A comparison of computer-aided tools for architectural design

N. M. Segers\textsuperscript{a}, H. H. Achten\textsuperscript{a}, H. J. P. Timmermans\textsuperscript{b}, B. de Vries\textsuperscript{a}

Technical University of Eindhoven
\textsuperscript{a}Design Systems Group
\textsuperscript{b}Urban Planning Group
Eindhoven
The Netherlands

ABSTRACT

Several computer-aided tools could be used to support a broad area of architectural design. The intention of this paper is to give an overview of possible tools that support the analysis, synthesis, or evaluation processes underlying architectural design. We evaluate these tools and elicit their requirements as tools for computer-aided architectural design. Potential improvements are a broadening of the solution-space of the architect (stimulate inspiration), an increasing speed or ease of generating and evaluating design alternatives, a better accessibility of required information, and verification of the rules and demands of the brief. In developing a tool for the very early design stage, an overview of existing tools, including their potential advantages and drawbacks, that could somehow support these ideas is a necessary stepping-stone.

1 INTRODUCTION

The very early phase of architectural design has many faces: it typically is a chaotic and complex process during which every architect appears to use his or her idiosyncratic methods. Regardless of such individual differences, design processes seem to have some common elements, or at the very least the literature on design methodology has identified some common elements. Bryan Lawson’s work (1997) can serve as an example. He describes the architectural design process as portrayed in Figure 1. He distinguishes three stages in the design process: analysis, synthesis and evaluation. Usually, the brief is the starting-point. In the analysis phase, the designer must explore, assimilate, order and structure various kinds of information, including the requirements and rules, but also non-relevant information that might inspire him. The analysis should result in a formulation of design objectives. Only some of the information that was assimilated is used for problem-solving activities.

Designers develop their ideas through many different variants working on all scale-levels. The synthesis thus involves activities such as brainstorming, modeling, sketching, and thinking. It typically results in variants and schemata that need to be evaluated against some explicit or implicit criteria. Some ideas and solutions are kept
in mind, but not further developed and some are critically examined, checked and verified with the brief, the objectives, requirements and rules.

This characterization of the architectural design process is typical of many similar conceptualizations. Similar schemes can be found in many textbooks and industry documents. Figure 1 may suggest that architectural design processes typically follow such a rigid, sequential structure, which may be at variance with design practice. Although certain aspects of these three phases or facets can always be discerned, actual processes have many feedback mechanisms and design processes may evolve more simultaneously than sequentially. Important for the discussion to follow is not so much whether this conceptual framework provides a valid representation, either normatively or empirically, of architectural design processes, but rather that it is a useful vehicle to evaluate and position the design tools that we will discuss in the remainder of this paper.

What this conceptual framework however fails to articulate is the aspect of communication. Design decisions need to be communicated to the clients, the technical experts, legislators, users and other designers. Participants need to be able to express their findings about the design and discuss strengths and weaknesses. Therefore, we added the element of communication to Figure 1.

Traditional communication tools include paper, pencil and small-scale models. They are used to get a better grip on the design problem, and serve to explore ideas and thoughts and to exchange these with other members of the design team. Although these tools have proven and continue to prove their usefulness in many architectural design processes, they are limited, as any tool, by their specific characteristics. Thus, there is a constant search for new tools that may have certain new characteristics that may support particular aspects or components of architectural design processes.

The aim of the research project reported in this paper is to contribute to this quest for new, and possibly improved or complementary, design tools. In particular, we plan to develop a computer-aided tool for architects, which can be used in the early stage of the architectural design-process to stimulate, generate and communicate new ideas. The challenge is develop a natural, interactive, "solution-and-question-generating" design tool. In formulating the requirements of such a tool, we first
explored existing tools. The present paper gives an overview of the kind of tools that might be relevant. In addition, these tools are systematically compared in terms of their potential usefulness for the various phases or aspects of the architectural design process.

The paper is organized as follows. First, we will motivate the underpinnings of this explorative study. Next, we will systematically describe and evaluate the selected design tools. Finally, we will draw some conclusions and discuss future avenues of research.

