Development of an intuitive 3D sketching tool

Bauke de Vries, Joran Jessurun[1] and Maia Engeli[2]

[1]Eindhoven University of Technology
Design Systems Group (www.ds.arch.tue.nl)
[2]Eidgenossische Technische Hochschule Zürich
CAAD group (caad.arch.ethz.ch)

ABSTRACT

3D sketching is a confusing description of an architectural design means. It is confusing, or even worse apparently wrong, because traditional sketching is inherently a two dimensional activity. Though the final stages of design are currently well support by CAD packages, almost every architect prefers paper and pencil for the early sketching phases. The challenge is to develop a (computer supported) design tool that is as direct and intuitive as paper and pencil. The computer enables us to directly map our spatial mental model into 3D rendered images. As such a new kind of design means is created which is best indicated as an intuitive 3D sketching tool. Within the VR-DIS research programme of the Design Systems group of the Eindhoven University of Technology a tool named DDDoolz has been developed as an experimental 3D sketching tool. This paper will report on the preliminary phase in which several input devices such as mouse, bird and voice were tested. For this purpose simple prototype applications were implemented. Building upon these experiences a functional brief was defined for the sketching tool. The system design will be elaborated in this paper using OO schema techniques. Because of its limited yet powerful functionality, a comprehensive system description can be presented. The application has been used in a first years CAD course. Students' experiences will be discussed demonstrating the strengths and weaknesses of DDDoolz. In conclusion, a list of improvements will be presented and the future directions are indicated that will be followed in regard of the continuing research on design and decision support tools.

1 INTRODUCTION

Sketching in 3D seems to be a paradox. The sketch activity is inherently 2D since it is executed in a plane on a flat surface using some drawing device (e.g. a pencil). Three dimensional creation and manipulation of objects presumes the activity being executed in a 3D environment on spatial objects.

Nevertheless, sketching in 3D represents a design activity that plays a very important role in the architectural design process and it has attracted many researchers ever since the Electronic Sketchpad of Ivan Sutherland (Sutherland 1963). The early efforts tried to interpret the line drawings made on an electronic drawing board. Computational limitations of the hardware in those days seemed to be a major bottleneck. Neural networks were a promising technology for handling the drawing pattern recognitions algorithms (Vries and Wagter 1989). Since then, sketching has

become a research issue with a heavy geometric modeling aspect, searching for drawing and editing algorithms that adequately support the user in creating and modifying 3D objects (Zeleznik, Herndon and Hughes (1996), (Igarashi, T., S. Matsuoka and H. Tanaka 1999). On the other hand 3D modeling is researched from a domain context, i.e. architectural design, to support user reflection on (building) objects and spaces (Gross and Yi-Luen 1996), (Kurmann1998), (Donath and Regenbrecht 1999).

In this paper DDDoolz will be presented, which is a new innovative dedicated system for mass study and spatial design in the early design stage. It is a low-end user system in the sense there is almost no learning time and the software will run on standard computers.

2 3D MODELLING SUPPORT

Current commercial CAD packages do support 3D modeling but at the cost of a lot of effort. 3D modeling consists of two basic elements, namely viewpoint management en geometrical operations. Since the modeling still takes place on a surface (i.e. the monitor screen) we need extra features to support us in managing the right view on our designed object. Multiple views only partly solve this problem because the user has to preset a viewpoint position and does not really know if the created object will obstruct the view. The result of geometrical operations depends on whether the operand is a geometrical basic primitive (e.g. a cube, sphere, etc.) or a parameterized building element. In the first case (e.g. 3D Studio Max) a stack of operators is used for describing the desired result, which requires quite a lot of knowledge from the user about the operator functionality. In the latter case (e.g. Archicad) parameters like length, width, and height can be manipulated to adjust the result. However, in the early design stage the architect does not articulate his/her thoughts in these terms.

Apart from the viewing and geometrical modeling capabilities, the user interface plays an important role. Currently all commercial CAD packages work with the window, menus, and mouse interface. We feel a designer should use his/her hands or body and directly operate upon the 3D object without the need to select the appropriate command beforehand. By moving one's head and body the designer should be able to observe the object from the preferred position or even better, experience the designed object.

