

## Views on architecture: Different abstraction layers of building information imply special working methods and interaction metaphors to support a variety of courses of action.

**Langenhan, Christoph**

Technical University of Munich, Germany  
langenhan@tum.de

**Petzold, Frank**

Technical University of Munich, Germany  
petzold@tum.de

### Abstract

In the early design phases, designers usually only have a vague idea of the building they are designing. Different aspects of potential design variants need to be tested, assessed and compared with one another. The ability to consider design variants in different ways – based on floor plan, schematic concept, section or 3D visualization, for example – helps to identify problems as well as reveal areas of potential. This paper describes software prototypes developed to help designers input and present different levels of abstraction. The transformations between these levels of abstraction reveal new solutions and make designers aware of issues that need to be considered during the design process.

**Keywords:** Semantic fingerprint, Building Information Modeling, HCI, retrieval

### Introduction

As the complexity of building tasks and requirements increases, designers often find themselves confronted with interdisciplinary problems that go beyond the specific challenges and methods of architecture and engineering. The iterative nature of the design process results in a continuous exchange between creative, analytical and evaluative activities, through which the designer explores and identifies promising design variants. The ability to compare and evaluate relevant reference examples of already built or designed buildings helps designers assess their own design explorations and informs the design process.

The design process is an iterative process (Buxton, 2007) of searching for a plausible solution involving a continual back and forth in which potential solutions are developed by various means before narrowing down the selection to the most promising candidates. Architects typically work with traditional design tools in the early design stages, such as model-making, sketches and the use of reference examples. Using reference examples is an acknowledged method in both architectural education and architectural practice and helps in learning design principles and guiding the design process, as well as for inspiration or even as an explicit solution (Richter, 2010).

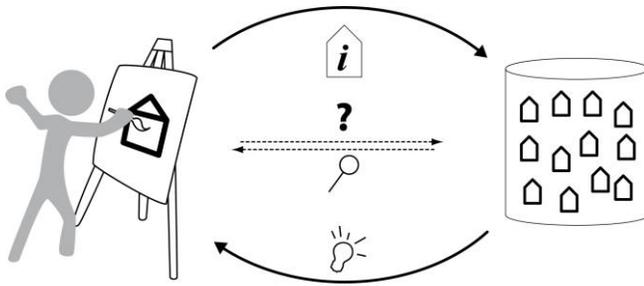
Most computational search methods available today rely on textual rather than graphical approaches to representing information. However, textual descriptions are not sufficient to adequately describe spatial configurations such as floor plans. To address these shortcomings Langenhan et al. (2013) introduced a novel approach which facilitates the automatic lookup of reference solutions from a repository using graphical search keys.

For indexing and determining similarities, the use of semantic fingerprints (Langenhan & Petzold, 2010) has been proposed as a way of describing the arrangement of buildings in a

manner analogous to the way that fingerprints can be used to identify a person. The system derives semantic fingerprints – representing e.g. accessibility and adjacency as features for search criteria – from a reference solution that describes the spatial relationship of rooms extracted from building information models. This forms the basis for assessing the similarity of different reference solutions to a specified problem and serves, accordingly, as an index for the building model repository.

In contrast to the fixed characteristics of human fingerprints, semantic building fingerprints can communicate multiple characteristics such as geometry, topology, orientation of spaces or elements, or energy efficiency characteristics. A building can have different fingerprints that describe sub-aspects of the building and are stored as consistent semantic data, for example for search queries and analyses.

In the application area described in this paper (see figure 1) for a means of researching semantic building information models (BIM) using hand-drawn search queries, the system assists the designer by analysing sketches made in the early design phases and deriving a structure that can be compared with the fingerprints in the information repository. Using a graphical user interface, the designer can sketch room configurations as a bubble diagram, freehand sketch or schematic digital layout. On the left-hand side (figure 8), the user can switch between the semantic fingerprints derived from the sketch, for example to focus on the passage through spaces, the adjacency of spaces or the degree of natural illumination, in turn influencing the search results shown on the right-hand side (figure 8).



