

Understanding Design Through Design Support Tools

Bauke de Vries, Henri Achten, Jos van Leeuwen

Design Systems

Faculty of Architecture, Building and Planning

Eindhoven University of Technology

B.d.Vries@tue.nl

1. Introduction

The Design Systems (DS) group of the Faculty of Architecture, Building and Planning of Eindhoven University of Technology deals with Computer Aided Architectural Design (CAAD) support. We can note that even after many years of development, mainstream CAAD software remains notoriously bad in design support. This concerns aspects such as concept generation, maintaining divergent thought processes, offering fast and numerous sketch-like representations, propagating design concepts to final design, and so forth. Good support in this phase can help the designer identify critical issues beforehand, understand implications more deeply than by manual methods, investigate more alternatives before settling on a design move, and so forth.

Obviously, we are not alone in trying to improve design support in the early phase. There is much academic research in this area. Very often such work is explorative and technology-driven through prototype systems that demonstrate how various techniques can be applied. Examples of such work are found in great numbers and in a great variety of applications – to name just a few: Esquisse (Leclercq 2001), Electronic Cocktail Napkin (Gross 1996), Phidias (McCall 1999), and Sketchbox (Stellingwerff 2005). This kind of work is important to show the potential of CAAD for architects. Indeed, one cannot even do without this exploratory work if genuinely novel systems should be created – much of our research reported in the previous DRN symposium falls in this category (Achten, de Vries and van Leeuwen 2001). The creation of a new system however, answers only *how* we may improve design support, but not *whether* a system actually supports the design process, and if so, to which degree. Questions that we need to answer are “how do we define support,” “how do we measure degree of support,” and “how do we infer the influence of the support tool on the design itself?”

The research in our group therefore, has evolved from explorative to the development of innovative systems and subsequently testing these systems on their performance as design support tools. In this paper we will focus on two main areas in which we look at design support and the measurement of improvement: novel user interfaces and Bayesian networks. This will take place after an overview of Design Systems and a brief sketch of the context. Following this, we present a recently started applied-research collaboration called Janus. In the conclusion, we summarise our findings from the past years and sketch our view for the near future.

2. Development of the Design Systems group

At the last DRN symposium in 2000, the Design Systems group had been recently established and at that time did not have a professor. A dedicated research programme termed VR-DIS was started in which the expertise of the group, in particular Virtual Reality (VR) and design knowledge modelling, was applied in research projects ideally facilitating all disciplines in the

Faculty of Architecture of Eindhoven. In a larger framework, the research of Design Systems was part of the Design & Decision Support Systems (DDSS) research programme. In this programme we participate with the Urban Planning group (UP) headed by professor Harry Timmermans. Design Systems and Urban Planning share strong methodological viewpoints, but applied on different levels (architecture and urban planning), and with different perspectives (design support through VR-DIS and planning support through choice modelling and prediction).

In terms of staff, the Design Systems group has seen a number of changes. Most importantly, in 2001 Bauke de Vries was appointed professor of Design Systems. In 2000 Henri Achten became associate professor (UD) in Design Theory and CAAD; in 2002 Jos van Leeuwen became assistant professor (UHD) in Collaborative Design, and Aant van der Zee became associate professor in Evolutionary Design. With the retirement of John Carp in 2004, much of the first-hand knowledge of the Stichting Architecten Research (SAR) disappeared from the group. We are maintaining however, a course in this area on MSc-level. In the period 2001-2002, Sverker Fridqvist worked on a post-doc project concerning Feature-Type modelling, recognition, and management.

Table 1: Finished, running, and related PhD-projects in the period 2001-2005.