2 A COMPARISON OF TOOLS FOR THE DESIGN PROCESS

2.1 Working method

Several tools from a variety from disciplines, many outside of architecture, are compared to illustrate the differences in the tasks that the architectural designer performs and that are supported by the various tools. We will position the tools using Lawson’ general scheme, representing the architectural design-process. This scheme is appropriate because of its level of abstraction and comprehensiveness. We will concentrate on computer-aided tools for the analysis- and synthesis-processes.

An examination of the relevant literature resulted in ten computer-aided tools that we will compare in this paper. These tools in particular were chosen because they were found to be intuitive to work with, innovative, or a typical example of other, similar, computer-aided tools. We do not claim this list to be complete. The criteria or requirements are partially elicited from a brainstorm-session we had with architects.

Table 1: Overview of Comparison

<table>
<thead>
<tr>
<th></th>
<th>Organic Art</th>
<th>Treetown</th>
<th>Flatland</th>
<th>CreaPro</th>
<th>Cocktail Napkin</th>
</tr>
</thead>
<tbody>
<tr>
<td>analysis</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td></td>
</tr>
<tr>
<td>synthesis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>++</td>
</tr>
<tr>
<td>evaluation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>++</td>
</tr>
<tr>
<td>Teddy</td>
<td></td>
<td>Holosketch</td>
<td>Sculptor</td>
<td>DDDoolz</td>
<td>VIP</td>
</tr>
</tbody>
</table>

2.2 Analysis of Design Tools

2.1.1 Organic Art

Computer Artworks Ltd. has developed a screensaver, based on the art of William Latham, which is called Organic Art (1992). An interesting feature of this screensaver is the user can manipulate what is seen on the screen. There are many combinations of
shapes, mutations (ways of interacting), movements, textures, colors and backgrounds to explore. Organic Art generates these combinations in real time.

Although the tool is not specifically developed with architects in mind, it still is interesting to examine its usefulness for supporting the architectural design process since the tool deals with shapes.

In terms of analysis, the tool demonstrates how a solution space can be explored. The application in the architectural domain, however, is not possible due to the lack of architectural elements and shapes. The mutations or movements also do not support architectural relations. As for synthesis, it is unlikely that the tool will be very relevant without architectural elements and without the user being able to control the interaction of the elements. Moreover, it is not possible to specify the architectural problem; there is only a background image. Finally, the tool has nothing to offer in terms of evaluation and communication. We concluded that this tool has only very limited relevance in the context of architectural design.

2.1.2 Polyshop / Treetown
Treetown, developed by Peter Oppenheimer, is a program that makes three-dimensional models of trees using a fractal branching metaphor. It is implemented in PolyShop, a program developed at the University of Central Florida. It is possible to model architectural spaces and urban structures that resemble existing historic cities and structures. Form geometry and topology can be specified by algebraic equations, production rules, form abstraction and direct model modifications. Thus, in term of analysis, this tool demonstrates and supports form principles of urban structures, and may thus be interesting for an architect. The tool has less to offer in terms of synthesis as it merely is generating shapes, and offers only limited possibilities of placing things in their context. Similar to Organic Art, Treetown has nothing to offer in terms of evaluation and communication.

Figure 2: Screenshot of Organic Art. (from the internet, 1992)

2.1.3 Flatland
The third tool that we discuss is Flatland (Mynatt et al., 1999), published by E. Mynatt, T. Igarashi, W. Edwards and A. LaMarca at the College of Computing (Georgia Institute of Technology), the University of Tokyo, and Xerox Palo Alto Research Center.
Figure 3: **Model of Treetown.** (from the internet)

Figure 4: **Impression of Flatland** (Mynatt et al., 1999)

It is an augmented whiteboard interface designed for informal office work. The user can edit it with a pen-like device, as if it was a normal whiteboard, but added are some functionalities, such as scaling what has been written or drawn, drawing a map (automated double line), color palette, take snapshot, time slider, calculator, and recognition of some strokes. Compared to the previous two tools, this one is different in a number of respects.

