THESIS FOR THE DEGREE OF DOCTOR OF PHILOSPHY

# Conceptual Studies in Structural Design

## pointSketch – a computer-based approach for use in early stages of the architectural design process

PIERRE OLSSON

Department of Architecture

CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2006

**Conceptual Studies in Structural Design**
pointSketch – a computer-based approach for use in early stages of the architectural design process

PIERRE OLSSON


ISBN 91-7291-753-9

Doktorsavhandlingar vid Chalmers Tekniska Högskola
Ny serie nr 2435
ISSN 0346-718X

ISSN 1650-6340, 2006:01
Publikation – Chalmers Tekniska Högskola
Institutionen för Arkitektur

Department of Architecture
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31 772 10 00
www.chalmers.se/arch

Pictures and figures by the author if no other reference is given.

Majornas Copyprint
Göteborg, Sweden 2006

# ABSTRACT

Pierre Olsson, *Conceptual Studies in Structural Design: pointSketch – a computer-based approach for use in early stages of the architectural design process.* Department of Architecture, Chalmers University of Technology, Göteborg, 2006.

The development of computer hardware and software has progressed rapidly in recent years. Nevertheless, the development of software concerned with load-carrying structures and intended for the architectural design process has lagged behind, the majority of it being developed by and for engineers, often making the architects who are involved unduly forced to rely on the know-how of the engineer.

Two case studies in the context of furniture design were carried out to investigate the requirements that a computer-based design aid intended to support collaboration between the architect and the engineer at early stages of the design process should fulfil in order to be of genuine help.

Computer tools based on the Finite Element Method (FEM) were used for performing structural analyses in both studies. Observations made in the case studies enabled a set of characteristics to be suggested that it would be desirable for such a computer-based design aid to possess. These characteristics made it possible to propose a basic approach referred to as pointSketch, within the framework of which the characteristics in question were developed further, were organised, and were described in greater detail. The pointSketch approach was put into concrete form through the development of two computer programs, pointSketch2D and pointSketch3D.

The programs were evaluated in three separate steps involving both small informal groups and a workshop attended by participants familiar with the collaboration between architects and engineers. Throughout the evaluation process the pointSketch approach was well received. The opportunity users are given to experiment freely with structural behaviour was appreciated in particular, both in a professional and an educational context. The pointSketch approach and the two programs associated with it show how structural mechanics can act as a design parameter on equal footing with traditional design parameters and can lead to the creation of new and innovative structures.



**Keywords**: Structural Mechanics, FEM, Scientific Visualisation, Conceptual Design, Architectural Design Process, Architecture, GUI, Canonical Stiffness, pointSketch, Furniture Design

# Contents


**1 INTRODUCTION ........ 1**

1.1 BACKGROUND ........ 1
1.2 AIMS OF THE THESIS ........ 3
1.3 METHOD ........ 3
1.4 STRUCTURE OF THE THESIS ........ 3

**2 FURNITURE IDS ........ 5**

2.1 SUMMARY OF THE PAPERS ........ 8
*2.1.1 Workshop – redesign of an armchair ........ 8*
*2.1.2 Case study – furniture design ........ 14*
2.2 DISCUSSION ........ 24
*2.2.1 Working with sketches ........ 24*
*2.2.2 Visualising cause and effect ........ 25*
*2.2.3 Different levels for different users ........ 26*
*2.2.4 Analysis of canonical stiffness ........ 26*
*2.2.5 The CAD and CAE tools ........ 27*
*2.2.6 The object of design ........ 27*

**3 STATE OF THE ART ........ 29**

3.1 COMPUTER-BASED TOOLS ........ 31
*3.1.1 CAE tools ........ 31*
*3.1.2 Real-time tools for Structural analysis ........ 33*
*3.1.3 Computer games ........ 36*
3.2 BASIC PATTERNS IN STRUCTURAL MECHANICS ........ 38

**4 DEFINING A CONCEPT – POINTSKETCH ........ 45**

4.1 VISUALISATION MODES ........ 46
*4.1.1 Sketch mode ........ 47*
*4.1.2 Appearance mode ........ 47*
*4.1.3 Physics mode ........ 48*
*4.1.4 Action mode ........ 51*
*4.1.5 Graphical settings ........ 52*
*4.1.6 Switching between modes ........ 53*
*4.1.7 Link to CALFEM ........ 53*

4.2 PRECISION LEVELS ..... 54
4.3 SKETCHING ..... 56
4.4 ANALYSIS OF CANONICAL STIFFNESS ..... 58

**5 TOOLS AND METHODS FOR DEVELOPMENT OF THE APPLICATIONS ..... 61**

5.1 PROGRAMMING LANGUAGE AND SOFTWARE DEVELOPMENT TOOLS ..... 61
5.2 EXTERNAL PROGRAMMING LIBRARIES ..... 62
5.3 METHODS FOR INTERFACE DESIGN ..... 63

**6 POINTSKETCH2D AND POINTSKETCH3D ..... 65**

6.1 STRUCTURE OF THE GUI ..... 66
6.2 THE MENU BAR ..... 68
6.3 THE ACTION-AND-TOOL BAR ..... 69
6.4 THE VIRTUAL CANVAS ..... 71
6.5 THE MESSAGE PROMPT ..... 71
6.6 PHYSICS MODE - VISUAL REPRESENTATION OF THE 'STRUCTURE' ..... 72
*6.6.1 Nodes ..... 72*
*6.6.2 Bars ..... 73*
*6.6.3 Supports ..... 74*
*6.6.4 Forces ..... 75*
6.7 ACTION MODE - VISUAL REPRESENTATION OF STRUCTURAL BEHAVIOUR ..... 76
*6.7.1 Forces ..... 76*
*6.7.2 Deformations ..... 78*
*6.7.3 Canonical stiffness ..... 78*
6.8 APPLICATION STRUCTURE AND PROGRAMMING ..... 80

**7 EVALUATION ..... 83**

7.1 SELF EVALUATION ..... 83
7.2 TEST PANEL ..... 84
7.3 WORKSHOP ..... 86
7.4 SHAREWARE ..... 91

**8 DISCUSSION ..... 93**

8.1 COMMENTS FROM THE EVALUATIONS ..... 94
8.2 CONTRIBUTIONS TO THE RESEARCH FIELD AND FUTURE WORK ..... 98

**REFERENCES ..... 101**

**APPENDICES** .......... **105**

I. pointSketch2D version 1.0

II. pointSketch3D BETA version

III. Olsson P., Eriksson P. and Olsson K.-G. *Computer-supported Furniture Design at an Early Conceptual Stage*
Also available on the web:
http://www.arch.usyd.edu.au/kcdc/journal/vol7/papers/olsson/index.htm

IV. Olsson P. and Olsson K.-G. *Applied Visualisation of Structural Behaviour in Furniture Design*
Also available on the web:
http://www.arch.chalmers.se/staff/pierre.olsson/dissertation/applied_vis.pdf

# Acknowledgments

The work presented in this PhD thesis was carried out at the Department of Architecture at Chalmers University of Technology. Financial support for it was provided by the faculty, Innovative Design (an organization concerned with design at Chalmers), and Chalmers Medialab.

During the nearly six years that work on the thesis has been underway, a substantial number of people have played an important role in its completion. First, I am very much indebted to my two supervisors, Senior Lecturer Karl-Gunnar Olsson and Professor Ulf Janson. Karl-Gunnar aroused my interest in structural mechanics and FEM during my time as a student at Lunds Institute of Technology (LTH) and later gave me the opportunity to develop this interest further as a PhD student at Chalmers under his supervision. His encouragement and his devotion to the academic interests we share has made the work highly interesting, challenging, and rewarding. I am grateful to Ulf for all the good advice he has given me and all the fruitful discussions we have had concerning academic writing.

I would also like to thank Dr. Jonas Lindemann at LTH for our discussions on the development of the pointSketch programs. Thanks are also due to Dr. Monika Billger and Senior Lecturer Catharina Dyrssen for their help and advice in the structuring of the thesis. Thanks also to Robert Goldsmith for revising my English; any mistakes that remain are entirely mine. I would like to thank Per Eriksson for the many discussions we had concerning the role of the architect in the design process. Very special thanks to all my colleagues at the Department of Architecture, particularly to Dr. Ylva Sandin and MSc Carl Thelin for our sharing of an office and for their being available whenever I needed to discuss matters connected with the thesis or other matters of general concern. Finally, I want to thank my family and my friends for the support they have given me during my time here at Chalmers.

Göteborg, February 2006.

Pierre Olsson

# 1 Introduction

## 1.1 Background

We are living at a time when the development of technological aids for architectural and engineering work is progressing at an astounding pace. Increasingly advanced computer programs for computations, modelling and visualisation have become available for architects and engineers at the same time as the computational capacity and graphical performance of computers has been vastly improved. The development of hardware, i.e. both of computer components and of peripheral equipment, has been a prerequisite for such advances. Today, one can carry out advanced computations for structural mechanics on an ordinary laptop. High-performance graphics cards have become standard equipment in most personal computers, something that no more than 10 years ago only super computers had. These technological developments have made it possible to use one's own personal computer to perform computations and visualisations that previously demanded use of advanced computers systems, which in turn required specialised knowledge to manage. Also, the amount of the time computations require has decreased immensely as new hardware has been developed. It is now possible to carry out rather advanced computations almost in real time. This means that the execution of the computations themselves does not have to be planned far ahead. One can be lavish in one's use of computations, carrying out series of fast tests created on the spur of the moment, with no need of relinquishing one's desires in this respect because of technological limitations. In the context of architecture and design the developments that have taken place have opened up new possibilities for use of computation and visualisation. At early stages in the design process, the architect and the engineer can investigate the effects that the shape and the inner structure of buildings have on the distribution of forces and on deformations. Architectural and engineering means of investigating and defining shape and geometry such as sketching, rapid prototyping, rendering and solid modelling can now be supplemented by computerised tools for investigating the interplay between shape, inner structure and forces.