PhD Candidate	Title of thesis	Collaboration
Finished PhD-projects at Design Systems		
Amy Tan	The Reliability and Validity of Interactive Virtual Reality Computer Experiments	DS & UP (-2003)
Nicole Segers	Computational Representations of Words and Associations in Architectural Design	DS & ID (-2004)
Maciej Orzechowski	Measuring Housing Preferences using Virtual Reality and Bayesian Belief Networks	DS & UP (-2004)
Running PhD-projects at Design Systems		
Jan Dijkstra	Simulation of Pedestrian Flow in Urban Environments	DS & UP
Aant van der Zee	Computer-Aided Evolutionary Architectural Design	DS
Vincent Tabak	User Simulation of Space Utilisation	DS & UP (2003-)
Jakob Beetz	Multi-Agent Systems for Collaborative Design	DS (2003-)
Nischal Deshpande	Co-located, Multi-Disciplinary, Collaborative Design Space	DS & CBS TNO-TU/e (2003-)
Related PhD-projects with other groups		
Dima Aliakseyeu	A Computer Support Tool for the Early Stages of Architectural Design	DS & ID (-2003)
Slava Pranovich	Structural Sketcher: A Tool for Supporting Architects in Early Design	DS & CE-V (-2004)
Maxim Ivashkov	ACCEL: A Tool for Supporting Concept Generation in the Early Design Phase.	DS & CBS TNO-TU/e (-2004)
Shauna Mallory-Hill	Supporting Strategic Design of Workplace Environments with Case-Based Reasoning	DS & CBS TNO-TU/e (-2004)
Ilse Oosterlaken	Scenario Methods Within a Development Planning Approach Towards Complex Urban Sites (Re-) Development	DS & CME (2005-)

During 2001-2005, three PhD-theses were completed within the Design Systems group (see Table 1). Currently, we have five running PhD-projects. In collaboration with other chairs and faculties of Eindhoven University of Technology, we were also co-promotor or advisor in four other PhD-projects. The abbreviations in the Table are used for the following institutes with whom we collaborated:

- *CBS TNO-TU/e*: Centre for Buildings & Systems TNO-TU/e;
- *CME*: Construction Management & Engineering (Faculty of Architecture, Building and Planning, TU/e);
- *ID-UCE*: User-Centred Engineering (Faculty of Industrial Design, TU/e);
- *CE-V*: Computational Engineering, Visualisation (Faculty of Mathematics & Computer Science, TU/e).

In the past five years, apart from the *DRN Symposium* (2000, 2005), Design Systems has co-organised the bi-annual *Design & Decision Support Systems in Architecture and Urban Planning* conference (2000, 2002, and 2004), and the *CAAD Futures 2001 conference*. We are currently in the process of organising the 2nd *International Conference on Design Computing & Cognition* which will take place in Eindhoven in 2006.

In our educational programme, design research features on the interaction between design theory, CAAD, and design methods (Achten 2003; van Leeuwen, van Gassel and den Otter 2004), and the integration of MSc.-graduation projects with our research work. In the past five years, this has led to 10 MSc.-degrees with a specialisation in Building Informatics and a number of publications (de Vries, Verhagen and Jessurun 2004; Willems 2004; de Vries, Tabak and Achten 2005; de Vries and Harink 2005; van Leeuwen and Jansen 2005). Currently, we have five MSc.-students in the area of Building Informatics.

3. Design support: user interfaces

Many different systems have been developed and are used for computer-aided architectural design during the early stages. These tools present a large number of useful ideas, many of which have influenced our research. In this section we give an overview of some design tools that are currently available or still under construction. We cannot be exhaustive (see de Vries et al. 2003 for a lengthy overview), but we tried to select a number of solutions that are typical for a certain class. These tools intend:

- to mimic the traditional architectural environment;
- to simplify interaction with the system (input, manipulation, and presenting design information);
- to support various design aspects (e.g., visual, spatial, structural);
- to bridge the gap between early stages of design and following stages.

The kernel of a design system consists of the design model and processing unit. They provide design elements and operations on them. We propose a classification of design systems on the basis of two criteria:

1. *Specificity to the architectural domain*: this criterion distinguishes generic systems from systems that are specifically oriented towards architectural tasks. The connection to architecture is defined on the basis of design elements that are provided in the system.
2. *Versatility of provided support*: this criterion is related to the number of different techniques that the system offers for the support of designing and for structuring relationships between design elements.