Before actually designing and developing a sketch-like tool the following subjects were explored and researched: geometrical representations, input devices and the application domain.

2.1 **Geometrical representations**

Ideally speaking the user should not be aware of the geometrical representation. Performing an object creation operation or an editing operation should be as simple as grabbing the object and modify it as you would have done in reality. Things become really interesting if you can perform operations that cannot be done in reality.

Current status in geometry modeling is that geometrical representations (e.g. CSG, surface models) suit a specific shaping purpose and thus are less suitable for other purposes. Conversion between representations (e.g. meshing) is sometimes supported but in any case the user has to have knowledge about geometrical representations. Conversions back and forth are usually problematic.

Geometrical representations of (parts of) the model change while the design evolves. Maintaining consistency between all geometrical representations of an object over the complete design process is practically impossible. Nevertheless within certain limits the designer wants to move back and forth in time. In this time interval geometrical representation consistency must be supported. This condition demands sketch and shape interpretation knowledge captured in the design system.

2.2 **Input devices**

Thinking about 3D modeling it seems a logical step to use 3D input devices, also often indicated as 6 Degrees Of Freedom devices. An example of such a device is the Flock of Birds (Ascension), consisting of a sensor, which translations and rotations are recorded as changes in a magnetic field.

An experimental application has been developed for testing the device. The application consists of a pen and drawing board in a 3D environment. The user has a fixed perspective view on the drawing board and the pen could be moved in any direction. If the pen hits the drawing board a rectangular block is created until the pen is lifted from the drawing board. The height of the block depends on the length of the pen on the upper side of the drawing board.
Tests with the experimental application show that:

- The sensor is very sensitive to metal in the neighborhood (e.g. a steel frame in the furniture)
- The pen is technically difficult to control. Precautions in the software have to prevent the pen from becoming very 'shaky'.
- The hand-eye coordination is uncomfortable. The distance between the sensor and the monitor is often too large to keep control.
- Force feedback from the drawing board is badly missing.
- The sensor itself seems a bit awkward if used as a pen.

With these experiences in mind we looked for a device that suits the task better. A logical choice would be a phantom consisting of an arm-like device with a pen, located on the desktop close to the monitor. Unfortunately these devices are very expensive. Another solution that came to the market is a touch sensitive LCD screen (Wacom). Our experiences with the device so far are very positive.

Apart from the single hand input device there are of course more options, such as the use of the none-dominant hand, the use of the head and voice recognition. In our experimental application we tested voice recognition for changing color. The voice recognition software performed very well for this very simple purpose of course, without any training.

2.3 **Application domain**

The importance of sketching in the architectural design process goes without saying. In our opinion VR offers new, yet unknown possibilities for architectural design (Vries, Achten 1998). As a metaphor for 3D modeling we choose claying. Here a lump of clay is deformed while rotating it on a round table. In our case instead of a lump of clay, a mass model of a building is located on the table, allowing for a fine impression of dimensions, shape composition etc, while modifying masses and spaces.

3 FUNCTIONAL SPECIFICATION

In the VR-DIS program [www.ds.arch.tue.nl/research] at the Eindhoven University of Technology an interactive architectural and engineering design system is developed. A series of PhD students all research a specific part of this rather ambitious system within well-defined framework. Since the VR-DIS programme focuses on the early design stage a strong need was felt to develop a dedicated tool for this purpose and not rely on existing designing and drawing packages. Communication with others packages is not our primary research goal but is supported through the use of the design information database implemented as a feature model (Leeuwen 1999).

Based on the above described research experiments, the objectives for the development of DDDoolz were:

- Minimal set of commands
- Easy creation and manipulation of masses and spaces
- Easy visual evaluation of the created model
- Implementation of the claying metaphor
- Use of the LCD sketchpad
- Integration with design information database

4 SYSTEM DESIGN

The system architecture of DDDoolz is presented by a Class diagram using UML notation (Booch, Rumbaugh and Jacobson 1998). The Class structure and the characteristics of the objects are explained but the Class methods will not be discussed in detail. The system functionality will be elaborated briefly by describing each command.