It supports quick editing, writing, sketching, and it gives a good overview, like a huge paper would, but it is more easily editable than paper. This is all useful in the analysis stage of architectural design, but the tool does not offer formal support for analysis. In terms of synthesis, the support for sketching is limited. In particular, the tool does not allow one to work with different layers. Moreover, there is no grid or recognition of architectural elements that are drawn. Where the tool provides some limited support for synthesis, it does not support evaluation at all. Communication is supported only in that information is made visible, but not in the sense that a group of people can edit the space at the same time during discussion.

2.1.4 **Creapro**

CreaPro is a text-based desktop application, developed by Robert L.A. Trost. There are many programs like CreaPro, popular among writers, less among architects. These programs has been developed to stimulate the user in various ways to make new associations and get new ideas.
This tool can be used on its own, but also worldwide through the Internet. First, a goal is set and a problem-definition has to be formulated. Then, this information is used as a starting-point for brainstorming. The system uses various tactics for generating ideas, such as free associations, pictorial associations, audio associations, analogies, vary attributes, wishful thinking, what if?, other views, I Ching, mindstepping, clustering ideas and evaluating ideas.

The interesting property is this tool concerns the support of creative thinking. However, because the generated ideas are purely textual and they cannot be related to each other or to images after being generated, the tool does not seem very suitable for architects. Moreover, some parts take too much effort to read and type in during the early design stage, where speed is essential. The tool does not support analysis, evaluation and communication.

2.1.5 Cocktail Napkin
The Electronic Cocktail Napkin (Gross and Do, 1996) was developed by M. Gross and E. Yi-Luen Do. Wacom digitizing tablets are used with the mouse or a trackball for input. Like Flatland aims to make an intelligent whiteboard, the Cocktail Napkin puts intelligence to ‘paper’. Its goal is to support informal drawing during the conceptual design stage. Added functionalities are creating, editing, and managing diagrams, information retrieval (from visual databases), simulation, design critiquing, and collaborative work. The Cocktail Napkin tries to recognize the glyphs that the user draws. Furthermore, Cocktail Napkin recognizes configurations at a higher level and can maintain constraints and maintain spatial relations among diagram elements.

Thus, the tool does not support analysis. As for synthesis, Cocktail Napkin is limited to two dimensions, but can recognize and parse configurations. It maintains an internal constraint representation that provides interactive edit behavior. For instance in drawing a floor plan, the user can drag and stretch it. The Napkin can display constraints in annotations that the user can edit.
The tool offers limited support for evaluation. Because of the feedback on the constraints, the user can better evaluate the design. Compared to the other tools, Cocktail Napkin has also more to offer to support communication. A team of designers can work on one tablet, interacting on the same sketch.

2.1.6 Teddy
Teddy (Igarashi et al., 1999) has been developed by Takeo Igarashi at the University of Tokyo. Two-dimensional strokes with a pen are automatically transformed into a three-dimensional polygonal surface. This is done by inflating the region surrounded by the silhouette drawn by the user, making wide areas fat, and narrow areas thin. The system works very intuitive and quick for making freeform models such as stuffed animals and other rotund objects. The system supports several modeling operations, such as cutting, bending, extruding and subtraction.

Figure 6: Screenshot from A Cocktail Napkin (Gross and Do, 1996)

Figure 7: Impression of Teddy (Igarashi et al., 1999)
Similar to most other tools, discussed in this paper, Teddy does not provide any specific support for analysis. With respect to synthesis, although Teddy works very intuitively, a disadvantage is that after the first stroke, whatever the user draws, it must be attached to the object. Only one object can be drawn. Furthermore, the tool does not support rectangular shapes, which architects often use. Finally, Teddy has nothing to offer in terms of supporting evaluation and communication.

Figure 8: Impression of Polyshop

2.1.7 Polyshop

The next tool that we discuss and evaluate is the PolyShop project that has been developed by the Realtime Interactive Simulation Group of the University of Central Florida. It is a project that works within a networked virtual CAD environment, which supports multiple users presence in the virtual world. In PolyShop, models can be directly manipulated in three dimensions, using data-gloves. The architects drafting-table is reproduced in the virtual world and added are virtual buttons, sliders, and controls. One can explore the model instantly. The networking component allows experts in other locations to be brought into the virtual world, and allows multiple modelers to work on the same database.