Figure 1 classifies a number of drawing systems that can be used for architectural design.

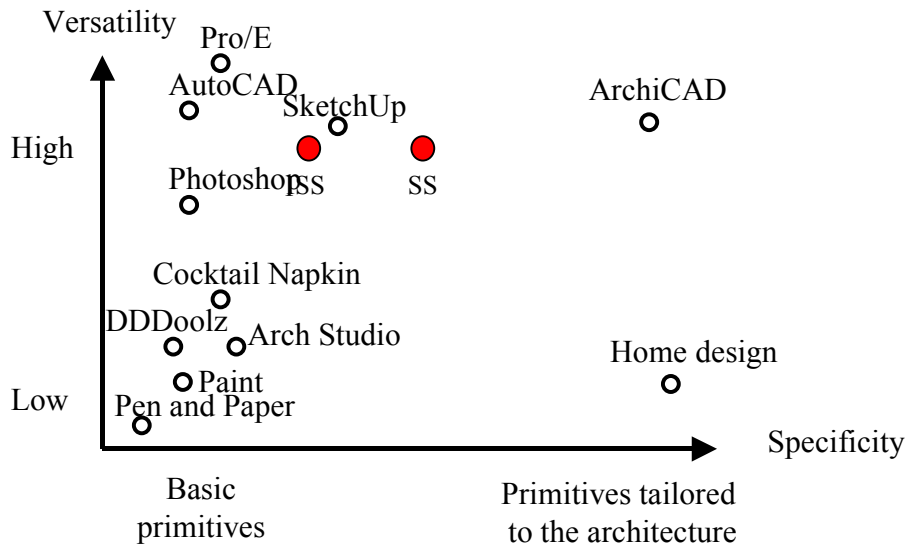


Figure 1: The classification of drawing systems.

The horizontal axis shows specificity: some systems are based on generic primitives such as circles, rectangles, etc.; other systems are based on primitives that are tailored to the architectural domain, such as walls, doors, windows, etc. The vertical axis in this graph displays the versatility of techniques that are offered for managing design elements. Some systems provide only basic manipulations, while others offer an extensive set of operations, options to define macros, constraints, etc.

Below we present two design support systems the intend to fill the gap in Figure 1, namely the Idea Space System (ISS) and Structural Sketcher (SS).

Idea Space System (ISS): digital graphics and annotations

Architectural design proceeds very much through the production of sketches. An often neglected aspect concerns the annotations that architects make while sketching and designing. What can happen to the design process if a design system could capture the annotations and offer verbal associations to the designer while he or she is designing? Would such help improve the flow of divergent ideas and lower the risk for fixation?

This question is addressed in the PhD-work by Nicole Segers on the Idea Space System (Segers 2004). The interface is built in the Desk-Cave (Achten, Jessurun and de Vries 2004), integrating the Visual Interaction Platform (VIP3), developed earlier by Dima Aliakseyeu (2003). VIP consists of a table top on which a desktop is projected, and which is also recorded with a camera. The camera tracks physical objects (movement and rotation) and uses this information to manipulate the underlying projected desktop. In this way, a physical interface is connected to virtual objects. The Idea Space System enhances this platform by adding an additional vertical screen, and using as input devices pens and a tablet. On the desktop, projected paper is used, which can be displayed in various transparency settings. Therefore, the architect works as if using a big sheet of paper. The ISS system has handwriting recognition, so there is no need for the distraction caused by a specialized input device for text.

To work with the annotations, we have developed a specialised component called the WordGraph component. It takes the annotated words produced by the architect, and searches for interesting relations among these words. These come scrolling by in a graph representation on the vertical screen as they are generated in real-time. New words found by the system are displayed in yellow boxes. Because the feedback is not displayed in the horizontal working field, the architect can ignore it and is not interrupted in the design process. When a graph is selected, it is inserted as an image on the desktop and becomes part of the information displayed on the horizontal working field.