The DDDoolz Class structure branches into two other Classes representing the two main system parts, namely the User Interaction and the 3D World. The UserInteraction object can perform any kind of task. A task typically operates upon the World Objects and its components. Which kind of task is executed is determined by the UserInteraction Mode. Tasks require user input which is obtained by continuously polling the InputHandler Class.

4.1 **Class diagram**

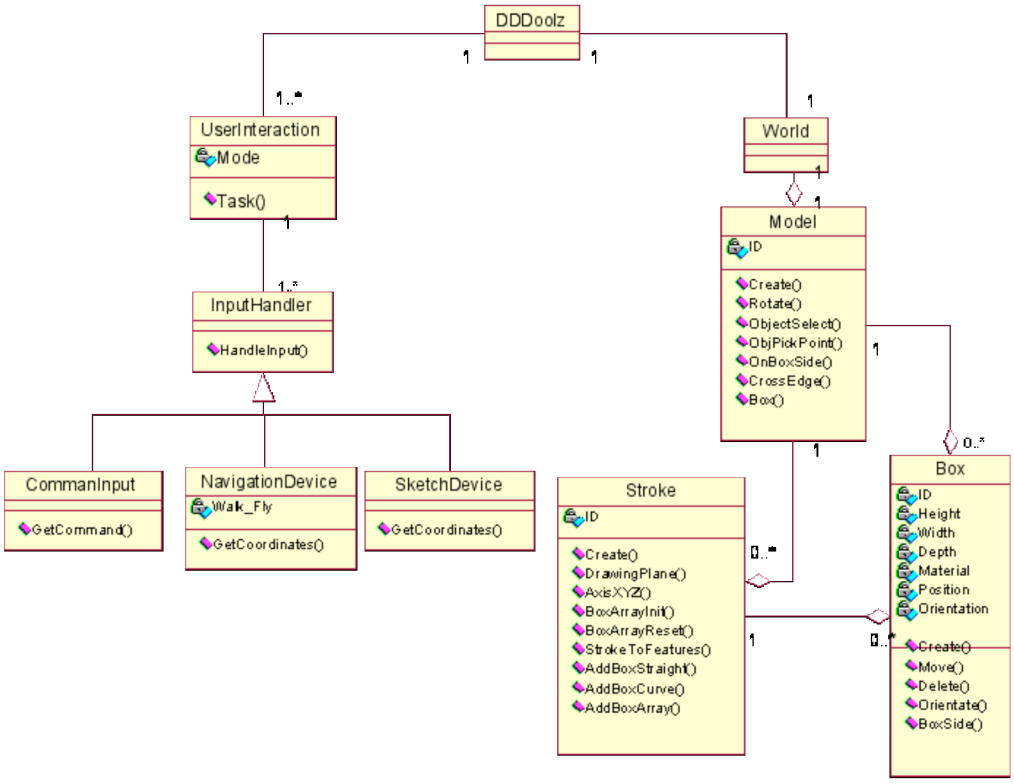


Figure 1: **DDDoolz Class diagram**

The InputHandler Class retrieves input from either subclasses: CommandInput, NavigationDevice and SketchDevice. Data are passed through to the UserInteraction Class along width an identifier for the input device type. In DDDoolz command input has been implemented by key pressed button, by screen button and by voice recognition. The NavigationDevice has been implemented by a (2DOF) mouse and by a 6DOF tracker. The SketchDevice has been implemented by a (2DOF) mouse and by a LCD sketchpad.

The UserInteraction Mode is set by command input. The UserInteraction Mode determines which Task(s) will be executed and thus which operation can be performed upon the Wolrd objects and it components.

