

# OBJECTIVEFRAME - AN EDUCATIONAL TOOL FOR UNDERSTANDING THE BEHAVIOUR OF STRUCTURES

**J. Lindemann, O. Dahlblom and G. Sandberg**

*Division of Structural Mechanics, Lund University  
John Ericssons väg 1  
221 00 Lund  
+46 46 222 73 70  
struchmech@byggmek.lth.se*

## **ABSTRACT**

To understand the behaviour of structures subjected to loads the 3d beam analysis program ObjectiveFrame has been developed. One of the features of this program is that it can visualise the response of a structure to a user-defined load in real-time. This makes it suitable for use as an educational tool in design science, architecture and structural mechanics. ObjectiveFrame also implements a heads-up display used for displaying messages and toolbars. Most operations in ObjectiveFrame are implemented using a direct manipulation paradigm.

## **KEYWORDS**

Virtual Reality, Real-time visualisation, Structural Mechanics

## **INTRODUCTION**

Creating a 3D beam model is a complex task in most pre-processors, due to the complexity of the user interface. These programs are often designed as general pre-processors handling all kinds of finite element models. ObjectiveFrame was designed to solve some of these problems, creating an intuitive tool for experiments with structures and forces.

To enable users to experiment with a structure, ObjectiveFrame has the ability to visualise the response of structures in real-time, when applied a user-defined load. To make ObjectiveFrame intuitive and easy to use, most operations in the user interface are implemented using a direct manipulation paradigm, giving the users direct feedback of actions taken.

## CONCEPTUAL MODEL

A conceptual model is a mental representation of how an object works. An application user interface must convey the conceptual model to the user. ObjectiveFrame uses a conceptual model based on Christiansson [1], see Figure 1.

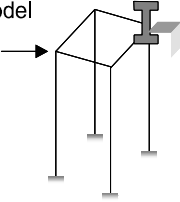
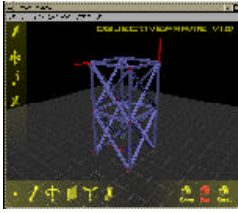

Domain	Product	Process
Application	Beam model 	Edit, change and move nodes, beams, loads, boundary conditions and materials.
System		User interface, editing functions, 3D graphics, view transformations, FE solver.
User	Students, Architects, Engineers 	Visualise behaviour of structures

Figure 1 - Conceptual model

In this approach the conceptual model is divided into three domains, the application, the system and user domains. The application domain is the problem area. The system domain is a description of the system. The user domain describes the users and the problems the users want to solve.

The application model is a 3D beam model using 6 degrees of freedom at each node, based on a Bernoulli beam theory.

Due to the low complexity of the user interface in ObjectiveFrame, it has been successfully used in the teaching of mechanics to architectural students. In this context the main user goals are:

- Provide an easy way of constructing a 3D beam model.
- Solving 3D beam problems
- Understanding the behaviour of structures and forces.

According to Preece [7], a direct manipulation system can have the following properties:

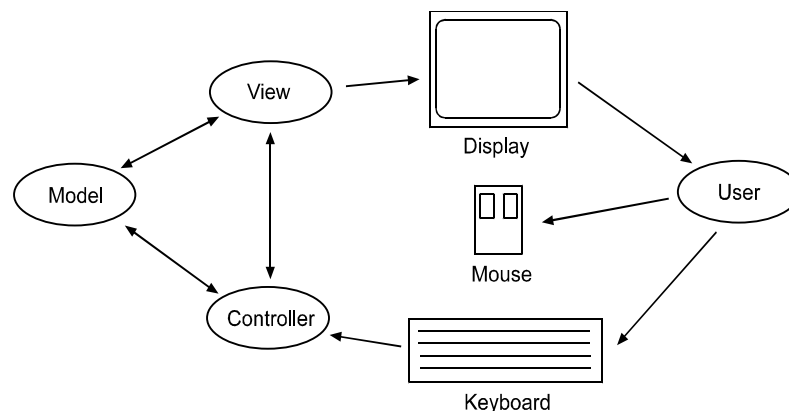
- Visibility of the objects of interest
- Rapid and reversible incremental actions
- Replacement of complex command language syntax by direct manipulation of the object of interest.

The system model of ObjectiveFrame was designed as a direct manipulation system with using the goals described above. Reversible action has however not been implemented yet.

ObjectiveFrame also uses the Model-View-Controller paradigm described in Dix [2]. In this paradigm the user interface is managed by three components:

- The model component, representing the application.
- One or more view components, responsible for displaying views of the model.
- The controller component, receiving input from input devices controlling the view and model components.

Figure 2 illustrates this paradigm.



**Figure 2 - Model-View-Controller paradigm**

ObjectiveFrame only uses one 3d perspective view of the beam model. The design principle used was to give the user the impression of directly manipulating a real 3d beam model. To further the experience, the beam model is displayed shaded using a lightsource.