**Figure 1.** An application scenario for knowledge-based design.

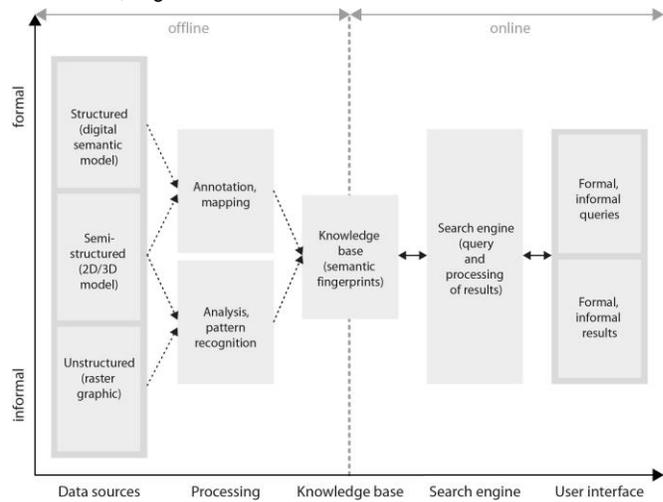
The intention is to provide architects with inspiration and/or specific solutions from comparable existing buildings or building designs as a means for assessing their own design ideas and approaches. Such information can take the form of images, descriptions or digital semantic building models, which all help to understand the search results and inform the designer's own design explorations. Information can, for example, be used directly with a self-developed "Dolphin" Add-on for 'Rhino3D' and the 'Grasshopper3D' parametric design environment.

## Software architecture

To realise the semantic search engine, we propose a system of data integration using a federated information system in which different autonomous sources of information (e.g. a BIM server, graph database and CMS) can be integrated using a common XML-scheme and queried as a single information source using a REST approach. "Data integration is the problem of providing unified and transparent access to a set of autonomous and heterogeneous sources, in order to allow for expressing queries that could not be supported by the individual data sources alone." (Keim et al. 2010, p. 23). This makes it possible to store and process information efficiently according to their specific properties.

In contrast to other "data warehouse" approaches, a federated information system does not copy the various sources but rather queries the respective sources individually using processing components and bundles the results for further processing by the coordinated system of components in the client applications. For offline processing and extraction of building fingerprints, components are needed that a) analyse and augment unstructured data sources (CMS) and b) components that extend and attribute information to concepts in structured data sources (BIM).

Figure 2 shows the software architecture of the information system which draws on the software architecture of the semantic search engine by Dengel (2012, p. 243). Topological information is extracted offline from formal (BIM) and informal (CMS) data sources using various methods and formalised in the form of graphs (fingerprints). The offline processing must structure the data in such a way that the user is presented with useful results within a reasonable time frame during online processing.



**Figure 2.** Prototypes for formalising a mental model of building designs for the purpose of research reference examples.

The components shown are groups of individual applications in the information cloud (server) and on different end devices (clients). The system is comprised of the following components:

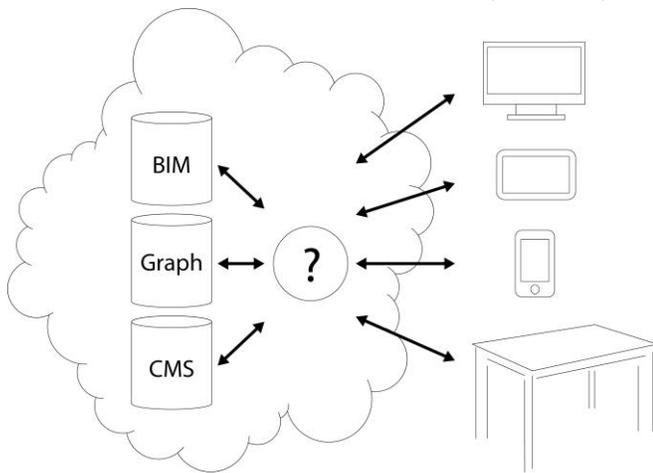
### Server applications

- Data storage
- Processing
- Coordination

### Client applications

- Desktop with mouse and keyboard (stationary workstation, e.g. in the office).
- Tablet Computer (transportable device, e.g. for client visits and meetings).
- Smartphone (mobile, e.g. on-site use or site research).
- Multi-touch table (stationary design environment).