The World Class contains only one object, namely the model that is or will be created. The Model object has methods to determine if and where the sketch device pointed at a Box, and to find out whether the sketch device crossed the Box edge. If the latter occurs then a Stroke object will be instantiated. The Stroke object has methods to determine the drawing direction of the sketch device, and methods for adding new Boxes to the stroke. Which method is used depends on the sketch mode, namely straight or curve. Moreover in case of drawing a U shaped stroke, an array of Boxes will be generated to easily create solid planes. Adding Boxes is established by

instancing the Box Class. Boxes are defined by their geometrical parameters in the 3D world and by their visual presentation. A Box has methods to move, to delete and to orientate each individual Box. To do so a Box can be requested to return the Box side and the pick point of the sketch device.

4.2 **DDDoolz functions**

The DDDoolz functional hierarchy is structured as follows:

- Sketch
  - Straight
  - Curve
- Navigate
  - Walk
  - Fly
- Orientate
- Move
- Delete
- Rotate

Sketch mode is decomposed into two sub modes, namely Straight and Curve. In Straight mode a new Box is created next to the previous one (Figure 2). The copy direction is determined by the sketch device drawing direction over the Box side. In Curve mode the drawing direction will determine the new Box's orientation (Figure 5). The position of newly created Box is calculated to exactly match the outside edge of the previous Box. Navigate mode is decomposed into a Walk mode and a Fly mode.

In Walk mode the viewpoint is set to a fixed distance to the ground whereas is Fly mode it is not. In Orientate mode an individual Box can be rotated around the selected Side's midpoint (Figure 3). From there new strokes can be made in the previously set direction. If in Straight mode a U shaped curve is drawn, the system will generate an Array of Boxes. Move mode allows for moving a Box continuously or over a snap distance equal to the Box's edge length. In Delete mode Boxes that are pointed at will be removed from the scene.

Rotate is a switch that will make the model rotate around its center or fix it in its current position. While in Rotate mode all other functions are available. Consequently the designer can sketch for instance while the model is slowly rotating.

