



## **AN ENVIRONMENT FOR COLLABORATIVE THREE-DIMENSIONAL MODELLING OVER THE INTERNET**

### AUTHORS

Kai Strehlke  
Architecture & CAAD  
ETH Hönggerberg  
Zürich, Switzerland  
strehlke@arch.ethz.ch

### ABSTRACT

This work results from a postgraduate thesis done at the chair for Architecture and CAAD at the ETH in Zürich. It presents a three-dimensional modeller written in VRML (Virtual Reality Modelling Language) which allows collaboration over the Internet. The aim was to create a modeller with a set of very simple tools to create complex forms. It was intended to realise the program in a way, that the user is always able to understand and control the process of form generation. A second goal was to connect the process of form generation to the output of a database. It was intended to interact between a locally created object and objects from the database created by other students.



## AN ENVIRONMENT FOR COLLABORATIVE THREE-DIMENSIONAL MODELLING OVER THE INTERNET

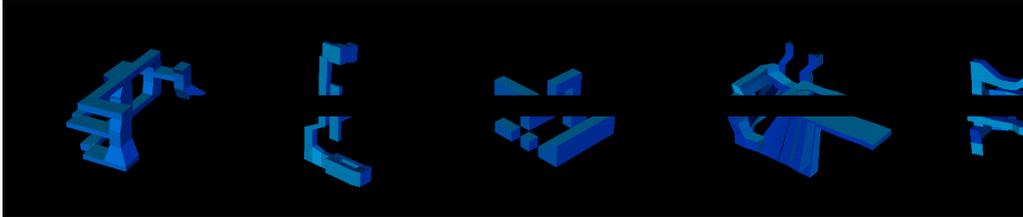


Figure 1.

### 1. Introduction

Computers have become commonplace in architectural practice. Highly complex CAD packages have been developed to help the working process in offices. Most CAD packages claim to support the entire work process from the first concepts until the cost calculation of a project. However, the applicability of existing commercial software in the early stages of a design process can be doubted, because most programs are far too complex to allow an intuitive way of designing. As a result, even today a design often is done in a traditional way and only afterwards it is translated into a digital form.

As in architectural offices, computers also have found their place in universities, for teaching architecture. In the educational environment the work process differs very much from the daily work in an office. Whereas in an architectural office, the design process accounts for only a small percentage of the overall office work, in architectural education the design process receives the main focus.

CAD packages are optimised for the production of plans and models of a completed design, not for the first design steps. Therefore, this software is not best suited for the use in education.

### 2. Traditional Design Education

In the Department of Architecture of the Swiss Federal Institute of Technology (ETH) students design classes are held in designated studio spaces. The essential advantage of this situation is that students can communicate during the entire design process. It allows them to discuss the different projects, thereby enabling them to learn to find the strong and weak points of their design. They receive a lot of input from the others students and they can control their own design process. When someone is stuck in his or her design process the social support qualities of the studio come into play.

A second important advantage is that students work with many different tools. Sketches are done with pencils, charcoal, watercolour, etc. Different materials are used for modelling, like cardboard, clay, wood or plastic. For each design intention an appropriate media can be chosen to express the idea in a simple but effective way.

## 2.1 The Use of CAD Software in Design Education

Designing with computers differs substantially from a traditional design process; especially if the design process takes place in an educational setting. The use of complex CAD packages in design education brings crucial changes. Most often students are working in front of a PC or a workstation.

Firstly, the work process is hidden from the others students. The possibilities of communicating the design steps are no longer available, because the whole work is very often located in one or more files. Only the final result can be seen, when perspectives are rendered and plans are printed out. Secondly, people use complex CAD tools for their design. The use of these tools requires a lot of time to learn the features of the package. Very often the students are not guided by their imagination and creativity but their design is restricted by how well they master the tool.