Although the development both of hardware and of software (i.e. of computer programs) has progressed rapidly, software development has lagged behind in some areas of professional interest here. Software related to structural mechanics, for example, has tended to be designed only for those with a high level of knowledge in the area. Similarly, computer programs designed for the analysis of the technical systems in a building have primarily been developed simply by and for engineers. The designer/architect tends to be neglected in this respect and to be forced to rely unduly on the know-how of the engineer. The design process becomes divided up into a designer/architect and an engineering part instead of a collaborative process being established. Computer programs that utilise the possibilities the new technology provides and at the same time are so designed that the information they provide is readily accessible to those outside the engineering area are rare. This is virgin territory ready to be explored and investigated.

I have entered this new research area with the background of being an engineer with the two directions of structural mechanics and computer science and with a position as a PhD-student in an environment of architecture and design research. An appropriate opportunity for such research appeared in the early spring of 2000 when the project Innovative Design was initiated at Chalmers University of Technology. The aim of this project was to integrate design, science and engineering into a single field. A sub project, Furniture IDS (Integrated Design Studies), was carried out within this framework of this overall project. It was concerned with the development of design methods applicable to the integrated collaboration of experts in design, materials science, mechanics of materials and manufacturing. My participation in the project and my development of computer software aimed at facilitating the collaborative design dialogue that took place represented the staring point of my thesis. The desire for a (general) tool for qualitative studies being developed had been expressed, one having the ability to switch rapidly between the appearance of a structure and its behaviour, and to rapidly perform virtual experiments by changing the shape and the material properties of a structure, promoting generative knowledge in this way. The designing of software and its graphical user interface (GUI) to provide a research methodology was well suited to an architectural research tradition of using design experiments and creative work of different sorts as elements of the research methodology employed, one used in research concerned with the interplay between man and material.

## 1.2 Aims of the thesis

The thesis concerns the interaction of architects and designers with engineers, the aims being are twofold:

- To investigate and exemplify the requirements to be met by a computer-based design aid intended for the early stages of the design process.
- To propose how such an aid could be designed.

## 1.3 Method

In the first part of the thesis, two case studies that were carried out are presented involving efforts to analyse design processes in which computer-based design aids were employed. Observations made in the case studies are compared with the results of research reported in the literature in the attempt to achieve a state-of-the-art understanding of the area. On this basis, the requirements a design tool should meet could be specified, this becoming the basis for the development of two computer programs, development tools for programming in combination with appropriate object libraries being used to create them. The programs were evaluated in four separate steps involving both small informal groups and a workshop attended by participants familiar with the collaboration between architects and engineers.

## 1.4 Structure of the thesis

The thesis is divided into two parts. The first part (chapter 2) deals with the matters just referred to and also contains a summary of my licentiate thesis.[1] The chapter begins with a report on the current use of computer-based aids in different stages of conceptual design in the Swedish furniture industry. This is followed by a presentation of two furniture projects carried out at Chalmers in collaboration with designers and manufacturers invited to attend. The outcome of the furniture projects is discussed and a set of requirements for a computer-based aid is formulated (chapter 2).

The second part of the thesis begins with a presentation of the state of the art in which related research and available design aids are analysed (chapter 3). This is followed by the account of a proposed concept of such a design aid (chapter 4). Next, development tools and methods utilised in creating computer programs based on the proposed concept are described (chapter 5). The account of two computer-based programs, pointSketch2D and pointSketch3D, with

[1] Olsson 2003

accompanying user interfaces, is presented (chapter 6). An evaluation of the two programs (chapter 7) and a discussion of them (chapter 8) concludes this part of the thesis.

# 2 Furniture IDS

Both the development and the construction of furniture tend to be largely based on experience rather than on scientific investigation, designs being strongly influenced by cultural traditions, conditions in the furniture industry and current trends and fashions. Few designers of furniture have furniture design as a full-time occupation. Many of them are architects, interior designers or industrial designers. During the design process, there are different ways in which a furniture designer can collaborate with those concerned with producing furniture. One way is to work in a furniture design office and collaborate closely with a furniture producer. Another way is to not simply design furniture but also to produce it. Many large-scale producers have their own designers. Designing of this sort is often almost completely oriented to the prerequisites of the industry, such as specific tools, machines and processes. Most furniture design follows a particular pattern, although this can vary slightly in terms of the designer involved and of the dialogue that takes place between the designer and producer.

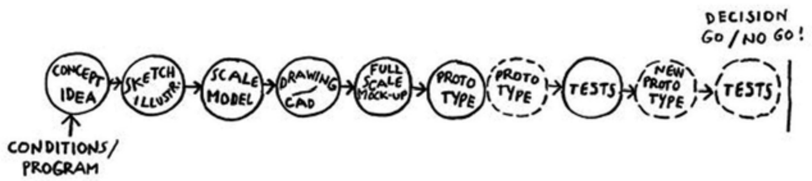


**Figure 1** A simplified model of a traditional furniture design process. (Source: Olle Anderson)

To satisfy the demands of users regarding safety, strength, comfort and accuracy, various recommendations and codes have been developed for the testing of furniture. The results of tests are often simply a yes or a no in regard to different testing categories. There is seldom room for any appreciable discussion of why a particular piece of furniture failed to pass a given test. Accordingly, there is a lack of feedback. Also, when testing is performed it is usually on full-scale prototypes or on test series. The testing of components or of alternative solutions early in a design process occurs very seldom.

For some years now, the *Studio Materials and Design* at Chalmers University of Technology has been a place for investigating new work procedures and methods concerned with collaborative and integrated design processes. Some of the projects carried out at the Studio have aimed at incorporating new methods and tools into the furniture design process. The collaborative design group has combined competencies from the areas of furniture design, structural mechanics, marketing, physical testing and production. Special attention has been directed at utilising competence in structural mechanics in the design process. The reason for using the furniture industry as an area of application was not to stimulate that industry as such, despite its being conceivable enough that this might occur. Rather, furniture design involves a limited but complex design task involving the interplay between artistic values structural design, making it an excellent environment for research on collaborative design generally.

The furniture projects carried out have provided a unique opportunity for investigating how structural mechanics, visualisation and graphical user interfaces can be utilised for facilitating the work of designers and engineers in the early stages of the design process, in accordance with the aims of the present thesis. Although these projects have been specifically directed at the furniture industry, they have concerned in a more general sense the collaboration between artistic and engineering creativity, which is the kind of situation in which these tools have particular potential as design aids. As a help in keeping my work directly oriented to the overreaching aim of the thesis while being concerned in a more concrete way with the furniture projects I was engaged in, there was a more general objective I adhered to. Keeping this objective in mind was a way of integrating the aims of the thesis with the aims pursued in the furniture projects:

> Pictures, models and movies created in the post-processing part of a computer aided engineering tool (CAE) are good examples of what is termed *scientific visualisation*, i.e. the process of creating images from data in order to assist comprehension. Scientific visualisation and presentation graphics differ in the sense of the latter being concerned primarily with the communication of information and results that are already understood, whereas the former concerns efforts to make the data as easily understood as possible.

*The objective of these studies is to formulate, test and evaluate how FEM[2]-based tools, supported by scientific visualisation and graphical user interfaces, can be used as a means of creating an integrated[3] process for furniture design, and also to show how these tools can provide a better understanding of the problem at hand and enhance possibilities for the development and evaluation of new designs. Making these tools available can also help those involved to better understand new and important dimensions of their respective professions. In addition, it facilitates the exchange of knowledge between professions so as to make it easier to create better and more adequate designs through collaboration.*

There is often a lack of understanding between the designer and the engineer, the artistic and the technical work, respectively, of the two differing so much. Since the information a computer simulation of mechanical behaviour can provide is of interest to both professional groups, it can be used as a common basis for discussions between them concerning matters of design. Scientific visualisation can be used here to create a common language for facilitating professional dialogue. Case studies can help to demonstrate how FEM computation and scientific visualisation can be integrated into the furniture design process as design aids.

One of the case studies conducted at the Materials and Design Studio was entitled "Half the weight twice the strength!".[4] It concerned the redesigning of an existing armchair design. The deeper aim was to investigate how collaborative and integrated design processes could help improve methods of furniture design generally. This case study is presented and discussed in the paper *Applied Visualisation of Structural Behaviour in Furniture Design* which is appended.[5]

Due to the success of this first project, a second project concerned with working and design methods was likewise carried out. One of the aims here was to develop integrated design methods enabling people with special competence in the areas of furniture design and structural mechanics, respectively, to collaborate effectively. Another aim was to design IT-tools and develop them further to the extent needed, as an aid to these new design methods. It was decided that to discover such tools a fictitious design process would be staged in which a number of relevant IT-tools would be utilised. This again provided the possibility of using

[2] The Finite Element Method (Ottosen and Petersen 1992) is the most common numerical method used for computations within structural mechanics. (Note added by author)

[3] In this case *integrated* means that the elements of a design process that ordinarily are performed separately are performed simultaneously instead. (Note added by author)

[4] Anderson 2003

[5] Olsson and Olsson 2003

FEM and scientific visualisation, both of which are clearly IT-tools, to aid the design process. This case study is presented and discussed in the appended paper *Computer-supported Furniture Design at an Early Conceptual Stage.* [6]

## 2.1 Summary of the papers

### 2.1.1 Workshop – redesign of an armchair

The participants in this workshop represented many different areas of expertise, such as furniture design, architecture, structural mechanics, furniture manufacturing and furniture testing. The object of design was an armchair designated as KS263.

**Figure 2** Armchair KS 263.

It had turned out that after extended periods of use in public environments, such as at conferences, such chairs had been found to be rickety. A visual inspection indicated that the glued joints between the wooden components of the chair had

[6] Olsson, Eriksson and Olsson 2004

tended to fracture because of the frequent lifting and stacking they had been subjected to.