## **USER INTERFACE**

The main user interface in ObjectiveFrame is the workspace. The workspace is the "workbench" on which the beam models are assembled and consists of an axis and a transparent grid. The user can change the workspace size to fit the model constructed. Object placement, movement and creation are done using a special 3d cursor. The position of the 3d cursor is determined by mapping the mouse coordinates on to the 3d grid. Movement perpendicular to the grid is done by holding down the [ctrl] button. 3d

cursor movement in a VR system would use some kind of 3d pointing device instead of the mouse. Figure 3 shows the workspace layout.

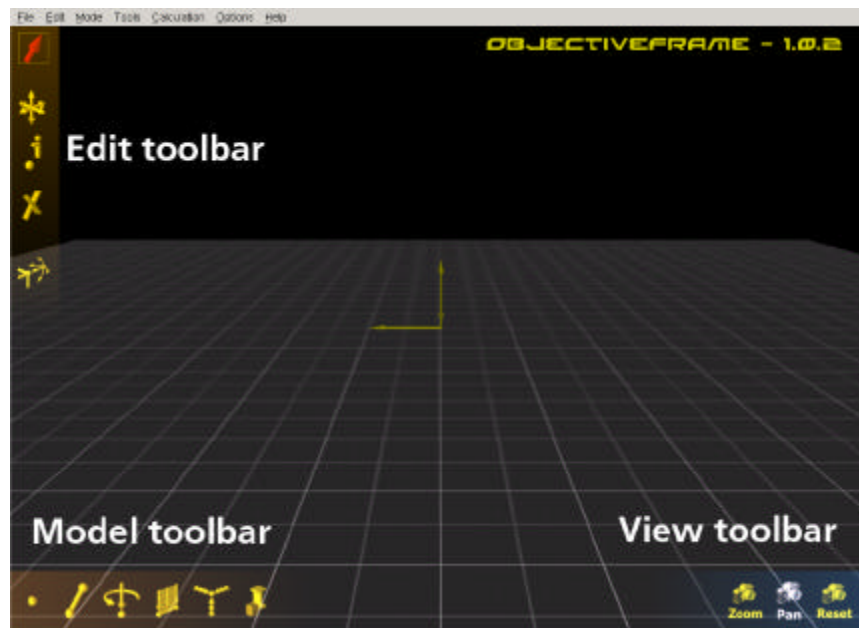


Figure 3 - ObjectiveFrame workspace and toolbars

Many modern applications provide toolbars for most user actions. The use of toolbars tends to reduce the available work area of the application. In ObjectiveFrame toolbars have been made transparent, so that the 3d workspace can be seen through them. To make it easier to find different toolbars coloured gradients are also placed behind the toolbar icons. When the mouse is over a toolbar icon, this icon magnified and its colour changed to white. A hint is also displayed in the middle of the screen. The toolbars are also visible in Figure 3.

The technique described is often found in many modern computer games, for example Half-Life [3].

ObjectiveFrame uses seven modes:

- Select
- Move
- Feedback
- Create nodes
- Create beams
- View/zoom
- View/pan

Clicking on a 3d toolbar icon switches between nodes. The current mode is indicated by a red square around the toolbar icon.

In the view modes the user can freely move around the beam model. Using the left mouse button always rotates the view. The right button is used to zoom or pan depending on the current mode.

In the selection mode objects can be selected. A selectable object is highlighted in white, when the mouse is over it. An object is added to the list of selected objects by clicking on it. Clicking on the workspace clears the selection. Selected objects can be moved, deleted, assigned loads, boundary conditions or material properties.

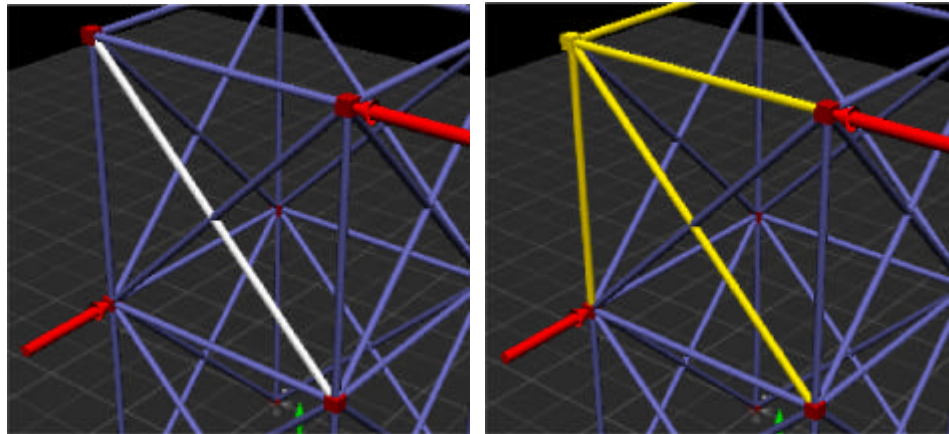


Figure 4 - Highlighting and selection

## VISUALISING STRUCTURE RESPONSE

To enable experimentation with the interaction of forces and structure, ObjectiveFrame implements a special feedback mode. In this mode a user-defined load can be applied to the structure. When the user moves the load, the corresponding deflections are computed in real-time. This enables the user to "feel" the response of a structure.