The applications on the various end devices query the information cloud (figure 3). The information system for assisting the designer in the early design phases comprises a series of components for data storage and processing in order to store and process semantic building model data, graph data and text.



**Figure 3.** Information Cloud.

The application contexts and individual design processes differ from case to case. Different abstractions are therefore necessary to allow the designer to communicate his or her mental model as completely and accurately as possible. Because designers often only have a vague and incomplete notion of the design in the early design phases, the system needs to accommodate a high degree of freedom of expression and types of abstraction.

## Different layers of abstraction

The system has to provide interaction approaches that allow the user to formulate the mental model that can be analysed and compared with stored model schemes to produce search results. Therefore, contentual coherences between the mental model and the representations of information is necessary. Accordingly, suitable strategies need to be implemented to achieve a beneficial outcome for the user.

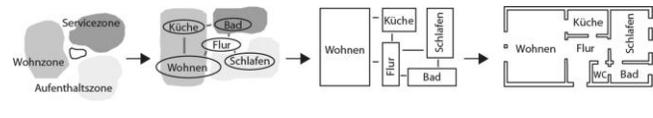
“Guidance is often required, especially for novice users, on what visualisations (scatterplot, parallel coordinates, treemaps, etc.) are appropriate for a given task; the focus should be on the problem to be solved rather than the type of raw data.” (Keim et al. 2010, p. 123). To make it easier for users to use and understand, domain-specific presentation methods need to be provided along with strategies for improving the ability to describe and communicate ideas (Roth-Berghofer und Richter 2008). The degree to which the user is involved in the process can be changed, for example, depending on the type of user, to show the user how the results have come about.

In addition, approaches are needed that present the user not just with results, but also show the often numerous and interdependent criteria in a transparent and understandable way (Gratzl et al. 2013) so that the user can adjust these as required. Keim summarises these requirements as follows:

- “Progressive analysis: provide quick answers first, then make improvements incrementally or on-demand;
- Management of dynamic data: incremental analysis instead of restarting it from the beginning;

- Steerable analysis: allow long-computations to be steered by users when possible.” (Keim et al. 2010, p. 106)

The aim is to gradually narrow down possible solutions or reference cases by providing ongoing feedback so that the query can be successively steered and defined by the user. An example of this is shown in figure 4 for building information in the early design phases. The degrees of detail shown are the product of thinking processes that gradually transform a design and make it more concrete.



**Figure 4.** Typical presentation schemes and their possible transformations, based loosely on (Hanson 1998, p. 308) and (Lieblich 1994, p. 75).

Corresponding interaction strategies for geometric, topological, geographic and lexical information are likewise required. In the information system we have implemented, we differentiate between five main representations for formalising mental models according to their respective characteristics:

- Texts (for example architect, year of completion, building costs, text search, descriptions, building typology)
- Tables (for example schedule of rooms, schedule of works, cost plan, specification of works, list of neighbours, access routes)
- Schemes (for example spatial diagrams, spatial arrangement, zones, orientation, proportions, passage through a room, adjacent relation to other spaces)
- Freehand sketches (for example, arrangement of spaces, zoning, orientation, proportions, passage through a room, spatial delineation, floor plans, elevations, sections)
- 2D/3D drawings (for example, arrangement of spaces, zoning, orientation, proportions, passage through a room, adjacency relationships, cubature, floor plans, elevations, sections, perspectives).

The transformation between different digital formalisations and presentations and their respective levels of abstraction is of particular help in the design process and is a topic of ongoing research. In the following section, we describe software prototypes for tablets, multi-touch devices or smartphones with touch input as well as for computers with mouse/keyboard entry. The aim is to examine the technical possibilities for computer-based interpretation of user input.

## Prototypes

The ability to add detail to or modify one’s input, for example by adding a room to a drawing, makes it possible to formulate more granular fingerprint-queries and the mental model corresponds better to the designer’s own mental model. For this purpose, we implemented separate prototypes that use

the same data stock to compare different input and retrieval strategies.