There is a need of adequate software in design education. It is important to point out the differences in processes going on in offices and in an educational environment in order to know which features have to be provided by the software for each of these. For educational purposes the tools should enable collaboration and communication between students who are working on the same project. The software should be so simple to use, that the possibility exists of an intuitive way of working with it.

## 3. A Prototype for Three-Dimensional Modelling Over the Internet

Given these concepts, the intention was to program a prototype which stimulates interaction between different users and provides the possibility to create complex forms with a set of very simple tools. The goal was to build a three-dimensional environment for modelling on the Internet and to connect this program to a database. It was intended to write a program which runs everywhere. The form generation process should be comprehensible and thus controllable. No filters should be applied, which would enable spectacular forms, but would complicate an understanding. In a first application step, a set of simple tools are provided to build a model. This modelling process is done locally without any connection to the database. In a second step, it is possible to submit the model to the database or to retrieve models from the database and interact with these.

### 3.1 Programming Tools

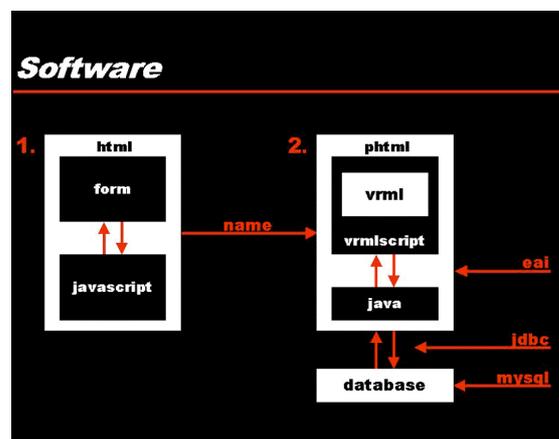


Figure 2 : Software components used.

The entire modeller works over the Internet and, is therefore written, using languages that can be read and interpreted by common browsers. The main part has been written in VRML (Virtual Reality Modelling Language) and VRMLScript. Other languages and file formats used are HTML (Hyper Text Mark-up Language), PHP3 (a scripting language called Professional Home Pages Version 3.0), JavaScript, Java, EAI (External Authoring Interface) and a MYSQL Database.

From a system point of view, first one submits a form with the name of the user. This form is embedded into an HTML page, with a JavaScript program checking the user's input.

This form is then send to the server, where a PHP3 program parses the name into a VRML file. The user receives an HTML file with both a VRML file and a Java applet embedded into it. On the client side of the connection, a browser, in this case Netscape Communicator 4.04 with a VRML Browser plug-in (Cosmoplayer 1.02), displays the VRML World. Through the EAI Interface the Java applet reads and writes data to the VRML scene and communicates this information through a JDBC driver to a MYSQL database. [Figure 2]

### 3.2 The Interface of the Modeller

The entire program is displayed in one full-screen window. The VRML scene takes up the full window and the complete menu of the interface is programmed inside VRML as a HUD (Head Up Display). It is divided in two horizontal rows of buttons on the upper and the lower part of the screen.

The menu of the upper part controls the modelling features; modelling is done locally on the machine. The menu on the lower part allows to connect to the database, in order to submit the work or retrieve objects for interacting with these. In order to make the interface comprehensible, the menu is reduced to the essential elements.

### 3.3 Modelling Tool

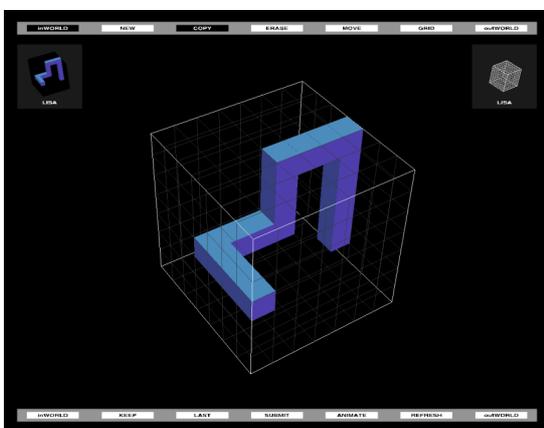


Figure 3 : InWorld modus.