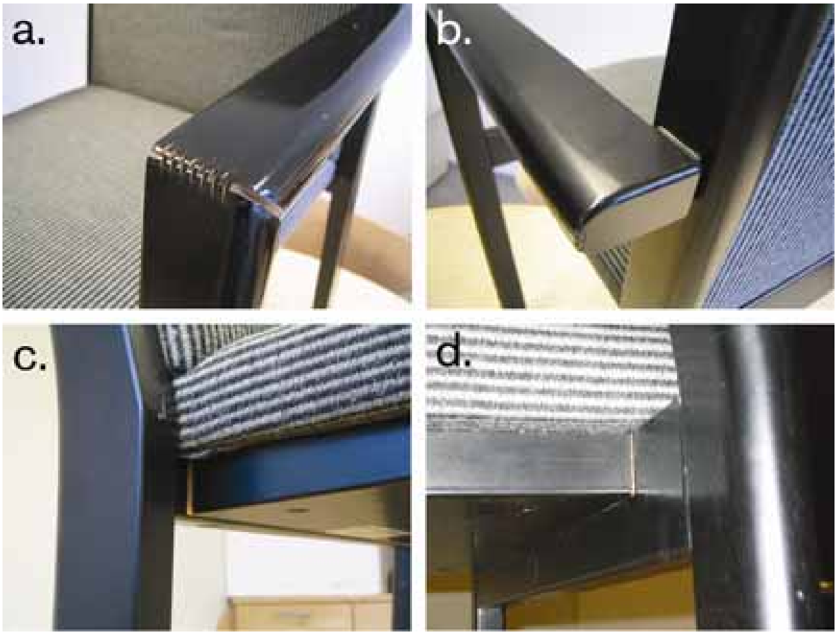


**Figure 3** The fractured joints in the chair, showing a. the joint between the arm rest and the front leg, b. the joint between the arm rest and the back leg, c. the joint between the seat and the back leg, and d. the joint between the seat and the front leg.

The overall goal of the workshop was to make the chair both lighter and stronger. The phrase "Half the weight twice the strength!" was coined to signify the workshops intension. The first objective aimed at was to explain how and why the different fractures had occurred, the second objective being to suggest a more favourable design for the chair.

In order to improve the chair it was first important to understand how the chair had broken down. The workshop asked for a tool that could easily simulate and explain this process and visualise the results for the group so as to facilitate further discussions. Even before the workshop began it became clear that a small and fast tool that could be interacted with in real time was needed so as to be able to maintain a continuous dialogue in the workshop without interruptions. Since no appropriate tools were available for purchase, a custom-made computer program for designing purposes was constructed, one designated as chairSketch.

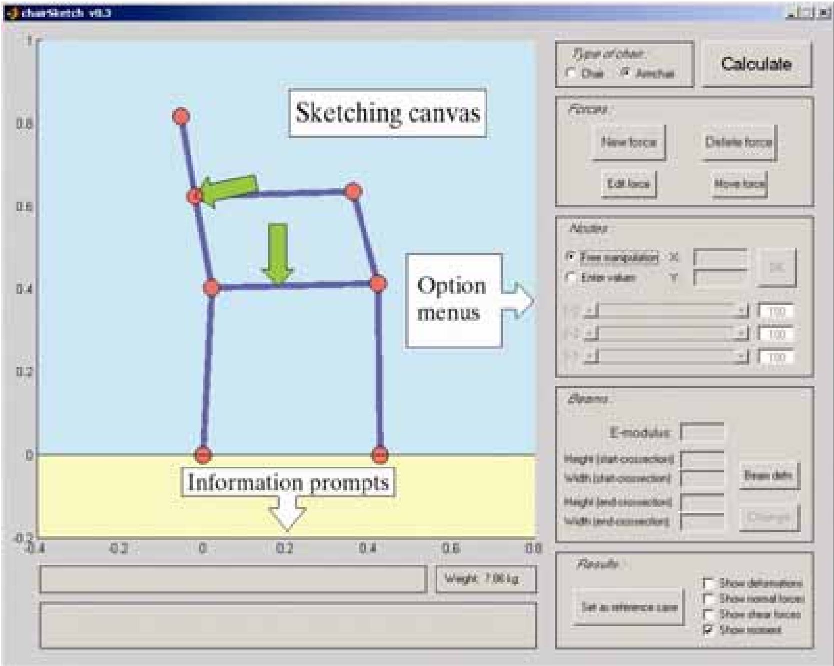


**Figure 4** The chairSketch GUI.

The first step in creating this design tool was to determine the important design parameters for the chair. Since most concerns applied to the joints between the chair's components, it was decided that the joints should be the primary design parameter in investigating structural behaviour here. The basis of the tool was a computational model created with a fixed number of beams connected to each other by use of translational and rotational springs, making it possible to consider different level of stiffness in the joints. On top of this computational model a GUI was created which allowed the design parameters connected with the chair to be modified. An important aspect of the GUI was to make the two-dimensional model of the chair as easy as possible to change and evaluate on the basis of suggestions made by the participants in the workshop. The computer tool was designed to act as a catalyst in the discussions of a professional character carried on within the workshop.

The first question addressed in the workshop concerned the manner in which the chair had broken down. The most visible fracture was located at the joint between the front leg and the armrest, which led many to believe that this was the weak spot of the chair and should thus be strengthened, see Figure 3a. Here chairSketch was used for the first time, for the purpose of explaining how the different fractures had occurred. A series of computations were carried out in which the joints were modelled differently as the successive failure of the glued joints progressed. The load case was that of a person sitting in the chair and thus applying forces perpendicular to the seat and to the backrest, as shown in Figure 4.

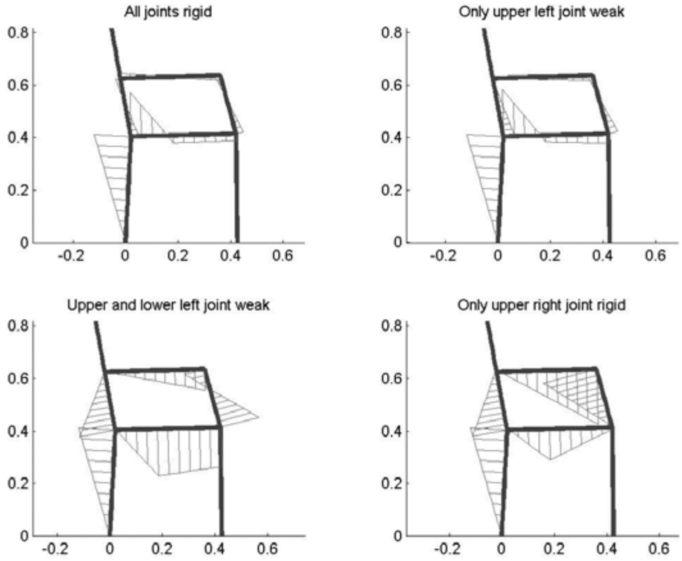


**Figure 5** Moment diagrams showing how damage starts and spreads through the chair.

In the first analysis (the diagram at the upper left in Figure 5), all the joints were modelled as being intact so as to show the initial structural behaviour of the chair. A quick inspection of the joint between the armrest and the rear leg was enough to indicate that this joint would never be able to carry any appreciable load compared with the other joints, due to the limited size of the area of contact between the adjacent surfaces of the two chair components involved, see Figure 3b. Accordingly, this joint was modelled as being free to rotate (the diagram at the upper right in Figure 5). The new distribution of moments indicates the joint between the seat and backrest to carry the major part of the load. This was surprising to the participants, who imagined the most visible joint (between the front leg and the armrest) to be the one carrying most of the load.

The next analysis involved imagining the joint between the back leg and the seat to be fractured, i.e. its not being able to carry any rotational load, its thus also being modelled as free to rotate. At this point, the only joints able to carry loads were the joints in the front of the chair, as indicated by the diagram at the lower left in Figure 5. The lower of the two remaining joints appears at that point to be carrying most of the load and therefore to likely be the next joint to break. At this point the only joint able to carry any load is the joint between the front leg and the armrest, as indicated by the diagram at the lower right in Figure 5. The most visible fracture turned out to be the last one to occur, when the chair broke down. This led to the participants who suggested improving this joint reconsidering their recommendation. Instead, putting efforts into making the joint between the back leg and the seat stronger seemed a better idea since it would ensure that the other joints would never be subjected to a load they were obviously unable to support.

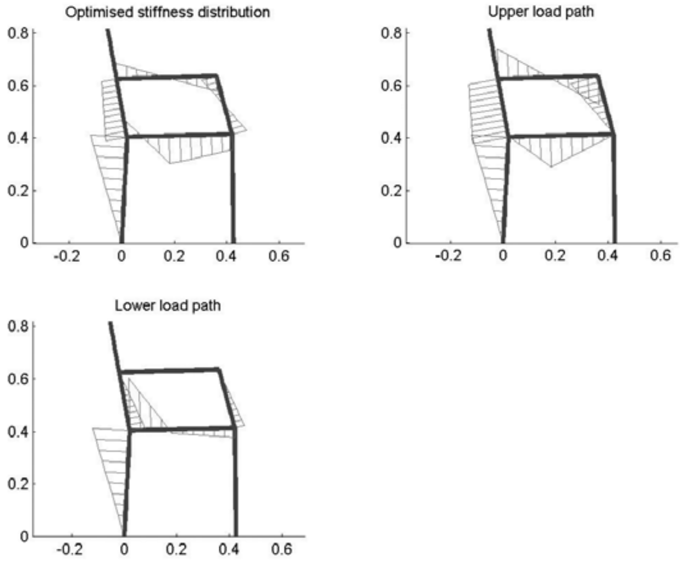


**Figure 6** Moment diagrams resulting from different distributions of rotational stiffness in the joints.

After these initial analyses, the participants realised the potential in discussing different design ideas by using chairSketch as a common language. The creative impulses this engendered led to three other suggestions for improving the chair being made. One was to spread out the stiffness evenly across the joints so as to enable them to carry the load together instead of one by one (the diagram at the upper left in Figure 6). Having all the joints equally strong also implies that a similar design can be used for each joint, giving the chair a uniform look, but it also requires accuracy in designing the stiffness of the joints.

