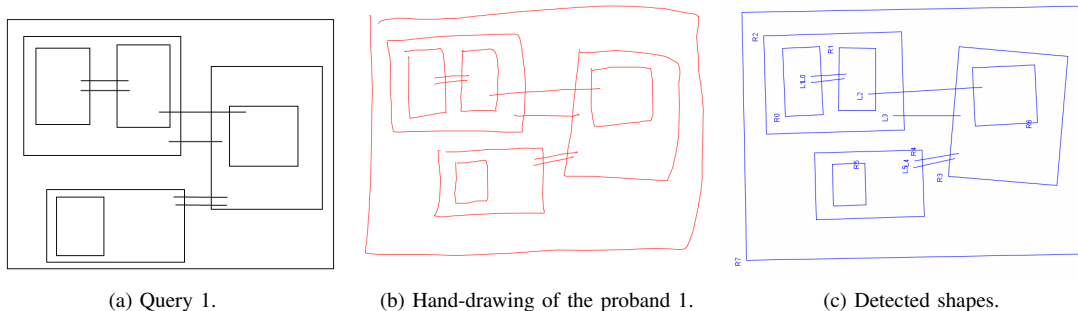


Figure 4: Example for a floor plan and its sketch

encloses another rectangle  $R_2$  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. As discussed in the Section III two different connection types have to be considered. Either two entities are adjacent or directly connected. In the schematic view this is indicated by two parallel lines for the direct connection and one line for adjacent connection. By interpretation of the sketch, a graph structure can be extracted for our query.

Sketch recognition is performed using the following steps. During sketching the online pen data is cached. As soon as the user stops drawing for a certain period of time, the recognition of the previous components starts. Since drawing of rectangles often is performed with several lines, succeeding lines are combined if one of their ending points is in the same vicinity. For the shape detection we used the Vision Objects shape detection<sup>4</sup>.

Detected shapes will immediately replace the strokes in the user interface and the user can interact with these shapes via touch gestures (moving, resizing). If a stroke is not recognized as a shape it won't be replaced. Thus the user has feedback which part of the drawing is not recognized.

Currently we use the following visual query language:

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

Finally when the user triggers the search the scene which is composed of rectangles and lines will be analyzed. For the part of relation we simply have to check if a rectangle encloses another one. The type of the drawn entity or connection currently is defined by the user. The following section will discuss how the

extracted graph structure can be used for the retrieval.

### C. Graph Structure

In this section we propose our initial concept for the retrieval. The extracted semantics are represented as a graph  $G = (V, E)$ . The vertexes  $V$  have a type  $T_{vertex}$  reflecting a level, unit, zone or room. Each of these types has finite set of subtypes (cf. Fig. 1). As the types are hierarchically ordered  $T_{level} > T_{unit} > T_{zone} > T_{room}$ .

The vertexes  $E$  also have different types  $T_{vertex}$  indicating if the vertexes are connected directly or are just adjacent, both of these relations are symmetric. The *part\_of* relation indicates which vertex adheres to a vertex of a superior type  $T_{vertex}$ , for instance a sleeping room which is part of a sleeping zone. As we are dealing with undirected and directed vertexes the graph is a mixed graph.

The types of the node and vertexes are assigned by labeling functions  $\alpha : V \rightarrow T_{vertex}$  and  $\beta : E \rightarrow T_{vertex}$ .

The retrieval algorithm is based on a subgraph matching algorithm, where the row column vectors of the adjacency matrices are arranged in a decision tree. A modified version of Messmer and Bunke's algorithm [11] has been implemented to compile the decision tree. A priori, each extracted graph structure is added to the decision tree. During the retrieval process, the row column representation of the query graph is determined and the sub graph is matched by traversing the decision tree.

Our current work focuses on researching approximate approaches for solving the subgraph matching problem. Furthermore we are researching techniques to cluster the graph in our repository, by comparing floor plan graphs among each other and group similar ones together. Thus, if a query is not similar to a graph of a cluster it might also be not similar to the other graphs in the cluster.