By providing ongoing feedback, e.g. in the form of reference projects that match the fingerprint, the information system prompts the designer to modify and adapt his or her design idea. By increasing or decreasing the degree of specification / generalisation, the set of possible search results can be expanded or focussed to better match the problem at hand.

What density of information is adequate and necessary depends on the specific application at hand and the user? For the general definition of spaces, their relationships and the passage between them, a simple graph can be sufficient. Graphs can take the form of bubble diagrams or tables (Beck et al. 2014, p. 88). In our information system, we have used node and edge diagrams to make it easier to depict the location and relationship between rooms or nodes.

Our user interface investigations looked at ways of providing vague input about spatial situations and constellations, i.e. their basic characteristics and arrangement. The corresponding mathematical formulation of this as a graph is shown to the user at different levels of abstraction.

The different use cases (e.g. in the office, when visiting a client, in a meeting, out on site or while designing) are conceived as prototypes for the respective user. For the most common office drawing situation with mouse and keyboard, we propose using the 'a.vista' concept previously developed by Christoph Langenhan in 2008 (Langenhan und Petzold 2010), which has different semantic ways of describing a building (level, unit, zone, and room).

The "ar.searchbox" (figure 5) is a media reference collection from the "mediaTUM" (Langenhan et al. 2012) that was devised as a research tool for buildings. The university library and computer science students implement changes while students of architecture at the TUM enter and maintain data in the system. It is possible to search using traditional search terms in the browser and the entries are linked back, based on an analysis of the pixel images, building models and links, to other information sources, such as the architects' website or a digital building model.

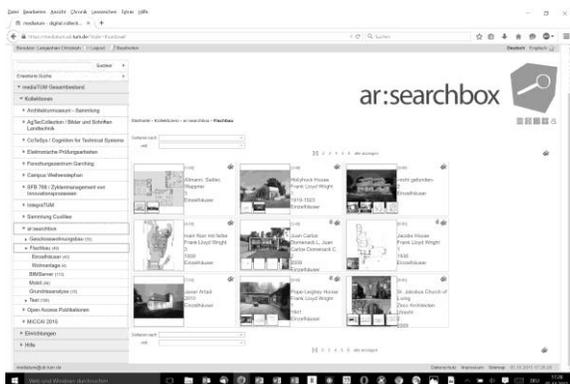


Figure 5. User interface of 'ar:searchbox'.

For the 'metis WebUI' (figure 6), Johannes Bayer at the Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI GmbH) took a platform-independent 2D modelling approach and implemented, among other things, a means of enriching search queries (Bayer et al. 2015). In figure 6 one can see the spatial arrangement as a bubble diagram on the left and actual spatial arrangement with symbols for doors and windows on the right, demonstrating the different formalisations of levels of abstraction and detail.

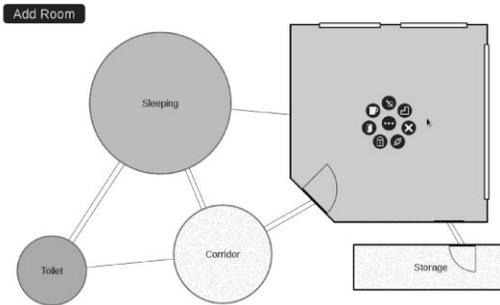


Figure 6. User interface of 'metis WebUI'.

An approach for formalising queries as freehand sketches was developed by Markus Weber in 2009 (Weber et al. 2010) with 'a.scatch' (figure 7). As with the method shown in figure 6, doors are drawn as double lines and adjacent walls as single lines. The rooms (figure 7, left a) are recognised and the room type labelled by hand (figure 7, left c). The resulting room (figure 7, left b) and be used for the search query and a list of corresponding search results is then displayed (figure 7, right).

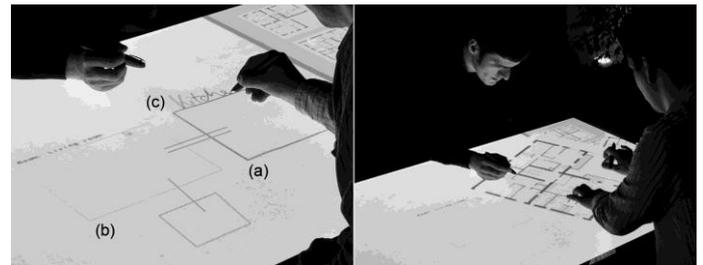


Figure 7. User interface of 'a.scatch'.