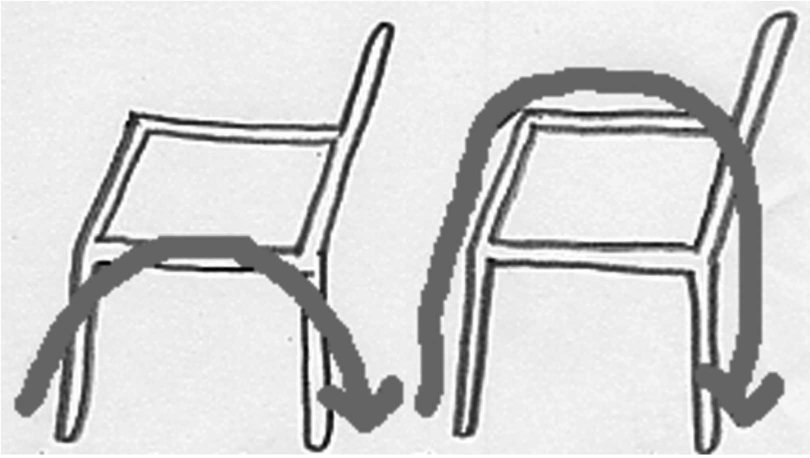

**Figure 7** Additional suggestions, that of having an upper load path and that of having an lower load path. (Source: Olle Anderson)

Another idea was for the load to be carried entirely by the upper joints (the diagram at the upper right in Figure 6), enabling the seat to have a slim design, which would also improve the stackability of the chair. The opposite would be for the load to be carried entirely by the lower joints (the diagram at the lower left in Figure 6). According to the manufactures who participated in the workshop, it is very common to use a design of an ordinary chair and to simply add an armrest to make an armchair. Consequently, the structural systems of most armchairs are designed with strong lower joints in mind.

At the end of the workshop, many of the participants felt that they had not only managed to create a better chair, but more significantly that they had obtained new perspectives on how computer-based tools can enhance a design process. chairSketch had obviously also been very useful to them since the changes the participants suggested could be tested there on the spot, rather than its being necessary to manufacture a prototype to use in full-scale testing. The analysis made by use of chairSketch can be regarded as rather crude since the chair was simplified to a two-dimensional model, but the program was nevertheless able to answer the questions the participants posed. A more refined analysis could have provided more detailed answers, but none of the participants expressed the need for such an analysis in the course of the workshop. It was agreed that the level of complexity of the analyses that chairSketch provided was quite sufficient at an early stage of the design process. On the whole, the comments made in the workshop concerning chairSketch and its role was very positive.

For further details, see *Applied Visualisation of Structural Mechanics in Furniture Design* (appended CD).

### 2.1.2 Case study – furniture design

The approach of the second study took differed from that of the first, especially in terms of who participated and what tools were used. This second study was more similar to a traditional case study, due to the fact that a tangible design process was carried out rather than a brainstorming session. This case study had only two participants: an architect and an engineer. The design process aimed at creating a chair for public use, a chair with an organic shell-like look. The aim was to investigate how the normal tools of an architect and of an engineer can be modified and be used in new and inventive ways to facilitate a more collaborative design process. Thus, no custom-made design tools were employed, but rather only the tools that architects and engineers normally use.

The first step in the design process was to create an initial sketch of how the architect imagined that the chair would look. This sketch was created using solid-modelling software and then exported to a neutral file format to enable the engineer importing the geometry to computational software.

**Figure 8** Design 0.

At the next step, the engineer used this sketch as a basis for creating a computational model. If the geometry had to be modified in some way to fit the template of the computational program, the engineer and the architect discussed these alterations with each other before applying them, so as to minimize any distortion of the original design.

This was followed by the computations being executed and the results being visualised. Here the engineer suggested computing of the canonical stiffness and the accompanying deformation modes of the chair to evaluate the chair's structural behaviour.[7] It was decided that the first five eigenmodes (deformation patterns) and the canonical stiffness associated with each of them would be analysed. The stiffness values associated with the other eigenmodes were much higher, making the accompanying deformation patterns unlikely to occur.

[7] Olsson 2005

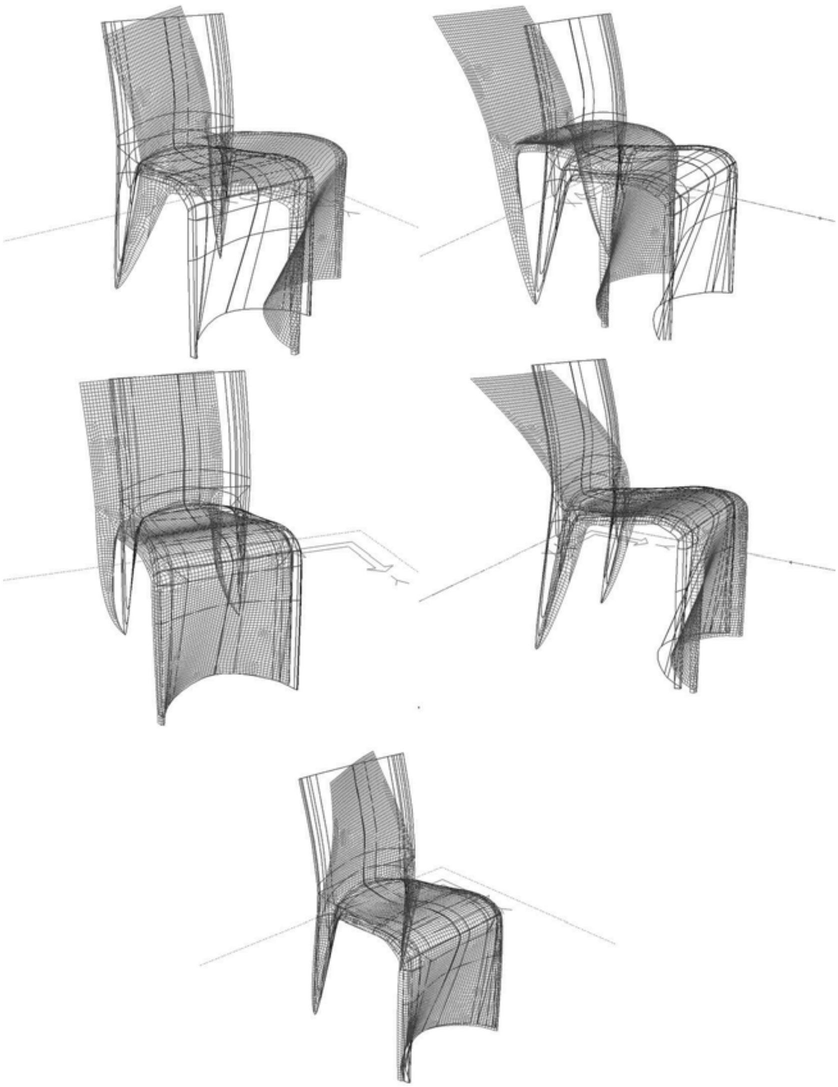

**Figure 9** The first five eigenmodes in design 0.

Pictures showing the distribution of stress were used to determine what parts of the chair were particularly strained when carrying a load that would result in the deformation patterns that had been computed. The results compiled were visualised the same way throughout the computations, so as to facilitate comparison of the different design ideas, see Figure 10 below (in which only stresses from the first eigenmode are shown).

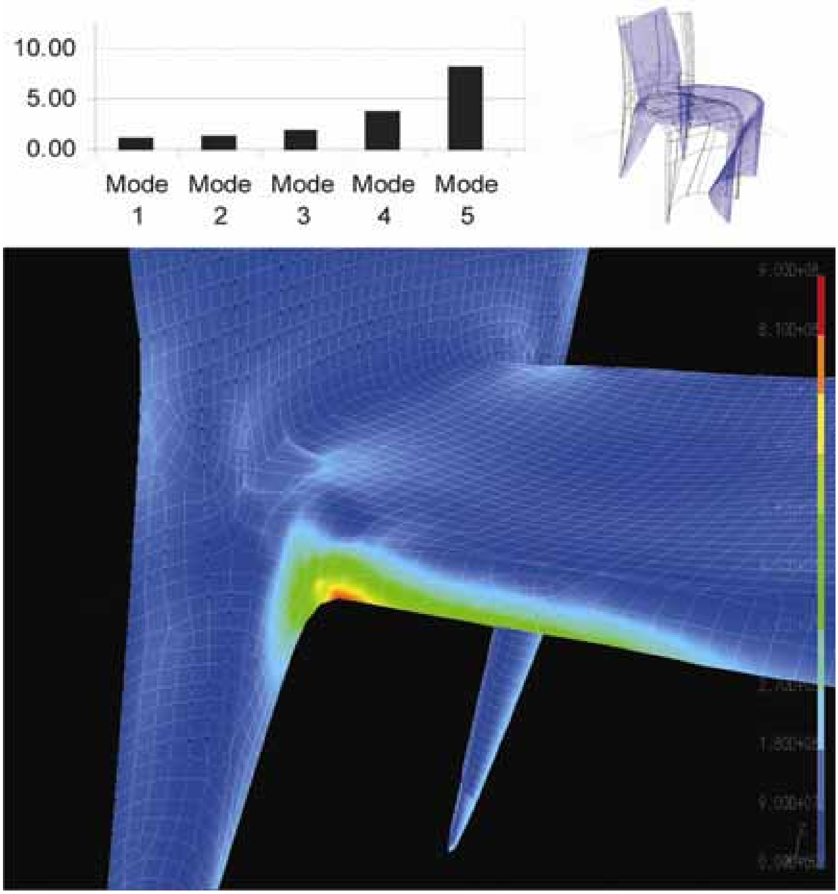


**Figure 10** The first five eigenvalues, the first eigenmode and the von Mises stresses associated with the first eigenmode in design 0.

Analysing the visualisation associated with the computations for design 0 indicated to both the designer and the engineer that a person leaning back in the chair while turning the to side represents the load case for which the chair is most sensitive. The other eigenmodes concern  of similar load cases in which the person sitting in the chair is leaning backwards and forwards or from side to side, which results in the connection between the seat and the back legs and backrest being highly strained, as shown in the visualisation of the stress distribution.

After discussing the results the designer and the engineer decided to try out designs in which this connection was changed to in an effort to increase the canonical stiffness of the first eigenmode (making the chair less sensitive to that particular load case) and to reduce the level of stress in that part of the chair (making the material less likely to be fractured). Three design alterations were suggested:

- Moving the rear legs farther back, design 1:1.
- Decreasing the curvature at the juncture between the rear legs and the backrest, design 1:2.
- Changing the profile of the rear legs and of the backrest, design 1:3.

**Figure 11** Designs 1:1, 1:2 and 1:3.

After the computations and the visualisations had been carried out for each of the three design alternatives, these alternatives could be evaluated by comparing them with the results for design 0.