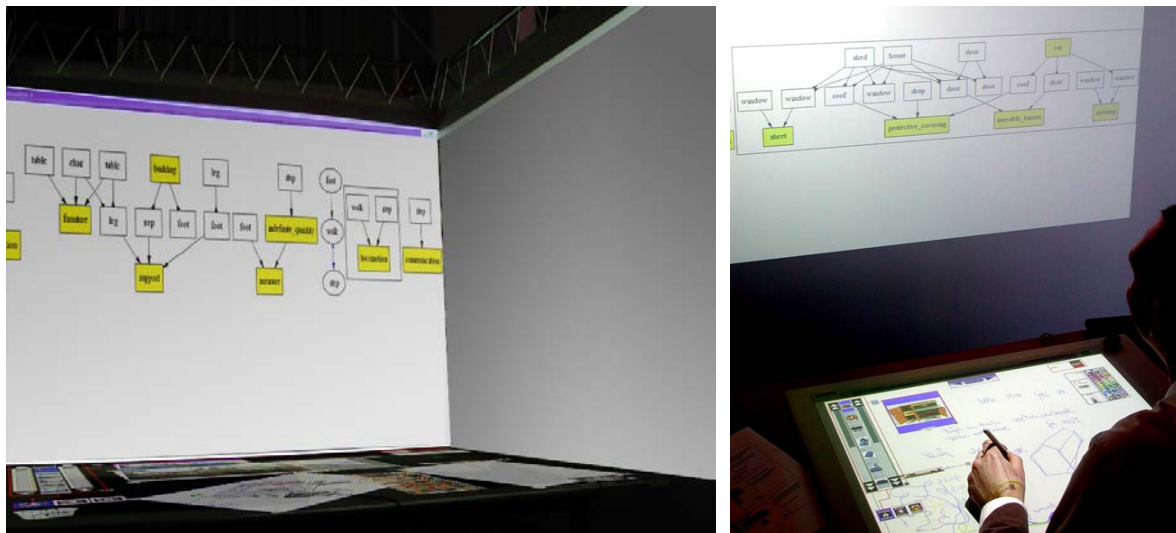


Figure 2. Left: Projection of found new words displayed up front. Right: Person working with ISS.

The effectiveness of the system was measured with practising architects who were given two different design tasks, one to make with the system with word associations, and one task to make with the system without word associations. We measured several aspects: periods of activity and inactivity and their correspondence with acceptance of word graphs; judgement of a panel on the quality of the design; the word graphs displayed and words written down on the page, and so forth (see Segers 2004 for more details). We found that the system did not influence all architects equally; only those who indicated they verbally engage design tasks seemed to profit from the system – for the others we could not determine a significant difference.

Structural Sketcher: designing with graphic units and relations between them

Although pen and paper makes an almost unbeatable combination in terms of ease of use and speed of production, improvements are not too difficult to think up: what if the sketched elements remain persistent and do not have to be drawn time and again; what if implicit relationships in a sketch are maintained when the architect only wants to rearrange some elements? For this purpose, Slava Pranovich (2004) developed Structural Sketcher, which utilises a number of architectural graphic primitives – graphic units, see Achten (2004) – as a basis for a more structural approach to sketching.

In order to make Structural Sketcher simple and natural for the architects we developed new interaction techniques. These techniques are based on architectural metaphors of the early design process that are easy and intuitive for the architect. We define interaction techniques on top of the geometry engine that provide a possibility to explicitly/implicitly control interactions

between objects from different layers. For example, the architect can define relations by changing the rank and the layout of graphic units, and he can use layers to structure the relations between graphic units. With straightforward metaphors (*pins* and *clips*) he can determine more advanced manipulation relationships. The *pin* is used to connect a pair of graphic units and to block propagation of particular transformations. It is visualized as a nail-head pyramid, where each nail-head has its own colour and marks the blocking of a particular transformation (see Figure 3 Left). The user can modify the transmission properties of a pin using a special properties manipulator called *Kite* (see Figure 3 Middle).