The application 'Touchtect' (Figure 8) by Thomas Kinnen and Dario Banfi for a multi-touch table combines freehand drawing, geographic input and meta information from the ar:searchbox (Figure 5). Here, however, the room type is not selected from a predefined list, nor is it recognised by means of a hand-written label.

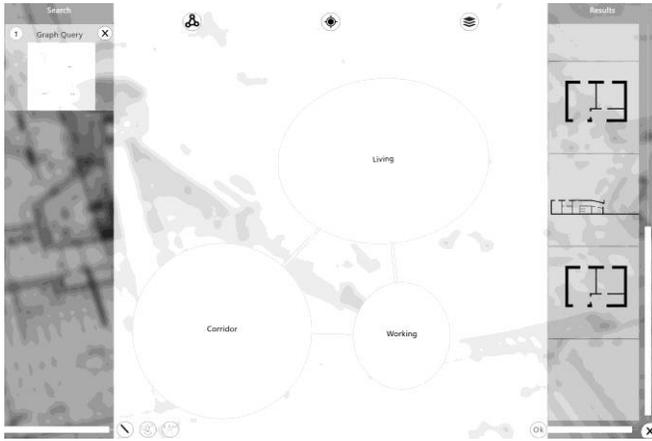


Figure 8. User interface of 'Touchtect'.

The application 'ar:searchDroid' (Figure 9) by Sebastian Seitz is designed for a tablet computer for mobile research and can restrict search results to buildings that are in the vicinity.

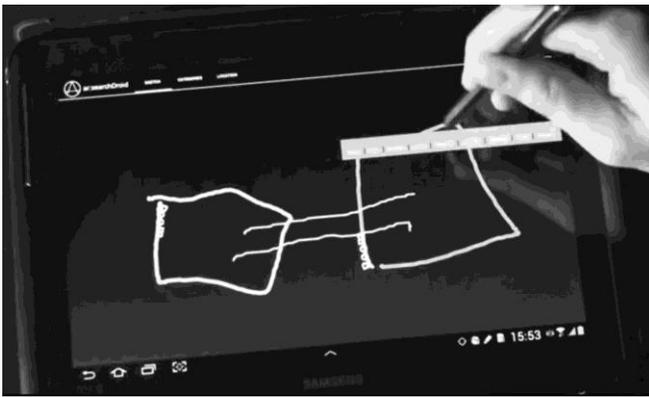


Figure 9. User interface of 'ar:searchDroid'.

All of the above applications for freehand drawings do not detect what has been drawn by visually processing the results but by detecting the movement of the stylus. As such, a polygon denoting a room needs to be closed in order to be recognised for use in the spatial query.

To afford the user greater flexibility while drawing, the 'cormorant' application (Figure 10) by Dominic Henze uses the \$P program library (Vatavu et al. 2015) for gesture recognition developed by the University of Washington.

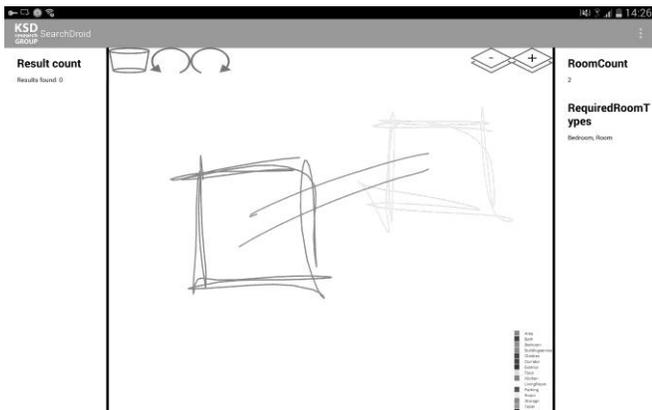


Figure 10. User interface of 'cormorant'.

The application 'ar:searchPad' by Jana Pejić (Figure 11) does away with the input of geometric figures altogether. This application is based entirely around the input and comparison of bubble diagrams.