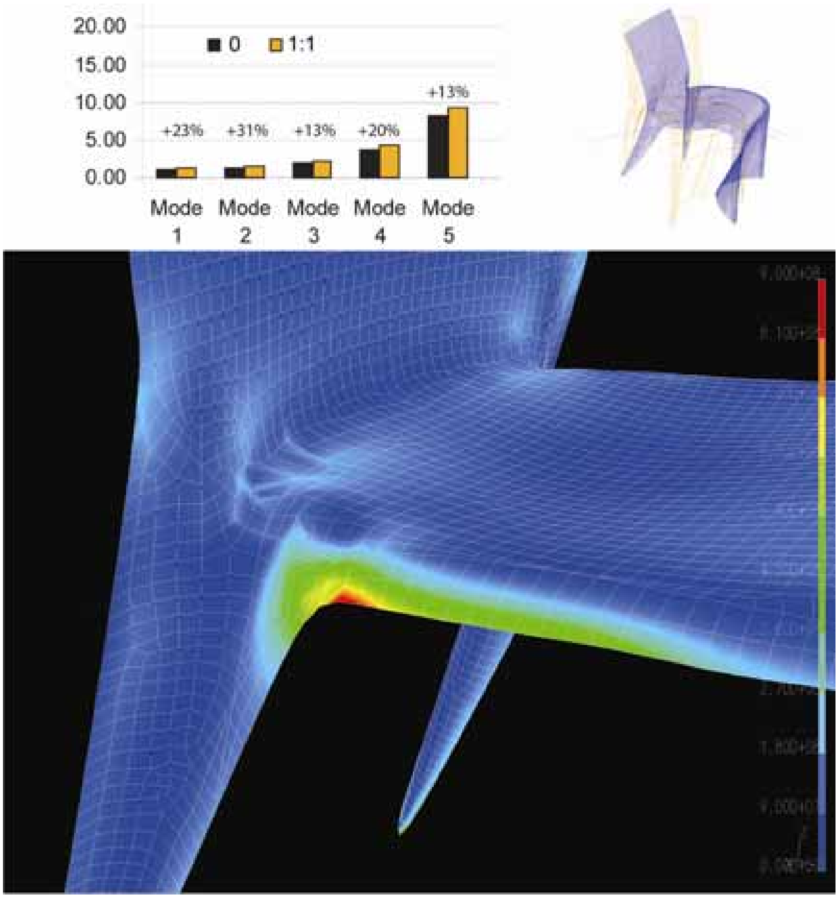


**Figure 12** The first five eigenvalues, the first eigenmode and the von Mises stresses associated with the first eigenmode in design 1:1.

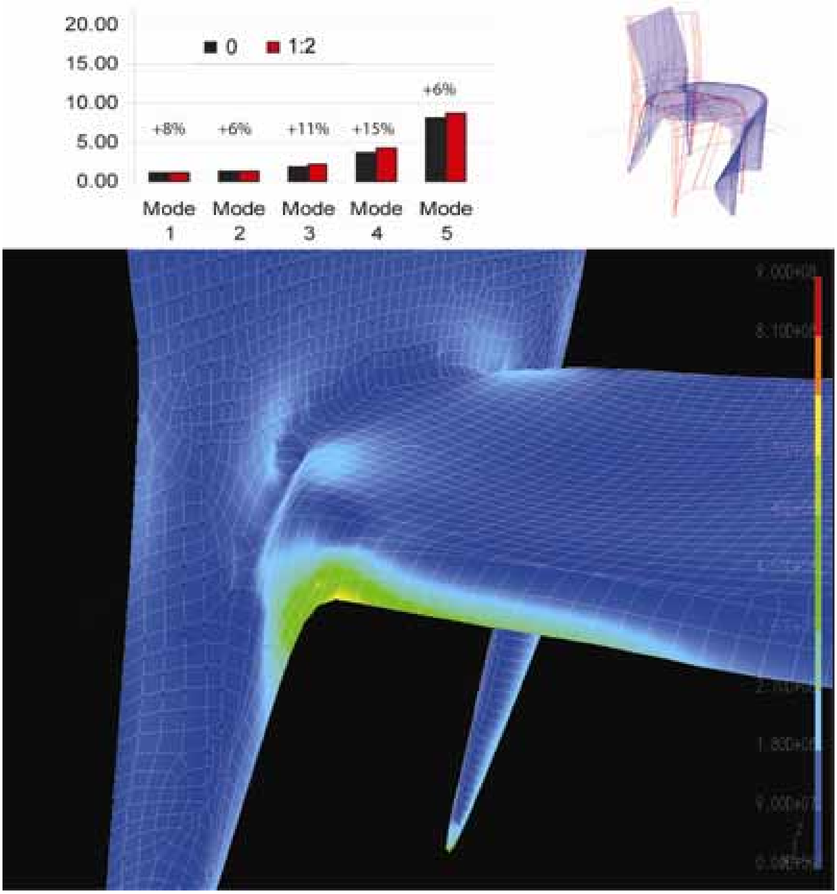


**Figure 13** The first five eigenvalues, the first eigenmode and the von Mises stresses associated with the first eigenmode in design 1:2.

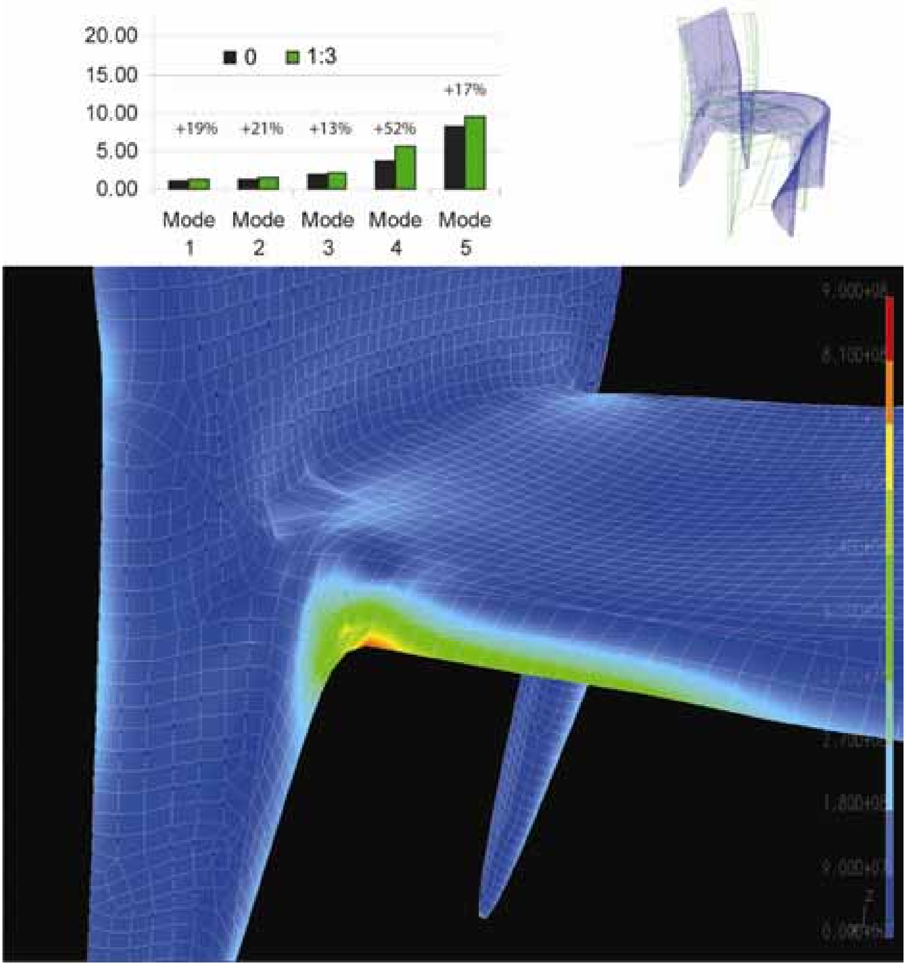


**Figure 14** The first five eigenvalues, the first eigenmode and the von Mises stresses associated with the first eigenmode in design 1:3.

By reviewing the canonical stiffnesses and the associated eigenmodes, the designer and the engineer learned that each of the three design suggestions had made the chair less sensitive to all five load cases, designs 1:1 and 1:3 being slightly better than design 1:2. For all three design suggestions the eigenmodes were ordered in the same way as for design 0, none of the design suggestions leading, therefore, to any change in the ranking of the different deformation patterns. Accordingly, the first eigenmode for design 1:1–3 is the same as for design 0, the chair still being most sensitive to a person’s leaning back and turning to the side. The visualisation of the stress distribution for the different design suggestions indicated all three of them to have improved the chair by lowering the high (red) values in the chair. In contrast to the results for canonical stiffness, design 1:2 was found to yield the most favourable stress distribution of all the designs.

On the basis of the visualisation of design 1:1–3 and the discussions concerning it, the designer created the next generation of this chair. This particular design was quite different in appearance from that of the others. The designer used sharp, well-defined angles here instead of smooth curved surfaces, endeavouring to see how this would affect the chair from an aesthetic point of view. He also wanted to determine how it would affect the chair in constructive terms. Although the shape of this chair was quite different, it still had the properties the designer had sought in the original chair.

**Figure 15** Design 2. Note that although the shape and the detailing are quite different here, the chair still possesses the inner properties the designer was aiming at in the original chair.

Next, computations concerning the new chair and visualisations of it were carried out in accordance with the previous designs.