The strong point of this tool therefore relates to synthesis, evaluation and communication. It is not supporting analysis. The library of elements on the sliders provide means for quick modeling. It is unclear if one can start from scratch, which is desirable in the early design stage. Virtual Reality provides the ability to better generate and evaluate the design, by means of a three-dimensional walkthrough. A team of people can edit the model at the same time in the virtual world.

2.1.8 HoloSketch

HoloSketch (Deering, 1996) has been developed by M. Deering, at Sun Microsystems Computer Corporation. The input-device is a three-dimensional mouse/wand. It is much like PolyShop, but it has a ‘taskbar’ and it supports more drawing modes, not only plain objects, but also ‘polylines’ like ‘toothpaste-wires’, three-dimensional solid text, and random fractal mountains. This tool is a more sketch-like environment for designers. There is a possibility for a shared (net-worked) real-time three-dimensional
whiteboard. The potential advantages and disadvantages of Holoskeych are similar to Polyshop. It does not support analysis. In the phase of synthesis, the advantage of the sketch-like environment is that it maintains the ambiguity that is present in a sketch. The aspect of virtual reality is relevant for evaluation as it provides an opportunity to better generate and evaluate the design by means of a three-dimensional walkthrough. As far as communication is concerned, it is not clear whether more people can walk in the virtual model at the same time, and whether they can edit the model at the same time.

2.1.9 Sculptor

The next tool that we discuss is Sculptor (Engeli and Kurmann, 1996), developed by David Kurmann at the Eidgenössische Technische Hochschule Zürich (ETHZ). Sculptor allows a direct, intuitive access to three-dimensional design models. Modeling is possible with massive, closed objects or solids, but also with voids or negative volumes. The prototype supports interactive specification of objects, models and scenes with attributes such as form, geometry, color, and material. In Sculptor, objects can be grouped together hierarchically. The behavior of objects is reactive (motion and transformation), interactive (producing, attracting and destroying), and autonomous (agents containing knowledge, working on a specific task, having the ability to learn). There are three kinds of agents in Sculptor: agents that enhance the virtual environment, agents that take over tasks, and agents that help to test the design.

This agent technology implies the tool has some built-in support for analysis. It is also supporting the phase or aspect of synthesis as the modeling method implies both architectural elements and architectural relationships. Using both mass and void is linked up with the architectural way of thinking about mass and space. Although it is without measures it is not ambiguous, or sketch-like. Therefore, it does not work entirely intuitively and direct. Because it is a virtual reality, the tool allows one to generate and evaluate designs by means of a three-dimensional walkthrough.

Figure 9: Impression of Holosketch (Deering, 1996)
2.1.10 DDDoolz

DDDoolz (De Vries et al., 2000) was developed by B. de Vries, J. Jessurun, from the Design Systems Group at the Eindhoven University of Technology. It is a three-dimensional modeling tool, which comes very close to sketching. It works very intuitively and direct. The input is done with a mouse or a pen-like device. Dependent on which side of the starting block of a stroke the user holds the pen, the user sketches in a certain direction.

The tool is a typical sketching device, it does not support analysis. In the context of synthesis, the modeling method is closely related to the one in Sculptor, but it comes closer to a sketch-like expression. For this reason, it is more direct and intuitive like the use of paper and pencil. Common with the previous virtual reality tools, designs can be created and evaluated in a three-dimensional environment. Further evaluation, such as in Sculptor, is not implemented. Moreover, a team of designers cannot edit jointly in the same virtual world, which limits communication.

2.1.11 VIP

The Visual Interaction Platform (VIP) has been developed by Rauterberg, and IPO, Center for User-System Interaction. It allows users, grouped around a table, to interact in a space of virtual and real world objects. The display area includes two planes on which the object is projected. One is a table with a white surface, which is used as a horizontal working area. The other plane is the vertical communication area. On the horizontal part, one might see a floor plan and on the vertical plane one might see the interior that goes with this floor plan. Interaction is performed through physical blocks, which have a coating on the top that can be detected by a camera. Thus, the user can select and manipulate the model (or parts of the model). Everybody can grab a block to manipulate the scene and it works very naturally. Thus, the tool is especially appropriate for communication, but does not support the other aspects of design.
Figure 11: Impression of DDDoolz (De Vries et al., 2000) Sketch Straight, Orientate, Sketch Array, Sketch Curve

Figure 12: Principle of VIP. The figure in black is from the internet. Additional parts are in gray.