<sup>4</sup><http://www.visionobjects.com/>: Last accessed 04/02/2010

## D. User Interface

As the pen and touch paradigm is more intuitive for the work of an architect, the prototype is implemented for the *Touch & Write*. The *Touch & Write* table [8] combines the paradigm of multi-touch input and pen input. Architects prefer to sketch in their initial design phase. A pen gives them more freedom than using a mouse with Computer Aided Design (CAD) software. Using the *Touch & Write* pen device to draw in a digital environment allows more immediate interaction and the architects immediately benefit from the digital representation of their drawings.

The pen is an adequate tool to sketch the current architectural problem. The results of the semantic search will be represented as graphical information and the touch interaction is an intuitive metaphor for interacting with the displayed information. For example, the architect is able interact with the graphical information using simple and intuitive gestures to zoom or navigate within the floor plan.

The a.SCatch system provides tools for two purposes. First, an input interface offers the architect a possibility to edit the results of the automatic room and interconnection detection, discussed in Section IV-A. Here the pen device is used to frame rooms, zones or units in a floor plan. The currently selected structural entity affects how the node in the semantic representation is labeled.

Second, the architect can sketch a query by using the discussed visual query language. The results are displayed as images ordered according to the calculated similarity measure. The interaction with the result is done by using touch interaction gestures.

## V. Experiments and Results

The query generation is a crucial part of the a.SCatch system, thus the first experiments focus on the visual language. We defined ten example queries covering different complexity levels (see Table I) and asked ten probands to copy these sketches, resulting in a total of 100 sketches. The probands were male and female students in the age between 23 and 29 years. All sketches were drawn on the *Touch & Write* and the handwritten strokes were recorded. To assess the pure recognition performance, we did not give a direct feedback of the recognized shapes. In order to measure the accuracy of the detection algorithm we count the correctly detected quadrangles and connections.

An example of a query taken for the evaluation is given in Fig. 4a. Furthermore Fig. 4b shows the

Table I: Complexity and Detection rate for each query

Query	Quadrangles		Adjacent C.		Direct C.	
	#	Corr	#	Corr	#	Corr
1	8	0.96	2	0.95	2	0.8
2	5	0.96	2	0.9	3	0.9
3	8	0.93	2	0.95	5	0.7
4	6	0.95	3	0.93	1	0.7
5	3	1.0	2	1.0	1	1.0
6	9	0.99	3	0.97	5	0.96
7	6	0.98	2	0.95	4	0.65
8	9	0.99	4	0.93	4	0.9
9	10	0.97	1	0.8	4	0.9
10	3	0.97	1	0.9	2	0.9
Overall		0.97		0.93		0.86

Table II: Detection rate for each proband

Proband	Quadrangles	Adjacent C.	Direct C.
1	1.0	1.0	0.97
2	0.96	0.82	0.81
3	0.93	0.82	0.77
4	0.91	0.95	0.74
5	0.99	1.0	0.87
6	1.0	0.9	0.97
7	0.97	0.95	0.83
8	0.97	0.95	0.84
9	0.99	0.95	0.94
10	0.97	0.91	0.84

recorded sketch of a proband. The detected shapes of the shape detection algorithm are illustrated in Fig. 4c.

For the evaluation we distinguished between detection rates for quadrangles, adjacent and direct connections. Note that whenever a quadrangle is not detected all corresponding connections with this quadrangle will also be wrongly detected.

Table I shows the detection rate for each query. As can be seen, the detection rates for the quadrangles are very promising. Most of the other errors are due to missing quadrangles. Considering Query 2, 5, and 10, one can see that a high recognition rate of the quadrangles corresponds to a high recognition rate of the connections.

The detection rate for each proband is given in Table II. An interesting observation is that there are persons who have a very nice way of drawing which is easily to be recognize. Other users tend to produce gaps in-between the strokes or draw very rough quadrangles only, which are hard to be recognized, as the following example will show.

An example for a false detection is given in Fig. 5c. As several probands had different strategies of drawing rectangles, the current methods seem not to cover all of them. Especially if a long pause has been made during sketching a rectangle, the lines are not merged together. Figure 5c also illustrates that both connections of the rectangle are not recognized.