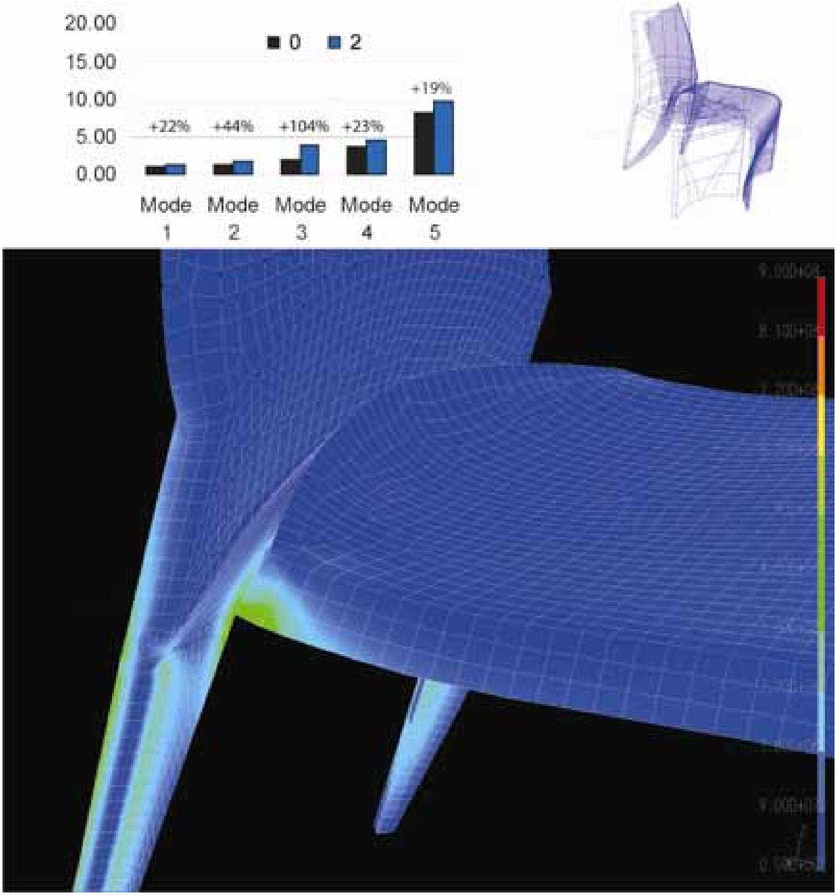


**Figure 16** The first five eigenvalues, the first eigenmode and the von Mises stresses associated with the first eigenmode in Design 2.

In terms of the eigenvalues obtained on the basis of the calculations, this new design was found to be stiffer, as was intended. This is especially true of eigenmode 3, for which more than twice as much strain energy is needed for a given deformation behaviour to occur. The von Mises plot of this design shows that the high level of stress is spread out here over a large area. Nevertheless, the stress levels are lower than the maximum stresses found in the earlier designs, which is generally preferable since it reduces the amount of material needed. Since the eigenvalues were increased and the von Mises plot also showed fewer high-level stress values, this design appeared to be an improvement over the earlier ones.

At the end of the design process both the architect and the engineer agreed that working closely with each other in the manner they did in the case study is far more efficient than communicating by means of the brief exchanges of information that are typical of how most engineers and architects collaborate. However, a good medium for communication is essential for collaboration of this type to function well. The medium involved can be a custom-made computer tool or simply the ordinary tools of each profession. Although a link between the CAE[8] tool of the engineer and the CAD[9] tool of the architect was created here, it still left a few details to be desired. For example, there were certain problems connected with using the appropriate neutral file format for the exchange of information on the geometry. Using the computer programs available to create a good medium for communication can work well, but its success depends heavily on the tools involved having sufficient possibilities for its functions to be used in ways other than originally intended by the developers. Many of the CAE tools failed to allow the user to edit each of the steps of the computational process, much of the functionality being hidden in a "black box" inaccessible to the mainstream user. When the tools available are closed to the user in this way, it is better to employ a custom-made tool so as to facilitate the collaborative design process in the best possible way. The downside is that it is very difficult to create the complex computational functions that would otherwise be available in a commercial CAE/CAD tool.

For further details see *Computer-supported Furniture Design at an Early Conceptual Stage* (appended CD).

## 2.2 Discussion

### 2.2.1 Working with sketches

Both case studies showed how use of structural mechanics, scientific visualisation and graphical user interfaces was able to improve the professional dialogue in the early stages of the design process. One thing in these studies that turned out to be very important was the process of using sketches instead of fully detailed models as the basis for computations. Evaluating a sketch, not only visually but also from the standpoint of structural behaviour, enables the designer to create a more complete design before leaving the sketching stage. A design that has been thoroughly worked through at the sketching stage is less likely to be problematical at later stages of the design process. Also, most architects want to

[8] Computer Aided Engineering
[9] Computer Aided Design

be free to change and test many different shapes and details in the early stages of the design process. Being forced to establish certain details of a chair this early in the design process in order to facilitate the computational process would impose too much on the creative work of the designer.

Working in the manner suggested in the case studies requires that the engineer will be capable of adjusting the computational process to fit the sketch-like model rather than relying on the sort of fully detailed model for which most computational programs are designed. The results computations based on a sketch-type input cannot yield as accurate results as those based on a fully detailed model. However, the studies in the furniture projects here clearly showed that the less detailed results obtained were more than sufficient to the answer the type of design questions that were asked during the early stages of the design process.

### 2.2.2 Visualising cause and effect

Visualising the interplay between cause and effect in the mechanical behaviour of a structure was an important element in both these studies. Being able to see how a change in the geometry of the chair directly affected its structural behaviour was something that all participants found very helpful in evaluating different design ideas. Those with extensive knowledge of structural mechanics also found the visualisations of cause and effect relations to be helpful for explaining such matters who were less knowledgeable of structural mechanics. The possibility to shift between a view of the physical appearance, a model of the physics involved, and the computed results more or less in real time enabled the professional discussion to be conducted with few interruptions. The participants in the furniture studies agreed that having computations performed alongside the discussions was essential for keeping the creative process running smoothly.

Experience in workmanship, materials, manufacturing and aesthetics is widespread in the furniture industry today, whereas experience in structural mechanics is not. The pictures created by the use of FEM and of scientific visualisation also gave the participants the opportunity to be informed regarding structural mechanical matters in the course of the design process. The visualisation based on the real time calculations contributed to a learning process for the participants. They became increasingly accurate in their guesses concerning the structural behaviour of the chair.

### 2.2.3 Different levels for different users

The chairSketch program used during the first workshop was designed to accommodate both novice users and more advanced users. By giving predetermined values to many of the necessary settings for the computations, a novice user could easily modify the model of the chair and run a computation without having to be familiar with the inner workings of the computational engine. By making it possible at any time to adjust these settings if so desired, more advanced users can modify the computations in accordance with an extensive understanding of structural mechanics. Having this functionality in chairSketch made it possible not only for the engineers, but also for the designers, architects and manufacturers who participated to use the program. The participants in the workshop could all use chairSketch as a tool to convey how they thought the chair should be modified.

The aim was to design a program to fit both a novice user and an advanced user without making it too difficult or too cumbersome for either of them to make use of it. A good example of this aspect of chairSketch is the computational engine. It was created in CALFEM, which is a script-based programming language. [10] This means that the computational model can be accessed and edited in accordance with the desires of the user. However, doing so requires the user having considerable knowledge of structural mechanics in general and of FEM in particular. For this reason, the possibility of accessing the CALFEM code that chairSketch generated was created as an optional function in order not to burden the novice user with matters concerned with the computational model.

Another example of catering to the need of different users concerns the sketching process. The length and the position of the chair's components are determined by the position of the connecting points. These points can be moved either by the mouse by use of the drag and drop function, or by exact coordinates being be entered for each connection point. These are two very different styles of modifying the chair that are both available in the program without their interfering with each other.

### 2.2.4 Analysis of canonical stiffness

Analysis of canonical stiffness was a central element in the second case study. It was also one of the main reasons for the design process being as successful as it was. The pictures and animations of the eigenmodes provided the designer and the engineer with indications of the overall effects of a particular shape, a

[10] CALFEM 2006

particular combination of materials or a particular set of support conditions, showing how the chair was most easily deformed. This can be interpreted as indicating what load cases the chair is most sensitive to, which is useful in a design tool since it ensues that no important load case will be overlooked in analysing the structural behaviour of the chair. Information regarding structural matters, when made available at this early stage of conceptual design, can be a guide and an inspiration in helping the designer and the engineer, through collaborating to transform their rough conceptions  into more concrete ideas regarding shape.

### 2.2.5 The CAD and CAE tools

In the case studies, both commercial and custom-made computer programs based on FEM and scientific visualisation were found to be useful tools. The choice of tools was based on the characteristics of the case in question. This was a choice that was not made in a general way in advance, but was made during the design process by those engaged in it, their doing so on the basis of their knowledge and experience. This was found to be better than to use some predetermined toolbox that the participants might or might not be familiar with. One particular problem encountered in the second case concerned the exchange of data between the CAD tool and the CAE tool. When a 3D model is imported, the CAE tool is very sensitive to the geometry being a coherent solid without gaps between the surfaces. Thus, the CAE tool has a lower tolerance level in this respect than the CAD tool. The sensitivity of this sort pertains to creating an FEM model, which requires a well-defined geometry. Several times in the second case when data was to be exchanged, the designer had to refine the CAD model, since it was not possible to import it into the CAE tool properly. Certain rules for creating geometries in the CAD tools were finally agreed upon by the participants to facilitate the problem-free importing of information into the CAE tool. These rules concerned mainly the omitting of large gaps between surfaces. The problems encountered during the exchange of data could have been avoided by choosing tools in terms of compatibility rather than of their familiarity to the users. However, it appeared better to solve compatibility issues when they were encountered rather than being hampered by use of unfamiliar tools during the design process.

### 2.2.6 The object of design

Both the case studies concerned the designing of chairs, a choice made consciously, since furniture of this kind was adjudged to illustrate the use of the proposed tools and methods in an appropriate manner, its representing a limited yet advanced form of construction. A pertinent question in this context was whether these tools and methods were better suited to designing some types of furniture than others. Participants in the case studies felt that these tools and methods were most applicable to design processes concerned with furniture of a type that could prove difficult to manufacture. The tacit knowledge of furniture producers was adjudged to be sufficient in most cases to solve problematical aspects of furniture production. Yet since this knowledge is primarily based on experience gained in the manufacturing of known types of furniture, it may not be sufficient for solving problems concerning new forms of expression in furniture design. Furniture expressing bold shapes or involving the use of new materials or the use of materials in new ways are examples of situations in which these tools can be expected to be particularly useful.

