Beyond the Bubble

**Computer-aided Topological Analysis and Parametric Design of Room Configurations in University Education**

Christoph Langenhan¹, Frank Petzold²
¹,² Technische Universität München
¹,² {langenhan|petzold}@tum.de

In the early stages of the design process, the conceptual idea of the envisaged building and its design parameters is still vague and incomplete. While the built environment, the end product of this design process, can be represented concretely in the form of drawings or computer models, the initial design idea can usually only be formulated abstractly, for example as schematic functional descriptions or as topological constellations of spaces. In this paper we discuss the use of reference projects to support the design process along with ways of formalising spatial configurations and their use in the design process, and examine how these can be supported using software tools. We discuss the elaboration of requirements for such software tools and their implementation as plug-in to facilitate a seamless process from analysis to evaluation in a parametric design environment. By way of example, we describe selected functionality of a plug-in developed for "Grasshopper" and "Rhino 3D" to support the design process in the early conceptual stages.

**Keywords:** Parametric design, Graph theory, Topological strategies, Interdisciplinary education

**INTRODUCTION**

Plans, models and the documentation of architectural designs and existing buildings and urban environments represent an extensive "architectural knowledge base" which contains implicit design knowledge that is valuable for the design process. The problem is that this knowledge is not available in an explicit form without first analysing and interpreting the information available. As part of the "metis" research project, funded by the German Research Foundation (DFG), innovative research methods are being developed to support design actions in the conceptual design phase. Approaches have been developed for the IT-support and linking of two key design strategies that architects use when developing ideas: functional and conceptual drawings and the use of reference material.

The complexity of modern building tasks is increasing steadily, in turn resulting in increasing specialisation among those involved in the construction process. Teams are therefore becoming ever more interdisciplinary. In addition, advancements in in-
formation technology have fundamentally changed the way in which people work in the building sector. The possibilities as well as the challenges that this presents must be considered critically. As a consequence, the objective must be to leverage the capabilities of computers for the storage of complex data and its analysis in the work process while allowing people to work creatively and intuitively.

The study of buildings that have a similar context to the design problem at hand, or that are based on a similar initial premise, is seen as way of approaching a design problem and developing a possible course of action. Rittel (Rittel and Reuter, 1992) differentiates between "tame" or well-defined problems, of the kind commonly found, for example, in the natural sciences, and so-called "wicked" problems, to which he counts design and planning problems. Reference examples of relevant existing buildings can help one to understand the problem at hand, to explore alternative approaches and generate variants, as well as to assess how well they address the problem. The use of reference projects as "a source of differentiated knowledge of the possibilities for design action" (Gänshirt, 2007) is an efficient method both in the early design stages as well as in the subsequent stages that follow.

For the analysis of architectural solutions (references) and its support using computer-aided methods, we need both a methodology for "interpreting and accessing" the knowledge they contain, and IT-based methods for formulating situations and for searching for similar reference examples, along with a corresponding intuitive user interface. In the field of architectural informatics, this topic has long been a subject of research, for example in the work of Steinmann (1997) and Mölle (2006).

To address data modelling issues, a semantic fingerprint (Langenhan and Petzold, 2010) was proposed as a means of characterising a building in much the same way as a fingerprint identifies a person (see Figure 1). This same approach can also be used as a means of formulating architectural situations and in turn for identifying semantic similarities (Langenhan et al., 2013). The aim is to develop innovative exploration and research methods for the early conceptual phase of the design process. The semantic fingerprint attempts to address the primary problem of the vague and incomplete nature of design ideas, creating a way of identifying analogous reference examples of existing buildings or building designs. By drawing on analogies from existing buildings or architectural designs, the designer can additionally verify his or her ideas, identify relevant design parameters or explore new directions and possibilities.

**BASIC PREMISES AND DESCRIPTION OF THE SOFTWARE DEVELOPMENT AND TEACHING APPROACH**

IT-based methods can both inspire and bring about new ways of teaching. Several IT-based tools, such as CAD, renderings and animations, are now widely used by architects on an everyday basis. In the vast majority of cases, though, these are used predominantly as a glorified "digital drawing board" or as a medium for presenting design proposals. In design teaching, it is therefore necessary to critically explore the fuller potential of design tools over and above these basic applications, and to explore where computers can be of assistance and where they are not the universal remedy they are often held to be.