The *Kite* combines several operations (skew, scale, rotate, and move) in a single metaphor. A mouse click on a corresponding zone of this manipulator switches on/off the transmission of related transformation (the centre of the manipulator for rotation, the corner for scaling, and the bars for skewing). If the transmission of a transformation is blocked then a corresponding zone in a manipulator is highlighted with red. The *clip* is provided for connecting objects, and is visualized as two balls that are attached to graphic units and are connected by a line (Figure 3 c).

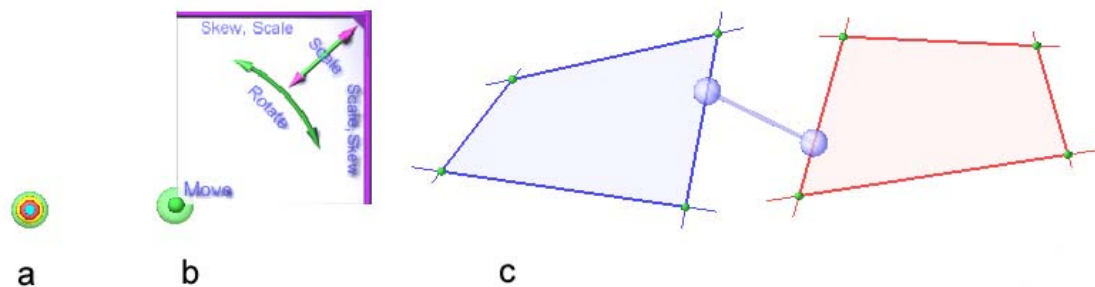


Figure 3: (a) The pin, (b) the Kite manipulator projected on a pin, (c) the clip.

The visualization of transmission properties and the manipulation of these properties is similar to the pin. The interaction mode is defined automatically: if the user manipulates graphic units, then natural mode is implied; if the user manipulates layers, then layers mode is implied; if the Ctrl button is pressed then manual mode is implied. Figure 4 shows a number of elements created in Structural Sketcher.

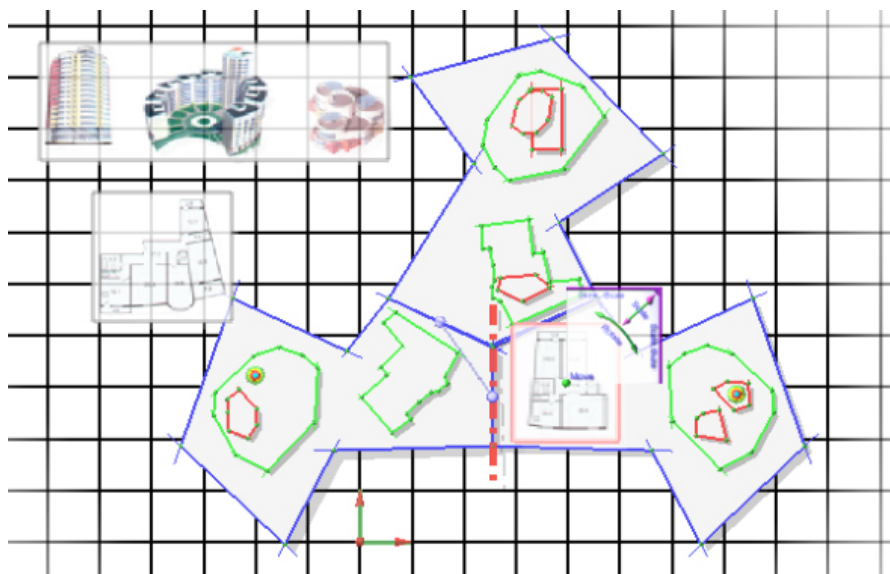


Figure 4: Example created in Structural Sketcher.

The system was tested in a qualitative manner by asking a number of architecture students to work through a number of prepared exercises, ending by making a small design for a doctor's practice, and asking them to rank-order Structural Sketcher relative to some major design software on several aspects. The system was tested in a quantitative manner by measuring over the prepared exercises the minimum number of required user-interactions compared to some major design software. The qualitative tests clearly placed Structural Sketcher between pen & paper (which usually ranked first) and the other software. The quantitative tests showed a reduction of user actions of 60%, 45%, and 30% compared to other software. We infer from this that the user-attention to interface issues is greatly reduced and this allows at least in principle more focus on design work.