The furniture studies clearly showed the use of tools based on structural mechanics, scientific visualisation and graphical user interfaces to have considerable potential in supporting the early stages of the design process. The studies made it possible to identify certain *aspects* of a desirable design aid:

- *Sketching* – the tool's being able to support sketching as a common professional language.
- *Computation* – its being able to predict structural behaviour by use of a computational engine. In particular, the results of an analysis of canonical stiffness are very useful in the early stages of the design process.
- *Visualising* – its providing pictures facilitating an understanding of cause and effect relations in connection with structural behaviour.
- *Levels in knowledge* – its enabling the interface to be adjusted in terms of differing levels of knowledge of structural mechanics.
- *Tools* – its making use of established conventions for tools commonly employed by users in question.

The aspects of an appropriate design tool just referred to are based on case studies carried out in the context of furniture design. How well do they agree with results of current research on computer-based design aids intended for the early stages of the design process generally?

# 3 State of the art

Research and development concerned with computer-based tools for use in the early sketching stages of the architectural design process has been carried out with a variety of purposes in mind. One of the most active areas of research of this sort concerns virtual environments (VE). The aim is that virtual environments of proposed projects be created that are accurate and photorealistic in appearance, the visualisations thus produced often being referred to as *architectural walkthroughs*.

Research on audiovisual aspects of virtual environments is another growing area one that concerns tools, concepts and theories aimed both at creating and improving upon such virtual environments. Examples of this are research on virtual light rendering, on representation of colour in virtual environments and on the matching the visual space to auditory space for establishing a sense of spatial presence.[11,12,13]

*Fabrication* refers, in a design and architectural context, to the process of making or producing an object of interest. Advances in manufacturing supported by CAD and by CAM[14] have enabled new fabrication methods to be employed, such as rapid prototyping, which can be described as the automatic construction of physical objects or representations of these by use of tools such as 3D printers, stereolithography machines and selective laser sintering systems. Industrial designers use methods of this sort in the early stages of the design process for evaluating objects in terms of their appearance, proportions or tactility, often creating either scaled or full-size prototypes. Such advances in fabrication have a clear impact on the designing of the objects in question, stimulating research on new tools and methods of design as in the research conducted within at the Design Fabrication Group at MIT.[15]

The examples above represent research directed at various of design concerned either with the visual and auditory perception of space and the qualities related to it or with associated processes of manufacturing. Research on those aspects of design concerned with exploring the interplay between the architectural qualities of space and the organisation of material and how it

[11] Slater 2000
[12] Billger, Heldal, Stahre and Renström 2004
[13] Larsson, Västfjäll, Olsson and Kleiner 2005
[14] Computer Aided Manufacturing
[15] http://web.mit.edu/ddfg/

produce an efficient load carrying structure is far less common. Such questions are dealt with, from differing points of view, as taken up by Sandaker in *Reflections on Span and Space* and by Olsson in *Strukturmekanik & Arkitektur* (Structural Mechanics & Architecture) the latter currently published (in Swedish) at Chalmers University of Technology. [16,17] Olsson considers the basis for structural design partly in terms of a "least common denominator" for a desired or proposed appearance, on the one hand, and efficient structural behaviour, on the other. He speaks of two fundamental mechanical concepts on a structural level, *equilibrium* and *stiffness,* and to a certain degree of a third concept too, that of *deformation patterns*, as together constituting the basis for a design dialogue that can serve to guide the early sketching process, providing a source of ideas for design and helping provide a better qualitative understanding of compound and complex structural behaviour.

In reflecting upon how dialogue can be carried out between an architect and an engineer, within the framework of the design process, Olsson makes use of two concepts, those of *structure* and *quality*. [18]

He refers to 'structure' as being similar to what the design engineer Cecil Balmond speaks about in his book *Informal*: 'structure' being reducible to a simple system lines, or points, that can be tested, moved, materialised, and be assigned different tasks or properties and again at some point reverting to being system lines with the inherent freedom that system lines possesses.[19] Olsson regards 'quality', in turn, as being a neutral description of an inner value or set of values brought forth through the properties rather than its being an assessment such as that of a good or bad, or of high or low, quality. Olsson considers 'structure' as the common denominator in a professional dialogue and 'quality' as the force that carries the design process forward. He also refers to *pictures* and *models* as providing the basis for a common language that enriches the professional language of the architect and the engineer by expressing and helping to explore the 'qualities' surrounding the 'structure' that gradually takes shape. On the basis of the description thus achieved of the design work in progress, the architect and the engineer can continue creating and formulating in their dialogue and the work associated with it, being supported the whole time by a computer-based tool allowing them:

---

[16] Sandaker 2000

[17] Olsson 2005

[18] When referring to this particular interpretation, single quotation marks will be used, 'structure' or 'quality'.

[19] Balmond 2002

- To initiate and test 'structures'.
- To use one or more parallel 'qualitative' languages able to help them develop 'structures' by gradually giving them shape.
- To produce pictures facilitating a professional dialogue in the context of design.

In an effort to characterise desirable aspects of an appropriate design aid further Olsson identifies six key words:

- *Associate* – to investigate an idea, an abstract conception, a picture or a metaphor.
- *Sketch* – testing, seeking, investigating, comparing objects by use of sketching
- *Build* – creating the stability and stiffness that are necessary by use of building blocks (bars, sheets, blocks, joints, etc.)
- *Change* – strengthening, weakening or changing the structural action and the global properties of an object through its shape, material, support conditions or inner compression/tension
- *Read* – examining pictures concerning either causes (outer loading or support, inner shape or structure) or effects (stress patterns, deformation patterns or eigenmodes)
- *Understand* – understanding how the interplay of outer loading and support with inner shape and structure results in or alters a particular structural behaviour.

These six aspects of an appropriate design and the ideas characterising them agree with what was observed in the furniture projects described in chapter 2. The need for adequate sketching, appropriate visualisation and an understanding of cause and effect in terms of structural behaviour is fundamental. A question that arises here is to what extent the computer-based tools available serve these needs. If they are not sufficient, how should such a tool or tools be designed?

## *3.1 Computer-based tools*

### 3.1.1 CAE tools

Most architects and engineers are familiar with CAD tools or at least with some kind of computer-based modelling tool. Architects often choose CAD tools that provide fast solid modelling and visualisation possibilities, whereas engineers tend to prefer tools that can be used not only for modelling but also for computations, such as CAE tools that also provide possibilities for scientific visualisation. Both the engineer and the architects appear to regard a computer-based graphical user interface as a valuable part of their professional equipment. A tool combining fast modelling, computation and visualisation seems well suited for facilitating the collaboration of architects and designers with engineers, as the furniture projects indicate.

CAE tools are the most common programs or sets of programs dealing with modelling, computation and visualisation. They are frequently referred to as a software suite, i.e. a group of programs. The different programs are designed to function as related links in a chain leading from the idea to the manufactured product. Such concepts as those of product-development, manufacturing and business processes are important in the design of these programs which is also associated with the fact that their strength in particular lies in the later stages of the design process. Several examples of the CAE tools available are Ansys, NX (formerly known as I-deas) and Pro/ENGINEER.[20,21,22]

Whereas it is relatively easy to create simple geometries and sketch-like models by use of CAD tools, the corresponding procedures can be more complicated when using CAE tools since the modelling process there is designed to provide a smooth transition to the computational parts of the program, the geometric modelling provided being limited by the characteristics that the computational engine possesses. If a model has been created in a CAD program that fails to take the limitations of a CAE program to be later employed into account, problems can be incurred. Problem of this sort were encountered numerous times in the second case study when neutral file formats were used to transfer the geometry from a CAD tool to a CAE tool. Generally when a computation fails, the feedback obtained is cryptic at best and is conveyed in the language of the engineer, making it very difficult to understand without extensive knowledge of the inner workings of FEM computations. Thus, CAE tools cannot be regarded as having the *sketching* possibilities required of a tool intended for supporting the collaboration of architects and designers with engineers in the early stages of the design process.

CAE tools generally have rather advanced *computational* capabilities. They are powerful in creating models and in performing simulations, those of complex material behaviour and nonlinear load response included, as well as in handling different levels and strategies, so as provide accuracy in the numerical treatment of a problem. In the case of more advanced computations, however, the user interface tends to be increasingly compound, creating a barrier for the

[20] Ansys 2006
[21] NX 2006
[22] Pro/ENGINEER 2006

mainstream user increasing the risk of mistakes being made in entering input data. This is one of the reasons for which programs having a steep learning curve. Traditionally, programs of this sort have served exclusively as tools for the engineer, their being strongly coloured by the professional language of engineers, making it difficult for architects to use them to support their professional work. Despite recent updates of CAE tools have been significantly improved in terms of their graphical interfaces and *visualisation* capabilities, the sheer numbers of functions implemented still make them very difficult to navigate. CAE tools thus in no sense cater to the needs of users with a lesser *level of knowledge* of structural mechanics and of FEM.

Although a standard CAE tool may seem like a colossus in this context, use of it can be essential for being able to readily complete the long design process, beginning with the initial geometry being defined and ending with plans for manufacture of the last building block being completed. Yet to facilitate a professional dialogue between the architect or designer and the engineer before the geometry of the final shape the work in question is to take has been chosen, another type of design tool, or a corresponding addition to whatever CAE tools are currently being used may the needed. This is a tool that provides a quick interaction between a given model, involving both shape and physical properties, and computational results that have been arrived at. Also, loads should be able to be applied to the model without detailed geometry needing to be defined.

### 3.1.2 Real-time tools for Structural analysis

There are some few computer programs that make a point of offering fast modelling and fast response in connection with computations a structural mechanical behaviour. These programs are often used in an educational context because their providing typical “trial and error” functionality suitable for laboratory lessons.

At Johns Hopkins University in Baltimore, Maryland in the US, a set of virtual laboratories has been designed in efforts to show how web-based tools and graphics can be used to introduce students to experimentation, problem solving, data gathering, and scientific interpretation. One of these laboratories, called Bridge Designer, allows students to design trusses. [23] By use of nodes, elements and the loads a truss are designed there on a grid-patterned background. In modelling the support conditions, certain of the nodes are defined either as being fixed, or as being either a horizontal or a vertical roller.