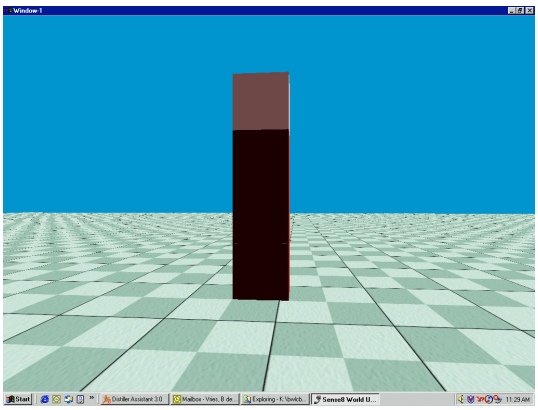


Figure 2: **Sketch Straight**

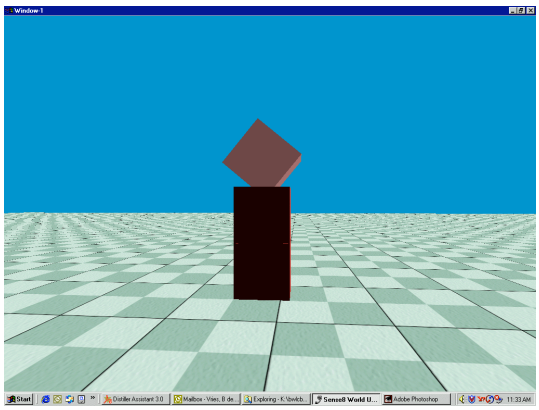


Figure 3: **Orientate**

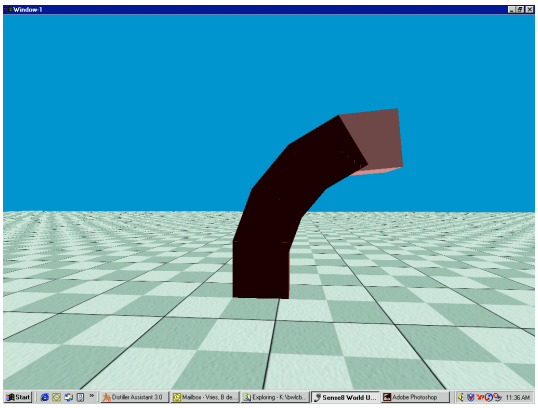


Figure 4: **Sketch Curve**

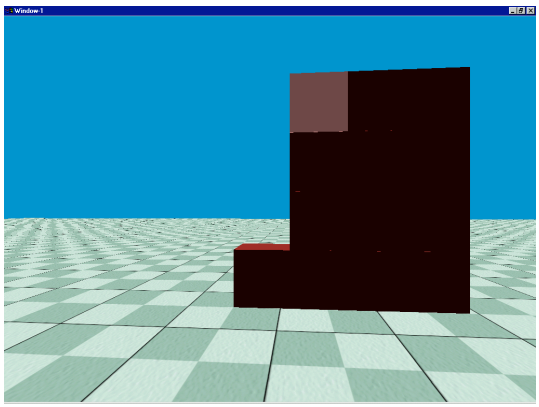


Figure 5: **Sketch Array**

5 SYSTEM IMPLEMENTATION

5.1 **Application development**

DDDoolz was developed using WorldUp from Sense8. In WordlUp Input handlers are available for an extensive set of commercially available input devices. A 3D scene is constructed as a hierarchy of geometrical objects. WorldUp has a few geometrical object types of its own (Block, Cone, Cylinder, Sphere) and it can import geometries in 3DStudio format. Special objects such as Viewpoint and Window are available for displaying the scene. To each object a script can be attached that will perform a specific task. Tasks will be executed in every rendering cycle of the scene following the objects hierarchy.

WordlUp comes with a player that can be distributed freely. The latest version of DDDoolz is almost completely implemented with WorldUp functionality including

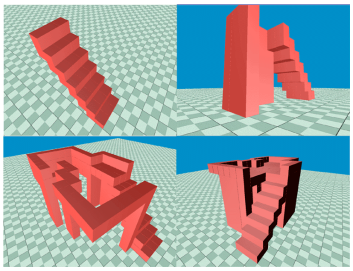
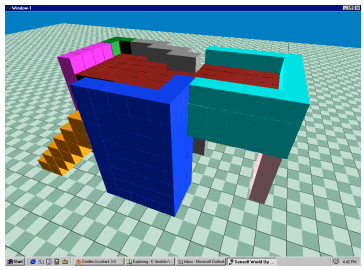
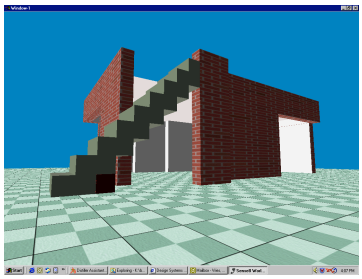
buttons for switching Mode. Consequently DDDoolz can be distributed as shareware and even be adapted to different hardware platforms with minor changes.

Evidently DDDoolz models can be stored its own format and reloaded. DDDoolz can import externally created objects in 3DS format. Imported objects are treated as Imported Geometries in DDDoolz. An Imported Geometry cannot be edited, it only can be moved or deleted. An import script (VBA) has been developed to import DDDoolz models into Autocad. This facility allows for post editing with Autocad and 3Dstudio. For instance the Autocad unify operation can be used to group all Boxes into one single Solid.

5.2 **Sample session**

To demonstrate the use of DDDoolz for architectural modeling purposes, a sequence of design stages for a specific design task will be explained. The design task is to create space that is attractive from the inside and from the outside because of its composition and its dimensions with only a few materials.

Table 1: **Sequence of design stages with DDDoolz**

The first picture is a compilation of four stages showing how the model is constructed using the standard DDDoolz features. The second picture shows the completed model. To identify building elements, Boxes are colored. The last picture presents the final design using textures. Therefore the previous DDDoolz model was imported into Autocad. In Autocad all Boxes from one color were unified into one solid. The Autocad model was imported into 3Dstudio, where textures were added. The 3Dstudio model finally was imported again into DDDoolz.