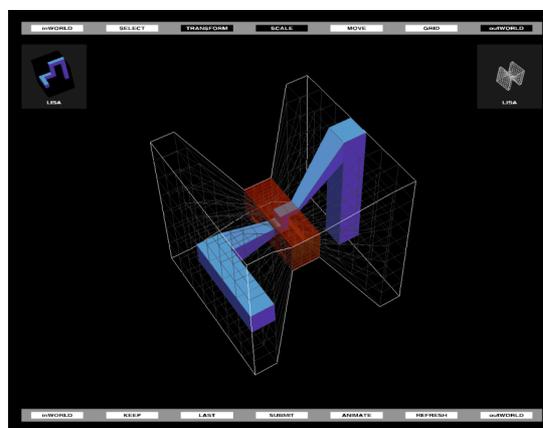


Figure 4 : OutWorld modus.

The modelling process is handled in two separate phases. These phases are called the InWorld and OutWorld modus. The InWorld modus controls the location of cubical forms within a grid structure, while the OutWorld transforms the grid structure by applying forces onto it.

In a first phase, cubes can be placed, moved and deleted within a frame structure of 7x7x7 units. To place new cubes in the grid one needs to click on a face of a cube. Afterwards, by dragging the mouse, it is possible to place new cubes into the grid structure in the same plane of the face clicked. To move cubes inside the grid, a similar mechanism is used. One needs to click on a face and drag the mouse. The cube will follow the mouse in the plane of the touched face. A collision detection mechanism controls this process. As such, it is possible to move a whole row just by pushing on one end. [Figure 3]

The second phase consists in the deformation of the grid structure. By activating the OutWorld modus the buttons in the upper centre switch to enable the OutWorld transformations. In this phase one has to select a part of the grid structure. For the selection, the grid structure is always displayed in an orthogonal way. The selection box is displayed in red. Once a part of the 7x7x7 grid has been selected, one can apply either a translation or a scaling transformation to the selected part. To apply a transformation one needs to activate the transform button. Subsequently, a deformation takes place by clicking on the selection box and dragging the mouse while being clicked. When the mouse is released a process recalculates the grid structure. As in the first phase, the translation is applied on the plane of the touched face of the selection box. [Figure 4]

It is always possible to switch between these two modes. Whenever the InWorld modus is selected, the whole scene is displayed as an orthogonal system. This system is more appropriate to control the placement of cubes inside the frame structure. Working in the orthogonal and transformed view allows a better understanding of the modelled object. The orthogonal view is well suited to understand the structure of the model, whereas the transformed view is obviously needed to control the shape of the form.

### 3.4 Interacting with the Database

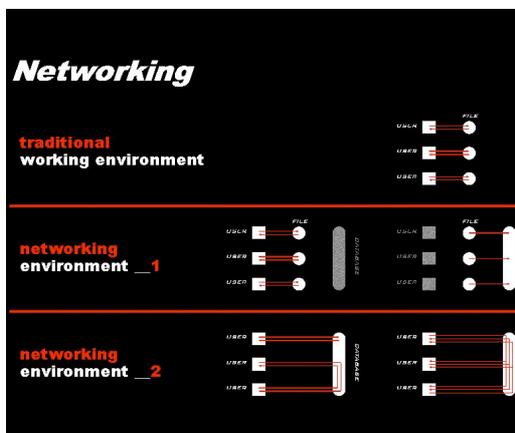


Figure 5 : Networking examples.