4. Measuring the effectiveness of design support: Bayesian networks

Understanding preferences of future house owners is important to design close to the desires of customers – even when they are unknown as in the case of many project developments. Current inquiry methods employ stated-choice or conjoint measurements that have two distinct disadvantages: they are text-based, and they involve judgement of many (often unlikely) profiles before reliable inference is possible. While Virtual Reality is commonly assumed to provide an experience much closer to lay-people's understanding, it remains a question whether the employment of this technique actually brings about more accurate measurement of preferences. Maciej Orzechowski (2004) developed a VR-based system – MuseV3 – in which future home owners adapt a design to their own preferences. For measurements we used the technique of Bayesian networks.

Bayesian methods provide a formalism for performing reasoning using partial beliefs under conditions of uncertainty. Propositions are quantified with numerical parameters indicating the strengths of beliefs, based on some body of knowledge. These parameters are combined and manipulated using the rules of probability theory. The Bayesian view of probability provides a natural way to encode expert knowledge in domains where little or no direct empirical data is available. An attractive feature of the approach is that when data becomes available Bayesian reasoning gives a consistent method for combining data and judgment to update beliefs and enhance knowledge. In this way a Bayesian network captures believed relations (which may be uncertain, stochastic, or imprecise) between variables that are relevant to some problem - in our case user preferences for a set of alternative housing designs or design attributes.

In order to acquire measurements that can be compared, we used a concrete project of a project developer, and asked potential house-buyers who were interested in that project to adapt a base-line design to their own preferences – much in the way as the project developer interviews their customers. The people could adapt their house according to the same possibilities as the real project, using three-dimensional projection and navigation (the VR-part) and a plan view (Figure 5). In both views, the house could be adapted. The changes were recorded by the system, and the values of Bayesian network adapted to the new evidence (the state of the house). The network was finalised for each project after the users indicated they were satisfied with their design.

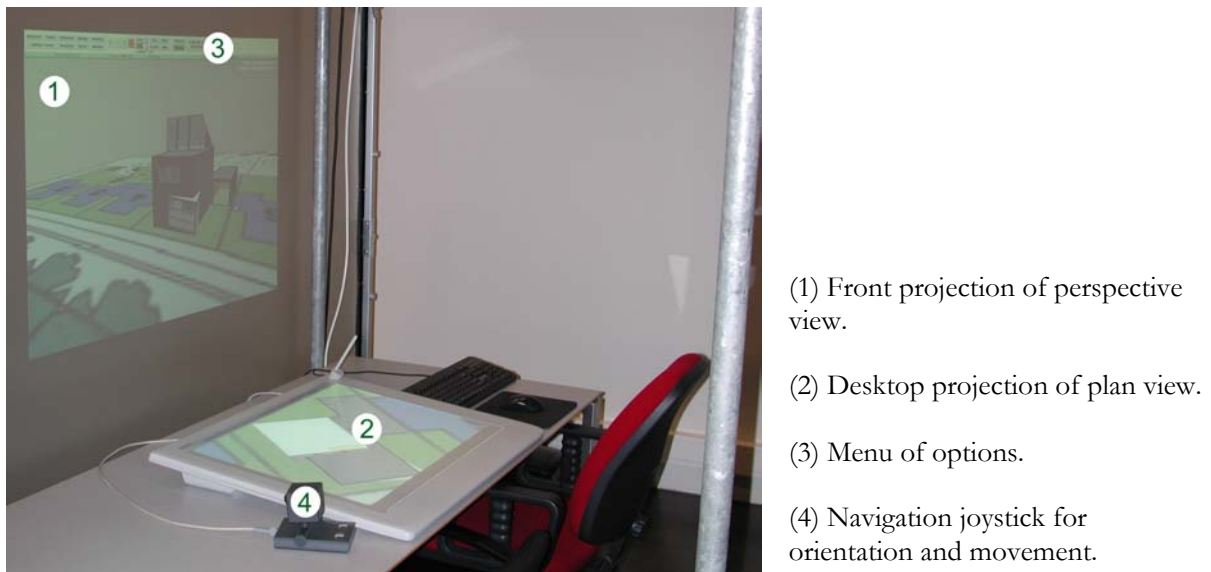


Figure 5: The MuseV3 setup.