6 DISCUSSION

6.1 **Experiences**

DDDoolz has been used in a first years CAAD course at the Eindhoven University of Technology. Notably, all students in Eindhoven have to buy a laptop computer at the beginning of their study. The objective of the CAD course is to give the student insight on computer usage in an architectural and design methodological context. Students practice with the drawing, painting and traditional sketching tools. DDDoolz

is used to explore mass and space study in an un-traditional way. The students' task was to create an exhibition center using DDDoolz and to pay special attention to experiencing the model from the outside and from the inside. The exhibition could be of any nature and the object (e.g. a sculpture) had to be created in Autocad and then imported in DDDoolz.

As we had hoped for, the students had no problems using DDDoolz despite the very limited time. Because the block array generation feature was not yet implemented in the student edition, most designs appear to have a quite open structure. Some students took the sketch functionality quite literal creating quite massy, though sketchy shapes. Still a lot of student did not take advantage of the possibility to design from the inside. Here, it seems that traditional sketching and designing is obstructive to new tools like DDDoolz.

DDDoolz has also been tested by a large architectural office in the Netherlands on our request to explore application in practice. Here, the architect used DDDoolz for designing masses corresponding with spaces. The architect liked the very intuitive interface and the ease of viewing the design. Major restriction that was reported was the impossibility of changing the Box dimensions. Working on a spatial plan, he wanted to adjust the Box dimensions to the areas that follow from the client's brief. Interesting in this case is that DDDoolz was not meant to be used in this way. Boxes are considered is the building block or voxels for creation of masses. The architect however, related Boxes to spaces.

6.2 **Improvements**

Based upon the experiences the following system extensions can be listed:

- Change Box dimensions
  Change height, width, length by dragging a Box side.
- Group Objects into Building Elements
  Select strokes and/or individual Boxes and group them to building elements. This task is currently performed outside DDDoolz.
- Interpret strokes
  Extend the U shape stroke interpretation with more stroke shapes. A vocabulary of strokes is required to support intuitive shape creation.
- Boolean operations
  Perform unify, intersect and subtract CGS operation upon Boxes, Strokes and Building Elements.
- Store DDDoolz models as Feature models
  Using the Feature manager the DDDoolz entities: Strokes and Boxes can be stored as Features. This facility is almost implemented.
- Texture mapping
  Support texture mapping on Boxes, Strokes and Building Elements. Adding textures and setting transparency should be manageable for each separate Side of a Box, Stroke or Building Element.

6.3 **Future directions**

DDDoolz as part of the VR-DIS research platform can be extended in many directions. Some researches already have started but are still explorative. Below the research initiatives will be discussed briefly.

- E3DAD.
  In the E3DAD project the faculty of Architecture cooperates with the faculty of Computing Science and The Center for User-System Interaction on the development of a design tool for Easy 3D Architectural Design. For more information please refer to (Segers 2000)
- Implicit Constraints.
  During the architectural design process relationships between objects come into existence such as wall connections, wall-floor connections etc. Preferably these so-called implicit relationships should be inferred by the design system to support intuitive manipulation of objects. A 2D prototype system was developed to research the potentials of implicit constraints (Skatt 2000).
- Feature View.
  M. Coomans develops a Feature View application as part of his PhD research (Coomans, Timmermans 1998). The Feature View will allow for creation and alteration of Features. Geometry Features must be communicated with the DDDoolz presentation in WordlUp. New Features can be added to the existing DDDoolz Features. For instance building components (e.g. a specific door) from the Feature Library can be instanced and associated with a Box geometry Feature representation of a door. Next step is to infer a routine that creates a parameterized geometry Feature representation of the door from the Box Feature representation. After this, door parameters (length, width, depth, material) are accessible through the Feature view.
- Performance evaluation.
  Though design information in the early stage is very incomplete, domain expertise can support the designer in learning about design solutions for similar cases and about the consequences of design alterations. DDDoolz can be used is a modeler for easy creation of shapes that are part of the design knowledge database (Groot 1999).
- New input devices: Head tracking, Phantom.
  To improve the ergonomics of the design system, new input devices can be tested. A head tracker should support better viewing of the model, responding to the head movement. A sketch phantom consists of an arm-like apparatus, holding a pen with force-feed back. With such a device touching a mass will indeed feel as hitting an object, thus allowing for intuitive use of the sketching device.