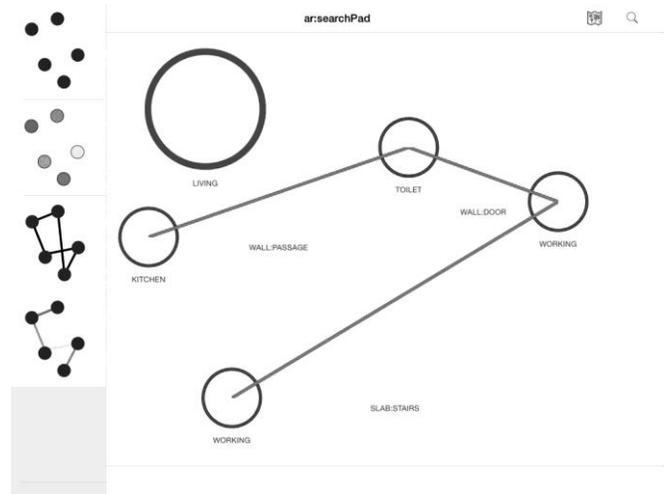


Figure 11. User interface of 'ar:searchPad'.

Similarly, the smartphone applications 'ar:searchbox.app' (Figure 12, left) by Alexander Sahn, and 'mediaTUM4android' (Figure 12, right) by Patrick Bernhard, both provide no means of drawn input. Instead they focus on local research using traditional search terms and the building up of a database of information. A student can create an entry for a building in 'ar:searchbox' (Figure 5) while on a field trip, and others can then augment the information with images and photos. When the student returns home, additional files can be added, for example a digital building model or plans.



Figure 12. User interface of 'ar:searchbox.app' (left) and 'mediaTUM4android' (right).

To incorporate such topological approaches in the design process, Thomas Stocker, Dario Banfi, Jana Pejić, Thomas Kühner, Markus Dausch, Bishwa Hang Rai, Dominic Henze,

Arno Schneider and Johannes Roith have jointly developed an Add-on call 'Dolphin' (Langenhan et al. 2014) for 'Rhino3D' and its parametric design extension 'Grasshopper3D'. The 'Dolphin' Add-on provides a series of components for visual programming using 'Grasshopper3D' that make it possible to query the information system using drawings in 'Rhino3D' and additional meta information. The data can be exported for further use directly from the graph database using a service called 'pigeon' developed by Leon Höß and Christopher Will. For example, AgraphML files can be imported with other components, either individually or as a collection, into 'Grasshopper3D' for further use.

The 'Dolphin' Add-on makes it possible to directly combine topological, lexical and geometric queries for the information system. The results, or the topological or building geometry information or semantic building model can then be used to inform the design process. The information can also be used as a basis for generating design variants using 'Galapagos' for 'Grasshopper' or as a means of automatically evaluating one's own design ideas.

## Conclusion

At the level of the user interface, a range of different input strategies for supporting the early phases of the design process and the input of vague drawn forms or models (Bayer et al. 2015) were explored as part of the 'metis' research project at the DFKI and TUM. The results show that the usefulness of the displayed results and input methods depends on what is being designed and the design strategy of the user. For example, if the user has a vague image of the building in mind schematic input and representation strategies like in Fig. 7 to 10 are more useful.

Goldschmidt and Smalkov have posed the underlying question of whether visual thinking is derived from mental processes or from preceding visual images "[...] are inner representations, using imagery, the prime generator of visual thinking in designing, or are external representations, in the form of drawings of all sorts and other two- and three-dimensional representations, indispensable to design thinking?" (Goldschmidt and Smalkov 2006, p. 549). The history of architecture shows that design tools have influenced how the built environment was made. The influence of tools on the thinking process is, however, hard to measure. Gänshirt has argued that "[...] every program implies a more or less concealed 'ideology'" (Gänshirt 2007, p. 193) which conditions every object constructed with them.