For teaching and learning in the education of young architects, it is essential that students also actively explore how IT concepts are used in architecture. This raises a number of aspects in how architecture is taught - most notably in the design process:

- Which software methods and tools do architecture students need to be proficient in at the start of the third millennium - and why?
- How does their use differ from that of conventional working methods?
- What benefits do they offer when compared with "traditional" methods?
- How can IT-based methods be incorporated into the design process?
What changes does the media age bring for future architects?
How are new IT methods and tools taught given the rapid developments in this sector?

For teaching, this means that students need to learn not only how to operate and employ commercially available software packages but also that they are able to critically evaluate the usefulness and power of program systems, to explore and identify the new possibilities they offer and to define what they require of such systems. Rather than architects adapting to the computer or the program, the computer or program system should support the way that architects work when designing.

To address and prepare students for the challenges of the future, a common design studio teaching format for architecture was incorporated with a software development cycle of interdisciplinary projects together with the Faculty of Informatics. Simultaneously, software tools were developed and evaluated through user tests in the design process, giving the architecture students greater insight into working with computers and providing informatics students with insight into the thought processes in the domain of architectural design. We explore how we can incorporate the didactic communication of these skills in the education of architecture and informatics students and outline how it has influenced teaching and research and what the students have learnt as a result. The evaluation was undertaken by architecture students while the development of the software tools was undertaken by students of information science.
For the students of architecture, the experience of the lengthy learning process shows them that the use of software tools requires careful consideration. Furthermore, using spatial reasoning methods and employing topological analysis and design strategies offers a whole new perspective on architecture. In terms of IT skills, students of architecture come into contact with state-of-the-art technologies such as XML (Extensible Markup Language) and flow-based programming to deal with topological information.

Computer-aided approaches for the topological analysis of floor plans can be found in the work of Strug (2013) or Hillier and Hanson (1988) and especially in the teaching work of Donath [1]. The formulation of requirement graphs as a computer-based formalisation of floor plan schemes has been used for many years, not just for the purposes of analysis but also for generation, for example in the work of Steadman (1983) or Suter (2012).

SMART WORKFLOW IN A PARAMETRIC DESIGN ENVIRONMENT
According to Lawson (2000), the different design steps (see Figure 2) are defining the problem and, through a process of iterative analysis, synthesis and evaluation, creating a new solution. To digitally support the iterative process between problem and solution, Grasshopper [4] for Rhinoceros 3D [3] is used to support a combined semantic, geometric and topological workflow.

The workflow aims to offer another perspective on spatial configurations by separately dealing with geometric, semantic and topological information during the early design stages. To integrate the workflow into an architect’s work environment, the "Dolphin" add-on [2] for "Grasshopper" and "Rhino" was developed. "Dolphin" permits the querying of geometric, semantic and topological information, filtering and analysing it as well as transforming the data so that it can support other Grasshopper add-ons like "Space Syntax", "EmbryoViz", " Sandbox", "Kangaroo", " Spider Web" or "Galapagos". The information can therefore be processed further and easily adapted to different use cases and projects by students. In the following section, we outline the core functionality of the freely available add-on for "Grasshopper" and discuss overlaps and synergy effects with other tools. We begin with a description of selected functionality of the "Dolphin" add-on for "Grasshopper".

Figure 2
The Design Process
(Lawson, 2000, p. 47)

DOLPHIN - AN INTEGRATED INFORMATION-BASED DESIGN PLUG-IN
The goal of the "Dolphin" add-on for "Grasshopper" is to support the early stages of the design process using information technology in order to find pre-existing architectural solutions that can serve as a source of inspiration, an explicit solution, or a means of acquiring a better understanding the specific design problem. "Dolphin" makes it possible to use digital semantic building information in the parametric work environment of "Grasshopper" and "Rhinoceros" using the XML-based AGraphML exchange format developed by the "KSD research group". The AGraphML format stores spatial and topological information that can be accessed through local sources and web-based sources. The information can either be downloaded in AgraphML files or accessed directly using a building information cloud that handles spatial informa-
tion in a graph database, geometric information on a BIM Server and meta-information in a content management system.

To encode spatial information in AgraphML, automatic and semi-automatic methods have been developed and implemented by the "KSD research group". To formalize an architectural situation in digital form, we propose creating a semantic fingerprint of the respective building that can identify a building in much the same way as actual fingerprints identify individuals. In order to map the spatial structure of architectural situations, graph-based approaches are being investigated to mathematically represent fingerprints for information such as spaces and their connections, alignment, orientation or urban integration. The nodes of the graph are labelled using a fixed taxonomy [ROOM, KITCHEN, LIVING, SLEEPING, WORKING, CORRIDOR, TOILET, BATH, EXTERIOR, STORAGE, BUILDINGSERVICES, CHILDREN, PARKING]. The labels of the edges [DOOR, ENTRANCE, PASSAGE, SLAB, STAIRS, WALL, WINDOW] represent the relation between spaces. In order to use the information in "Grasshopper", the "Dolphin" components are distinguished in the categories "Analyzer", "BIM Tools", "Filter", "Local Source", "Statistics" and "Web Source".