Figure 6 shows the structure of the Bayesian network. We can distinguish two levels. In the first, top most level, there are the variables expressing a subject's preferences for each design attribute. The second level contains variables representing probabilities of choosing a design attribute. At this level, the actual choices captured from the virtual environment are entered into the network. Consequently, each attribute creates a separate vertical branch. The price is treated as overall cost, represented by variable γ . The actual cost of each design option is encoded in its corresponding conditional probability table attached to the corresponding attribute node. Prices for options remain constant for all subjects.

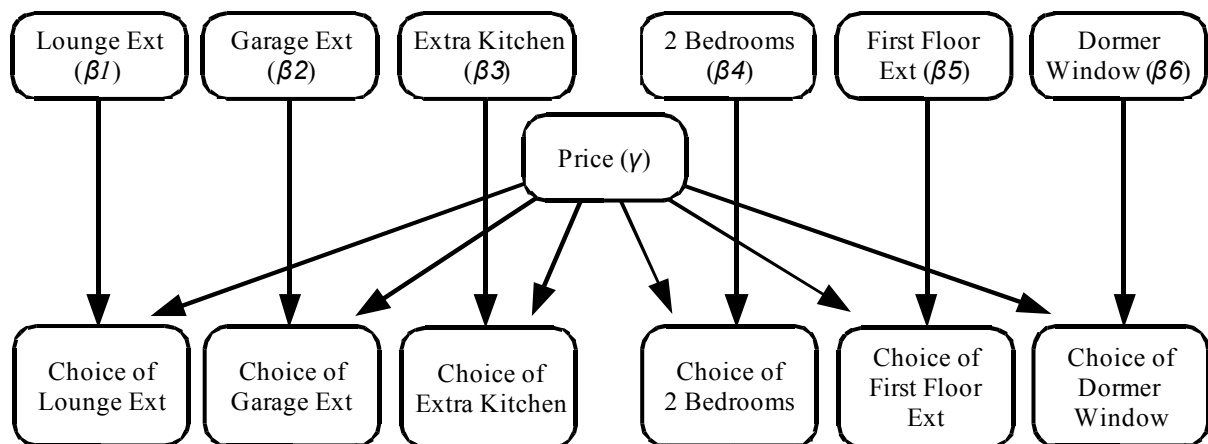


Figure 6: Bayesian Network for the experiment case.

When the experiment starts, the initial conditional probability tables for the parameter nodes has uniform distributions – meaning that every outcome is equally likely to occur (no preference is inferred). After each session, the network is updated, and the distribution changed accordingly. As can be imagined, the accuracy of the network increases when the number of cases increases.

We measured the effectiveness of the system in two ways: by examining the internal consistency of the Bayesian network, and by offering advice to the users after a session based on the differences between their final design, and the expectations encoded in the network. Considering the first, we found that with the 64 respondents in the experiment, the network was still subject to changes in values when a greatly divergent design was offered. Numeric simulation showed that the network becomes robust with 100 respondents. Considering the second aspect, we could make two observations: (a) the number of suggestions that were accepted by the respondents increased during the course of the experiment; and (b) the number of differences between the network and the realised designs decreased during the course of the experiment. This indicates that the system is indeed learning preferences, and that these increasingly matched the preferences of the respondents.

5. Janus: Joint Architectural Network for Urban Synergy

The Joint Architectural Network for Urban Synergy, or briefly put: *JANUS*, is a recent initiative launched in the beginning of 2005. This network was initiated in collaboration with a Dutch CAD software developer, with the objective to combine the best of the two worlds of scientific research and daily practice in design offices (www.urban-synergy.org/). It is established in the form of a foundation that acts as initiator and co-ordinator of research projects targeting issues of communication in the construction industry. The focus in these projects is mainly on *what* is (or should be) communicated between partners in construction, rather than on *how* they are doing this. The rationale behind this is that in most innovation projects, the development is driven by the technology itself, rather than based on a thorough analysis of requirements. In our projects we aim to bring professionals together and ask them what they need to know from each other to be able to collaborate, and why they need this information.