7 ACKNOWLEDGEMENTS

DDDoolz was developed during a sabbatical visit of Bauke de Vries to the CAAD group of the ETH Zurich. Progress was discussed with many group members,

especially with Kai Strehlke and Andrew van de Moere. Back at the Eindhoven University of Technology the development process was continued within the Design Systems Group.

8 REFERENCES

Achten, H.H. and B. de Vries (2000) DDDoolz: A Virtual Reality Sketch Tool for Early Design, In: *Proceedings of The Fifth Conference on Computer Aided Architectural Design Research in Asia*, (forthcoming).

Booch, G., J. Rumbaugh and I. Jacobson (1998) The Unified Modeling Language User Guide, Addison Wesley, Reading, Massachusetts.

Coomans, M. and H.J.P. Timmermans (1998) A VR-User Interface for Design by Features, In: *Proceedings of the 4-th International Conference on Design and Decision Support Systems in Architecture and Urban Planning*, Maastricht, The Netherlands.

Donath, D. and H. Regenbrecht (1999) Using immersive Virtual Reality systems for spatial design in architecture, In: J. Verbeke et al. (eds), *Proceedings of the 2nd AVOCAAD Conference*, Brussels, Belgium.

Groot, de, E.H. (1999) Intergrated Lighting System Assitant, Ph.D. Thesis, Department of Architecture, Building and Planning, Eindhoven University of Technology.

Gross, M.D. and E. Yi-Luen (1996) Ambiguous intentions: a paper-like interface for creative design, In: *Proceedings of UIST '96*, Seattle, Washington, pp. 183-192.

Igarashi, T., S. Matsuoka and H. Tanaka (1999) Teddy: A Sketching Interface for 3D Freeform Design, In: *Proceedings of SIGGRAPH '99*, ACM, Los Angels, pp.409-416.

Kurmann, D., (1998) Sculptor: How to design space? In: T. Sasada et al. (eds), *Proceedings of The Third Conference on Computer Aided Architectural Design Research in Asia*, pp. 317-325.

Leeuwen, J.P. van (1999) Modeling Architectural Design Information by Features: An approach to dynamic product modeling for application in architectural design, Ph.D. Thesis, Department of Architecture, Building and Planning, Eindhoven University of Technology.

Segers, N., B. de Vries and H.H. Achten (2000) Architectural Design Tools, In: *Proceedings of the 5-th International Conference on Design and Decision Support Systems in Architecture and Urban Planning*, Nijkerk, The Netherlands, (forthcoming).

Skatt, E. (2000) Sketching with multiple interpretations of user input, Masters Thesis, Department of Computer Science, Eindhoven University of Technology.

Strehlke, K. (1999) An environment for collaborative three-dimensional modelling over the internet, In: J. Verbeke et al. (eds), *Proceedings of the 2nd AVOCAAD Conference*, Brussels, Belgium.

Sutherland, I. (1963) Sketchpad: A Man-Machine Graphical Communication System, In: *Proceedings of the Spring Joint Computer Conference*, Spartan Book, Baltimore, MD, pp. 329-346, 1963.

Vries, M. and H. Wagter (1989) A CAAD model for use in the early design phases, In: *The Electronic Design Studio: Architectural Knowledge and Media in the Computer Era*, Cambridge, pp. 215-228.

Vries, B. de and H.H. Achten (1998) What offers Virtual Reality to the Designer, In: *Proceedings of the int. Conference on Integrated Design & Process Technology*, Berlin, Germany.

Vries B. de, A.J. Jessurun and R.H.M.C. Kelleners (2000) Using 3D Geometric Constraints in Architectural Design Support Systems, In: *Proceedings of the 8-th International Conference in Central Europa on Computer Graphics, Visualization and Interactive Digital Media 2000*, University of West Bohemia, pp. 76-83.

Zeleznik, R. C., K.P. Herndon and J.F. Hughes (1996) Sketch: An Interface for Sketching 3D Scenes, In: *Proceedings of SIGGRAPH '96*, ACM, Vol. 30, No. 4, pp. 163-170.