As the interactive system acknowledges correct de-

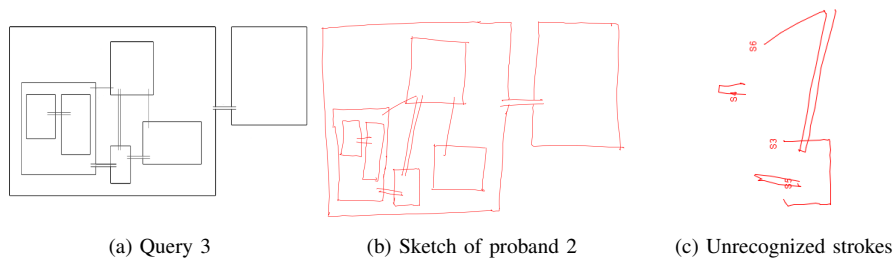


Figure 5: Visualization of the results

tected shapes, the user of the system has the chance to correct misinterpreted drawings before triggering a search. It is a good result that only one out of twenty rectangles needs to be corrected, making it practically useful.

## VI. Conclusion and Future Work

In this paper we have presented an intuitive system for searching floor plans by using a sketch-based interface. Furthermore we proposed a graph based semantic structure to capture the content of floor plan and associated visual query language.

While the final a.SCatch system is not available yet, we performed already first experiments on the recognition of abstract sketches. The recognition rates are already good for the use in practice. However, there is still room for improvement. For example, a dynamic programming approach to combine strokes [5] or using a combination of an online and offline detection should help to improve the detection rates.

Further work for the interactive system will be to offer the architect the freedom to choose between the proposed visual query language or let him sketch an initial floor plan with his symbolism and extract the semantic graph structure directly from this sketch.

The a.SCatch system is a successful approach integrating state-of-the-art offline image processing and online sketch recognition technologies. Together with the use of recent progress in knowledge management, it provides a novel powerful tool for sketch-based document work which can be applied to other application areas as well.

## References

- [1] P. Dosch, K. Tombre, C. Ah-Soon, and G. Masini. A complete system for the analysis of architectural drawings. *International Journal on Document Analysis and Recognition*, 3(2):102–116, 2000.
- [2] C. Eastman. *Building product models: computer environments supporting design and construction*. CRC, 1999.
- [3] C. Eastman. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Wiley, 2008.
- [4] M. J. Egenhofer. Spatial-query-by-sketch. *Visual Languages, IEEE Symposium on*, 0:60, 1996.
- [5] G. Feng, Z. Sun, and C. Viard-Gaudin. Hand-drawn electric circuit diagram understanding using 2D dynamic programming. *Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition, Québec*, pages 493–498, 2008.
- [6] L. B. Kara and T. F. Stahovich. Sim-u-sketch: a sketch-based interface for simulink. In *AVI*, pages 354–357, 2004.
- [7] C. Langenhan. *a.vista Semantische Suche Web 3.0 für die Architektur*, 2008.
- [8] M. Liwicki, S. El-Neklawy, and A. Dengel. Touch & Write - A Multi-Touch Table with Pen-Input. In *Proc. Int. Workshop on Document Analysis Systems*, 2010.
- [9] T. Lu, H. Yang, R. Yang, and S. Cai. Automatic analysis and integration of architectural drawings. *International Journal of Document Analysis and Recognition (IJ-DAR)*, 9(1):31–47, 2006.
- [10] S. Mac, E. Valveny, H. Locteau, and T. S. A System to Detect Rooms in Architectural Floor Plan Images. In *Proc. Int. Workshop on Document Analysis Systems*, 2010.
- [11] B. Messmer and H. Bunke. A decision tree approach to graph and subgraph isomorphism detection. *Pattern Recognition*, 32:1979–1998, 1999.
- [12] S.-h. Or, K.-h. Wong, Y.-k. Yu, M. M.-y. Chang, and H. Kong. Highly Automatic Approach to Architectural Floorplan Image Understanding & Model Generation. *Pattern Recognition*, 2005.
- [13] T. M. Sezgin, T. Stahovich, and R. Davis. Sketch based interfaces: early processing for sketch understanding. In *In Proc. workshop on Perceptive user interfaces*, pages 1–8. ACM, 2001.
- [14] S. Tabbone, L. Wendling, and K. Tombre. Matching of graphical symbols in line-drawing images using angular signature information. *Int. Journal on Document Analysis and Recognition*, 6(2):115–125, 2003.
- [15] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch. Text/graphics separation revisited. *Lecture Notes in Computer Science*, page 200211, 2002.