To perform this research, the foundation develops a national network of professionals in all disciplines involved in urban development and construction projects. The network currently includes architects, contractors, principals, and representatives from both national and municipal authorities.

Some of the topics that are being developed in these projects are the following:

- Digital code checking.
- Online building permit requests.
- Bill of Experiences (compared to Bill of Requirements).
- Architect-Contractor communication in early design and in procurement procedures.
- The role of art and history in urban development.

One of the first projects that will deliver tangible results is the *Digital Dormer* project (“*Digitale Dakkapel*,” see van Leeuwen, Jessurun and de Wit 2004). The Digital Dormer is a website where house-owners who want to build a dormer on the roof of their house can make a design of the dormer that will be automatically checked against the national building codes, the municipal aesthetics rules, and a number of technical requirements. Once the design is satisfactory, the user can use the system to generate and submit all information that is necessary to request a building permit for the dormer. The municipality can easily check if this information is complete and can follow up with a much enhanced procedure of granting (or rejecting) the request for permission. Nationally, dormers account for 10-15% of all smaller construction projects (< € 100.000) in the Netherlands. Apart from the direct benefit of this application, the knowledge that was gathered in the project will also be used in following projects that involve digital code checking and communication procedures between civilians and authorities regarding the built environment.

6. How to understand designers through tools?

We have learned that study of new techniques isolated from the design activity is not very informative about the impact of such techniques on the design process. The integrated study in a design context however, greatly complicates research methodology as many additional factors are introduced. It also places our work in a sometimes uncomfortable duality between systems developers – who usually consider the work finished after a prototype has been built – and psychologists – who care not much for the design context. As our experience grows with the research projects, we feel more confident about the methodological approach, although we still have a lot to learn.

Very often we find that our initial expectations what a technique or system “will do” are unsophisticated or simplified. In many cases complicating circumstances occur, and designers prove more varied than anticipated from the literature or our intuitions. This points to the limited value of broad, sweeping theories of design(ers). The conclusions that we draw from our work have to be understood within the sharp limits of the system, the experiment, and design task. Outside these limits generalisations are very likely inaccurate. We increasingly understand the complexities of computer aided design support, and the varied relationships that designers can have with such tools.

Research through tools offers two kinds of insights into designers. The first insight is a ‘productive’ insight, in terms of the problems designers encounter, what kind of tools they employ, and how they would like to employ these tools. We find that designers are often incapable at describing their needs in terms of design support, or to assess without hands-on experience how a given technology might work for them. Prototypes prove invaluable in this respect, and measured experiments are necessary to determine the actual use. The promise of new technologies is often exactly what it is – just a promise, without any specification what exactly will improve (and, equally important, what exactly will remain unchanged or even degrade). Experiments in design settings are modest in explanatory scope, but form an effective antidote to inflated expectations of new technologies.

The second insight is a more indirect ‘cognitive’ insight, in terms of reasoning mechanisms that designers employ. A particular tool or technique assumes certain behaviour or structures that will be supported, and thus that a particular task dependent on these structures or behaviour will be improved – in the examples discussed here, associations and creativity (ISS), graphic structures (Structural Sketcher), and spatial reasoning (MuseV3). We are not doing cognitive research per se, but we do find that our work yields insight in design cognition in particular through revision or refining of our grounding assumptions. In this sense we learn a lot about the complexities involved in design support.

7. Acknowledgements

The work in the Design Systems group is very much the result of teamwork, and a shared interest between all members of the group in each other’s work, both in the current staff and the people who have been with us in past five years. Working together with other groups in the Faculty and other Faculties of the university has proven very rewarding and stimulating. We gratefully acknowledge their contributions and collaboration.