## Acknowledgments

This work was supported by the DFG (Deutsche Forschungsgemeinschaft, German Research Foundation). We would also like to thank the students from the TU Munich who took part in our interdisciplinary projects and made this work possible: Alexander Sahm, Patrick Bernhard, Thomas Kinnen, Sebastian Seitz, Thomas Stocker, Dario Banfi, Jana Pejic, Thomas Kühner, Markus Dausch, Bishwa Hang Rai, Dominic Henze, Arno Schneider and Johannes Roith.

## Reference literature

- Bayer, J.; Bukhari, S.; Langenhan, C.; Eichenberger-Liwicki M.; Althoff K.-D.; Petzold, F.; Dengel, A. (2015). Migrating the Classical Pen-and-Paper based Conceptual Sketching of Architecture Plans Towards Computer Tools. *Prototype Design and Evaluation*. In: GREC 2015.
- Beck, F.; Burch, M.; Diehl, S.; Weiskopf, D. (2014). The State of the Art in Visualizing Dynamic Graphs. In: *EuroVis - STARS: Eurographics Association*, S. 83–103. Online: <http://dx.doi.org/10.2312/eurovisstar.20141174>.
- Gänshirt, C. (2007). *Werkzeuge für Ideen. Einführung ins architektonische Entwerfen*. Basel: Birkhäuser. Online: unter <http://www.gbv.de/dms/hebis-darmstadt/toc/187535930.pdf>.
- Goldschmidt, G.; Smalkov, M. (2006). Variances in the impact of visual stimuli on design problem solving performance. In: *Design Studies* 27 (5), S. 549–569, zuletzt geprüft am 27.09.2011.
- Gratzl, S.; Lex, A.; Gehlenborg, N.; Pfister, H.; Streit, M. (2013). LineUp. Visual Analysis of Multi-Attribute Rankings. In: *IEEE Trans. Visual. Comput. Graphics* 19 (12), S. 2277–2286.
- Hanson, J. (1998). *Decoding homes and houses*. Cambridge ;, New York: Cambridge University Press.
- Keim, D.; Kohlhammer, J.; Ellis, G.; Mansmann, F. (2010). *Mastering the information age. Solving problems with visual analytics*. Goslar: Eurographics Association.
- Langenhan, C.; Petzold, F. (2010). The Fingerprint of Architecture. Sketch-Based Design Methods for Researching Building Layouts Through the Semantic Fingerprinting of Floor Plans. In: *International electronic scientific-educational journal: Architecture and Modern Information Technologies* 4 (13). Online: <http://www.marhi.ru/eng/AMIT/2010/4kvart10/langenhan/abstract.php>, last accessed on 23.03.2011.
- Langenhan, C.; Petzold, F.; Stocker, T. (2014). Dolphin - Grasshopper. Online verfügbar unter <http://www.grasshopper3d.com/group/dolphin>, last accessed on 04.12.2014.
- Langenhan, C.; Seifert, A.; Teichert, A.; Petzold F. (2012). ar:searchbox. Knowledge management for architecture students. In: eCAADe.
- Langenhan, C.; Weber, M.; Liwicki, M.; Petzold, F.; Dengel, A. (2013). Graph-based retrieval of building information models for supporting the early design stages. In: *Advanced Engineering Informatics*.
- Liebich, T. (1994). *Wissensbasierter Architekturentwurf. Von den Modellen des Entwurfs zu einer intelligenten Computerunterstützung*. Weimar: VDG, Verlag und Datenbank für Geisteswissenschaften. Online: <http://www.worldcat.org/oclc/56160101>.
- Roth-Berghofer, T.; Richter, M.M (2008). On explanation. In: *Künstliche Intelligenz* (2), S. 5–7.
- Vatavu, R.-D; Anthony, L.; Wobbrock, J. O. (2015): \$P Point-Cloud Recognizer. Online: <https://depts.washington.edu/aimgroup/proi/dollar/pdollar.html>. last accessed on 20.06.2016.
- Weber, M.; Langenhan, C.; Roth-Berghofer, T.; Liwicki, M.; Dengel, A.; Petzold, F. (2010): a.SCatch Semantic Structure for Architectural Floor Plan Retrieval. In: I. Bichindaritz und S. Montani (Hg.): ICCBR. Berlin: Springer (6176), S. 2010; NUMB 6176, 510-524.