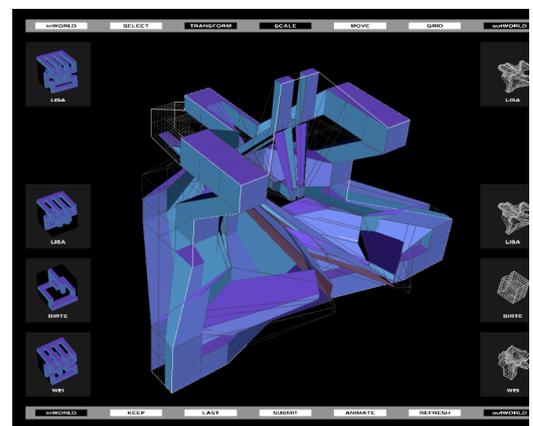


Figure 6 : Connection to the database.

In addition to the local modelling process, one can load the last three models from the database into the working environment. To display these objects, the left and right button from the lower menu open an iconographical view of these elements. On the left side the InWorld and on the right side the OutWorld views are displayed. To interact between the local model and objects from the database one only needs to click on the icons. By clicking on the InWorld icons a substitution takes place. By clicking on the OutWorld view a transformation process starts which morphs the actual grid structure into the grid structure of the object from the database.[Figure 6]

In figure 5 three different diagrams of work processes are shown. The first diagram shows the situation of a traditional way of working with a computer. A very strong relationship is established between user and computer. In this way nearly no communication takes place between different users. The second diagram shows a first step towards collaborative work. The user is still working with a local file. Only in a second step he submits his or her work to a database. This mode already allows a certain level of interaction. The third diagram illustrates that the user is no longer working with a local file, instead, submits his or her work directly to a database. Whenever an object has been submitted to the database, it can be seen by any other user.

#### **4. Conclusion and Future Work**

Although scripting languages are supposed to be slow and ineffective, this modeller written in VRML is surprisingly fast and stable. Using the computer knowledge background of an architect, VRML has been an excellent way to dive into 3D programming. Looking at the actual situation of VRML and the capabilities of Java3D as an API, this work has arrived at a point where it seems advised to port it to Java3D.

The modeller is quite powerful but has only very few modelling possibilities. An important task will be to find new features, to enhance the tool without making it too complex to use. An interesting feature from Java3D are the input possibilities. Finding an adequate input device other than a mouse or a joystick will make the modelling task a lot more intuitive. Until now the user is constrained to work in a grid of 7x7x7 units. It would be interesting to add a second level into the modelling process, so that once cubes are placed into the grid these could be substituted with more complex objects.

The database has a very rough interface at the moment. Although lots of information is collected, no intelligent way of browsing the database is available. For future work, the database should allow for retrieving objects by completion time or by author as well as displaying the relationships between the objects. An important issue will also be to allow more than one person to work simultaneously on the same model.

#### **5. Acknowledgements**

The author would like to thank all who have contributed to the thesis on which this paper is based. Special Thanks go to Prof. Gerhard Schmitt, Dr. Leandro Madrazo and Prof. Maia Engeli for their support and stimulation.

## 6. References

- D. Bienhaus, VRML+Java, Javasppektrum, July/August 1997, pp. 22-29.
- Mary Campione, Kathy Walrath, The Java Tutorial Second Edition, Object-Oriented Programming for the Internet, Addison Wesley, 1998.
- Andreas Gehmeyr, Multimedia-Anwendungen mit VRML und Java, Javasppektrum, May/June 1998, pp. 24-29.
- Joerg Kloss, Robert Rockwell, Kornel Szabo, Martin Duchrow, VRML97 Der neue Standart fuer interaktive 3D-Welten im World Wide Web, Addison Wesley, 1998.
- William J. Mitchell, City Of Bits, Leben in der Stadt des 21. Jahrhunderts, Birkhauser Verlag, Basel Boston Berlin, 1996.
- William J. Mitchell, The Logic of Architecture, Design, Computation, and Cognition, MIT Press, Cambridge, Mass., 1990.
- William J. Mitchell, Malcom McCullough, Digital Design Media, Van Nostrand Reinhold, New York, 1995.