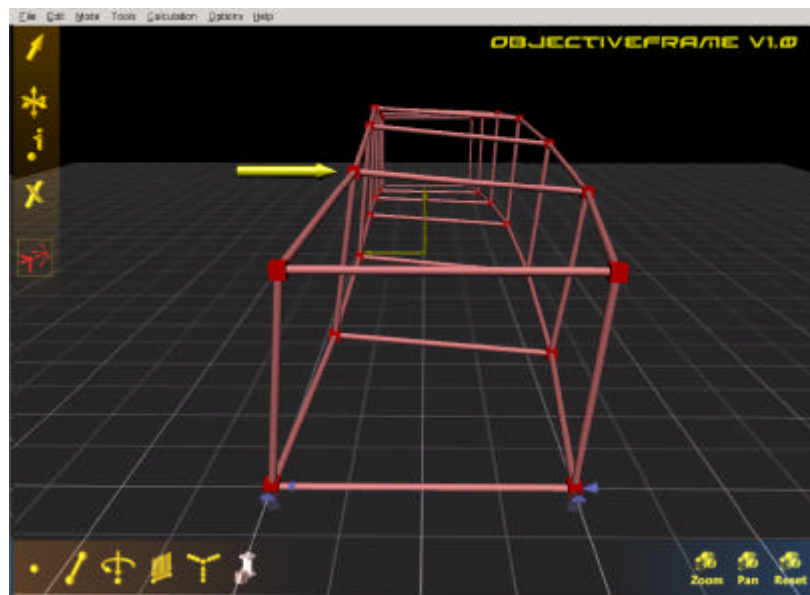
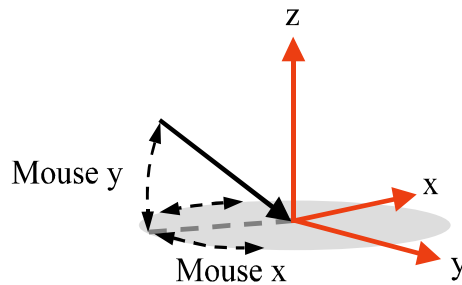


Figure 5 - Structure in feedback mode

When the feedback mode is selected, a node must be selected for the user-defined force. Moving the mouse over a node displays a force over it, indicating that it can be placed at the node. The force is attached to the node by clicking on it. A first calculation of the model is done as well. The force can now be moved using the mouse, updating the structure deflection continuously. Figure 6 shows how the mouse movements map to the 3d world. In a VR system the force could be controlled directly using some kind of 3d input device.



**Figure 6 - Force movement**

## EXAMPLE OF USE

ObjectiveFrame has successfully been used in the course "High Structures - A Creative Investigation" [4] at the School of Architecture at Chalmers University of Technology. In the course the students investigated the architectural possibilities and limitations of high structures. ObjectiveFrame was used as a tool for understanding the effects of different methods for stiffening a structure. Using the feedback method, the students could experiment and evaluate the different methods in real-time on the screen.



**Figure 7 - A student group working on the assignment**

## IMPLEMENTATION

The application was implemented in C++. The 2d user interface parts were implemented using the platform independent toolkit FLTK [8]. The 3d user interface was implemented using the Interactive Visualisation Framework - Ivf++ [5], which is a set of C++ libraries for creating 3d graphics. This library uses OpenGL [6] as a rasterisation interface.

## CONCLUSIONS

It has been shown that a direct manipulation interface can be effectively used in 3d beam analysis. Instant feedback of actions gives the user the impression of directly manipulating the beam model. By extending the direct manipulation paradigm to the post-processing and visualising, enhances this experience even further.

Students have successfully used the concepts introduced in ObjectiveFrame as a "Virtual Workshop" in architectural education as well as in design science education.

## REFERENCES

- [1] P. Christiansson, Knowledge communication in the global network, Position paper for the July 16-20 1995 Workshop on Research Directions in Architectural Computing, Published as a chapter in a book from KLUWER in June 1996
- [2] A. Dix et al, Human-Computer Interaction, Prentice Hall International (UK), 1993
- [3] Half-life, <http://sierrastudios.com/games/half-life>, 2000
- [4] High Structures - A Creative Investigation (Swedish), [http://www.arch.chalmers.se/projekt/high\\_structures/index2.html](http://www.arch.chalmers.se/projekt/high_structures/index2.html), 2000
- [5] Lindemann J., Interactive Visualisation Framework - User's guide, Report TVSM-3038, Division of Structural Mechanics, Lund University, 2000
- [6] OpenGL, <http://www.opengl.org>, 2000
- [7] J. Preece, Human-Computer Interaction, Addison-Wesley Publishing Company, 1994
- [8] B. Spitzak, Fast Light Toolkit FLTK, <http://www.fltk.org>, 2000