Three "Analyzer" components have been implemented to analyse building information and visualise the information in "Rhino". Four "BIM Tools" components facilitate interaction with BIM models to explore the references. Two "Filter" components were developed to make it possible to define basic requirements and to automatically check which conditions are fulfilled by which part of the spatial configuration. Two types of conditions are distinguished: 'simple conditions' that have direct access to the AgraphML data and 'complex conditions' that have to derive the values from the AgraphML data in order to ascertain if a condition is fulfilled. The "AgraphML Reader" loads several AgraphML files from one folder and the "File Checker" component tests whether the single file has all required AgraphML attributes. The output is either a list of AgraphMLs or a single AgraphML. In addition to local sources, four "Web Source" components integrate date from a building information cloud in "Grasshopper" and "Rhino". Three "Statistics" components compute AgraphML data, such as room type, togetherness, arrangement and sequence as well as size, number and likelihood of connections to other rooms.

Figure 3 shows an example of some "Dolphin" components of the categories "Local Source" and "Analyzer" as well as standard "Grasshopper" components that make it possible to visualise and process the information further. In panel (1), the user chooses the path of the file. In panel (2), the "Checker" component then ensures that the file contains the data necessary for the subsequent components (3) to (5). The remaining "Grasshopper" logic (6) is used exclusively for visualising using standard components.

Figure 3
User interface of "Grasshopper" showing the components of the "Dolphin" plug-in

To generate variations of the spatial configuration, the data can be processed using the "Kangaroo" or "Galapagos" plug-ins. Alternatively the "Spider Web" or "Space Syntax" plug-ins can be used to analyse the data. The table in Figure 4, left, shows the degree of integration, connection, accessibility and the possible traversal of the configuration from left to right. In "Grasshopper", the user can, for example, denote a room as the starting point for an assessment of the topological depth and have "Rhino" output a corresponding hierarchical spatial diagram.

The results from the analysis and variant generation can be exported or be used for further processing using the "Hummingbird" add-on for "Grasshop-
DISCUSSION
The evaluation of "Dolphin" was undertaken by architecture students while the development of the plug-in was undertaken by students of information science. For the students of information science, the application-oriented software development project improved their understanding of the context in which the application is used, and provided insight into the domain of architecture. This knowledge can be transferred to other domains and application areas in the future. Overall, the students learnt to think in a structured and strategic manner and to work in a focussed way towards their objective. In addition to their work on the design and software projects, the students acquired a better understanding of architectural relationships and spatial concepts, which will benefit them in the long term.

In fact, several of the architectural students have become extraordinarily adept at recognising and reading spatial situations and in transferring their "readings" of spatial ideas to other projects or in recognising them in other contexts. Furthermore it shows that IT tools offer much more potential than they are presently used for: as analytical tools they complement their traditional application as presentation tools and also offer new and pioneering ways of communicating architecture - in teaching and probably also in the public sphere.

CONCLUSION
Rapid technological advances have brought about increased possibilities for supporting architectural design processes using information technology. This includes the obvious use case of generating and presenting variants, as well as the analysis of dependencies. This makes it possible to present a wide variety of relationships that would not otherwise be available to the viewer, which can be used to communicate architecture, not just in an academic environment or to professionals, but also to the public at large to broaden the general architectural discourse.

The course entitled "Parametric architecture - beyond the bubble" communicates knowledge and skills in the use of digital tools in the design process from the analysis of existing buildings and architectural designs, and the derivation of design parameters to model creation and the creation of new design variants. The course confirmed the potential that IT methods hold for the field of architecture. It also shows that tools are at present still very complex and that IT beginners are often distracted...
and overwhelmed by the numerous possibilities such programs offer. Here we need to find ways of simplifying the programs for other applications, always taking into account the potential users - in this case architects - and their specific needs. All in all, involving students of architecture and information science in ongoing research activities has been beneficial for students, researchers and the scientific community.

REFERENCES


Suter, G, Petrushevski, F and Sipetic, M 2012 'Boolean operations on network-based space layouts', 9th Work-