Figure 13: Impression of VIP
3 REQUIREMENTS FOR COMPUTER AIDED ARCHITECTURAL DESIGN TOOLS

3.1 General considerations
It is clear that architectural design in the early phase has many aspects and that several computer-aided tools could be used in principle. Architects use many different tools. The identification of requirements for computer-aided architectural design tools is an overwhelming task, not only because the many different problems that a designer encounters, but also because it is a very personal process. Some architects may not even like tools to solve problems, because it is their core business. The art of architecture is difficult to grasp and it is doubtful that a single tool could support the wide variety of design processes. Whatever tool, we feel that there should at least be room for expression and emotion by the designers in their design-process.

3.2 Comparison
From part 1 we can derive the following criteria. Intuitiveness indicates that it is easy to use the system. The architect must concentrate on the design-problems to be solved, not on how the computer-aided tool works. Designing is getting your ideas expressed, which sometimes may be volatile in one’s head. Architectural elements points to in how far architectural elements can be made or are in a library. Architectural relations, however says something about the behavior of these architectural elements with one another. Relation to context refers to how architectural elements and relations relate to the context, meaning relevant information, like the situation of the design. Functionality to support process means that the program or tool is capable of making suggestions during the architectural design process. Exploration of solution space designates whether or not the program or tool is capable of making (design-) variants or give hints of possible solutions. Combining different information has to do with the program or tool being able to combine for instance text or image. Combining different representations means that you can represent the information in different styles, like font for text or line-type for sketches. The information should preferably be represented in different ways: transparent, wire-frame, shaded, sketch-like, or maybe even in a materialistic way, applying texturing.

With Direct editing possibilities we mean that the schemata can be modified in real-time, and in a direct manner. Thorough evaluation of design points to in how far the information of the design is clear and mistakes can be detected easily. It may be very efficient to directly make three-dimensional models because we believe, the architect can perform better in that way, and it allows more direct evaluations. It implies, however, a different way of thinking, compared to working in two-dimensions.

Support for logging refers to the possibilities concerning saving and restoring information. If there is a checking-procedure whether the design meets the requirements and whishes of the client, it is stated in Link to the brief. Ambiguity means that information can be represented by the user in a vague or ambiguous manner.
Table 2: Overview of Comparison

<table>
<thead>
<tr>
<th></th>
<th>Organic Art</th>
<th>Treetown</th>
<th>Flatland</th>
<th>CreaPro</th>
<th>Cocktail Napkin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intuitiveness</td>
<td>+</td>
<td>x</td>
<td>++</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Architectural elements</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Architectural relations</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Relation to context</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Functionality to support process</td>
<td>++</td>
<td>++</td>
<td>-</td>
<td>++</td>
<td>0</td>
</tr>
<tr>
<td>Exploration of solution space</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Combine different information</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Combine different representations</td>
<td>+</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Direct editing possibilities</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Thorough evaluation of design</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Support for logging</td>
<td>0</td>
<td>x</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Link to the brief</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Ambiguity</td>
<td>+</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Personal expression in editing</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Teddy Holosketch Sculptor DDDoolz VIP

<table>
<thead>
<tr>
<th></th>
<th>Teddy</th>
<th>Holosketch</th>
<th>Sculptor</th>
<th>DDDoolz</th>
<th>VIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intuitiveness</td>
<td>++</td>
<td>+</td>
<td>++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Architectural elements</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td>Architectural relations</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Relation to context</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>Functionality to support process</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Exploration of solution space</td>
<td>0</td>
<td>0</td>
<td>+</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Combine different information</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Combine different representations</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Direct editing possibilities</td>
<td>+</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Thorough evaluation of design</td>
<td>+</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Support for logging</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Link to the brief</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ambiguity</td>
<td>+</td>
<td>++</td>
<td>0</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>Personal expression in editing</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

+ is supported or good  - is not supported or bad  ++ is very good
x does not apply, not known  0 neutral  -- is very bad

(Important is that the architect should not be restricted by making too precise representations of his ideas for it may stimulate to see things from a ‘different perspective’), while in Personal expression in editing is pointed out whether people can give this information a personal touch, or can use the tool in an expressive manner.

Keeping in mind the criteria, the discussion of the various tools leads to the following conclusions. Organic art and Treetown are programs, which are purely based on images. Textual information, such as annotations, is needed. Furthermore, direct editing is not supported. Organic Art’s strength lies in exploring the solution space, but it does not support any architectural aspects like Sculptor does. Treetown can give insight into structures implied in the design, but due to lack of direct editing and placing the design in its context, architects may find this tool too limited. Textual
information is well supported however by Flatland, but sketching is not. This is because one cannot put layers on top of each other. On the other hand, in The Cocktail Napkin one can, plus it supports the process through constraints and recognition.

CreaPro is not meant for architects, but a tool such as CreaPro might broaden the solution space. The improved tool should include sketches, images, and pictures, which could be edited and related as well. Further it should include the making of diagrams and schemas of ideas.

Teddy and DDDoolz have the great advantage that they work very intuitively and direct. The model is actually being ‘sketched’. A great disadvantage of Teddy is that it does not allow one to develop two or more objects in the same session. DDDoolz has this problem as well, but you can solve this by deleting blocks. DDDoolz can relatively easy be improved by supporting architectural elements and relations. To improve the ambiguity, multiple representations could be added as well.

Like Teddy, Holosketch is also ambiguous. In sketching it comes close to DDDoolz, but it is not very intuitive. Holosketch has too many controllers in order to work intuitively. Sculptor is also more complex than DDDoolz, but in Sculptor evaluation is well supported through agents and architectural additions. The architect can easily work with both solids and voids. We can imagine that deforming basic shapes like with modeling-clay or sculpturing or carving would be a good addition. VIP’s strongest point is the communication because editing is very intuitive and supports the discussion.

4 CONCLUSIONS

The aim of this paper has been to discuss a set of selected tools, developed within and outside the architectural design discipline, and assess their potential applicability to architectural design. This literature was conducted to better understand the nature of architectural design in the early design phase and to learn about the various tools that are already available to better position and specify the tools that we plan to develop in our research program. The discussion emphasized that the many different aspects of the early design stage either require a range of separate tools or a more integrated tool of complex requirements, which are stated above.

The aim of the E3DAD project is to develop an architectural design tool that fills in the gap not yet covered by other CA(A)D tools. There is no relation to the brief or to the context or situation of the design (see Table 2). As for the latter part, there are some Geographical Information Systems (GIS) that might support this. Also personal expression in editing and combining different information and representation are not well supported in the tools that were compared in this paper. These seemed very important to architects. As for making a tool for architects there should be definitely support for architectural elements and making architectural relations. Since our intention is to make a tool for the early stage of the design process, it should also support ambiguous information. We are working on a CAAD-tool that supports these requirements and hope to report on its developments in the near future.
5 ACKNOWLEDGEMENTS

The current paper stems from a Ph.D. project that is part of the VR-DIS research platform (Virtual Reality – Design Information System). The goal of this initiative is to explore and develop the use of Virtual Reality technology beyond visualization. Three specific fields of research are identified: Interactive Design System, Distributed Multidisciplinary Design, and Interactive Measurement of User Reactions. The present Ph.D. project is part of the Interactive Design System, called the E3dAD project (Easy 3-dimensional Architectural Design). This project is a collaboration of three groups: the Design Systems Group (Faculty of Architecture, Building and Planning), the Computer Graphics Group (Faculty of Mathematics and Computer Science), the Center for User-System Interactions (IPO).

6 REFERENCES

Brainstorm-session: http://www2.ds.arch.tue.nl/projects/E3DAD/Bstorm.htm
Organic Art http://www.artworks.co.uk/flash/
Polyshop http://www.vsl.ist.ucf.edu/groups/ieg/polyshop/index.html
Treetown http://www.hitl.washington.edu/projects/treetown/
VIP: http://www.ipo.tue.nl/projects/vip/system.html