[23] Bridge Designer 2006

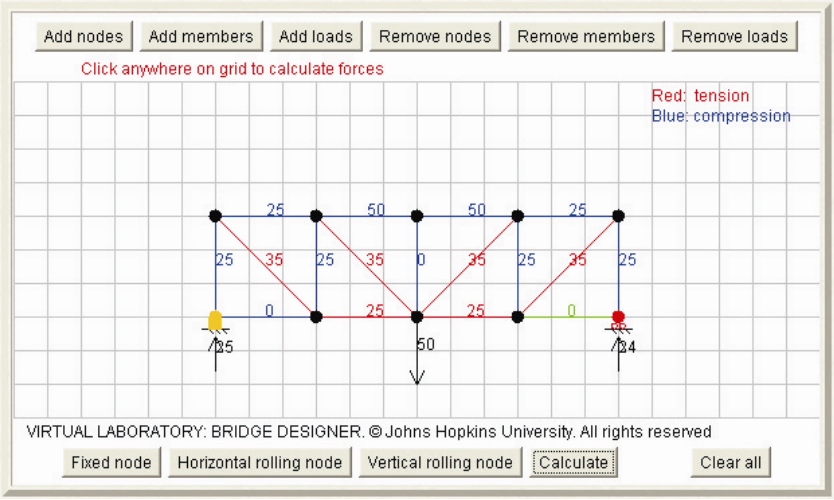


**Figure 17** The Bridge Designer GUI.

The computational engine is restricted to *statically determinate trusses*, meaning that the normal forces in the elements of the truss are calculated under equilibrium conditions. If a truss that is built fails to fulfil the rule for a static determinate structure (m+3>2j, where *m* is the number of elements and j is the number of joints), a warning appears.[24] If the rule is met but the structure still represents a mechanism, the program either returns vastly exaggerated values or provides the message 'No solution. Matrix is singular'. After the truss has been modelled, computations are made and the results are visualised. Although in this program one can switch from modelling to the computations and vice versa by the click of a mouse button, the program can be regarded as having real-time characteristics since it is so simplified and since there are only few steps to be taken from changing the design to having the computational results of it visualised. However, Bridge Designer lacks many desirable aspects such as those of adapting to different levels of knowledge and

A structure is regarded as being *statically determinate* if only one possible load path can occur in it. If even a single structural member of it were removed, such a structure would collapse. In contrast, if more than one load path can occur within the structure it is regarded as *statically indeterminate*. Such a structure can afford to lose structural members up to a certain point and still remain stable. A structure is regarded as being a *mechanism* if it does not possess structural members in number or type, or not arranged in a proper way, for a load patch to occur. Such a structure is not stable.

[24] Meriam and Kraige 1993 (page 178)

providing a GUI suited to easy and fast exploring so as to facilitate the comparison of ‘structures’ at an early stage of the design process.

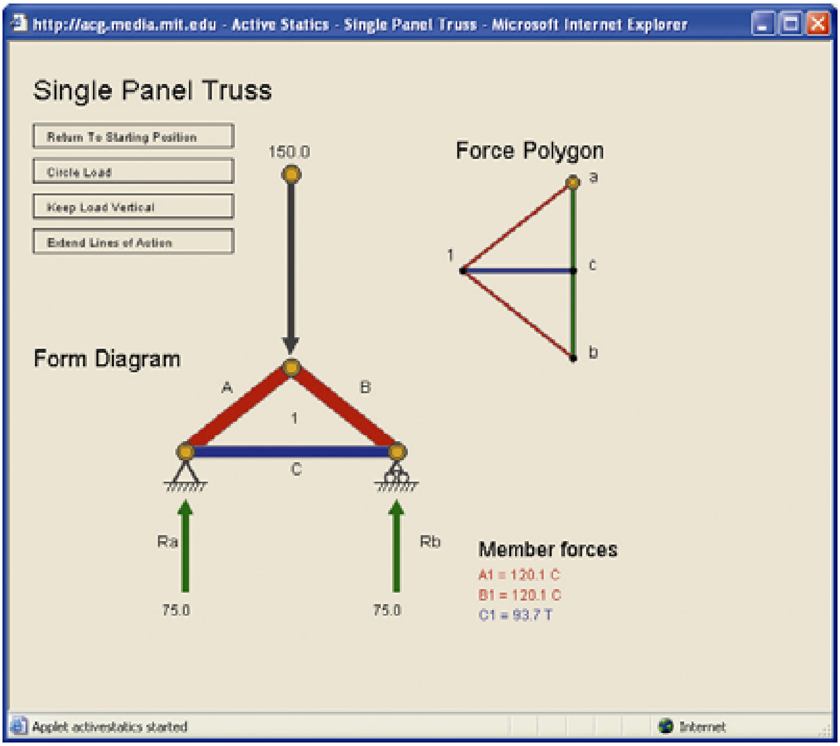


**Figure 18** Active Statics, the example of single panel truss.

Another application, developed at Massachusetts Institute of Technology (MIT), is Active Statics, which offers a set of web-based interactive demonstrations that get the user to experiment with the relationship between structural form and the forces involved.[25] This tool is intended not as a design tool but as an educational tool, but it provides real-time response through its interactive GUI. The user is presented with eight predesigned examples of trusses. In each example, the proportions of the truss can be changed, the computational engine (based of graphic statics) continuously updating the visualisation of it, the forces involved being predefined in the same way as the truss.[26] Deformations and normal forces are used to visualise how the truss responds to the loads applied.

This tool provides fast answers to questions pertaining to structural behaviour. The big downside of the tool is the fact that there is no sketching but only the

[25] Active Statics 2006

[26] Zalewski and Allen 1997

possibility of changing the proportions of the truss. There is no possibility of adding new components to the truss or of defining alternative loading and support conditions. Despite this tool not having been conceived as a design aid for developing structures, the design of the GUI provides an understanding of structural behaviour in a manner useful in any design process.

Other, more versatile tools that provide possibilities for real-time response are Multiframe, for example, and various tools developed by Dr. Software, Dr. Frame3D being one of them.[27,28] The latter allows the user to interactively build and analyse three-dimensional frames. Besides the ordinary building blocks (i.e. nodes, elements and supports), Dr. Frame3D offers advanced modelling features such as hinges, shell elements and joints. The computational engine offers support for second-order analysis and for analysis of plastic hinges besides the usual linear static analysis. The visualisation of normal forces and of deformations is superimposed on the main view of the structure. Moment- and shear diagrams are also possible to show in the GUI after frame analysis has been performed. Numerical diagram values, lengths of separate elements, force values, etc. are listed beside the main modelling window. In this respect, Dr. Frame3D is far more advanced than the other two tools mentioned above.

Although the GUIs in these programs are rather extensive, they lack the simplicity of the other tools, especially in terms of their accessibility to less knowledgeable users. These tools can be regarded, on the basis of experience gained in the furniture projects, as being engineering tools rather than tools for architects and designers since they clearly speak the language of the engineer. Using these tools in a professional dialogue with an architect or a designer would be likely to require that the engineer lead the dialogue resulting in an uneven distribution of the contributions the two parties would make to the design process.

### 3.1.3 Computer games

Surprisingly, certain computer games have many properties that make them an appropriate design tool. A popular game of this sort is Bridge Builder, which has an interface based on two-dimensional graphics.[29] The task in the game is to build a bridge over a river and make it strong enough for a train to pass over it. The player disposes of a fixed budget, meaning a limited number of structural elements. This forces a player to think cleverly in terms of structural stability in

[27] Multiframe 2006
[28] Dr. Frame3D 2006
[29] Bridge Builder 2006

completing the task. At any time during the modelling of the bridge the player can test its strength. The game shows the player, through the deformation behaviour of the bridge being animated and the stress distribution being visualised whether the bridge is stable and how it is deformed and stressed during loading. If the bridge collapses, the deformation pattern and the distribution of stresses give the player a hint of the weak spots in the design. The player is given a new chance in that case, through bearing these weak spots in mind, to improve the design to the point where the bridge is stable enough for the train to pass over it.

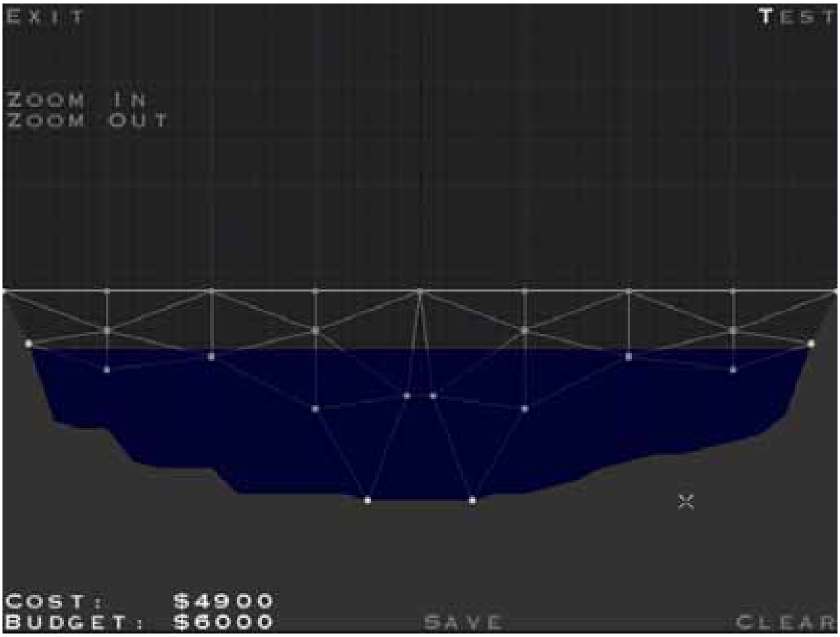


**Figure 19** The Bridge Builder GUI.

Although this game is very limited in scope and is geared towards entertainment, it does facilitate a manner of working that combines elements of fast modelling with fast feedback in dealing with structural behaviour. The game also provides an increasing level of difficulty meaning that experienced players are challenged just as much as less experienced players are. Bridge builder is a single player game, no collaboration with any other players being called for or supported during the game.

There are other games similar to Bridge Builder, that provide more advanced three-dimensional graphics, such as Pontifex and Bridge IT.[30,31] Although the graphical interface is very much better than in Bridge Builder, these games still involve the same structural problem, that of creating a two-dimensional truss strong enough for a train to pass over it. Although these programs are entertaining, they fail to meet the requirements for an adaptable and interactive medium for professional dialogue. Since the creation of the bridge in question is linked to a predefined scene, there is no opportunity for unbound modelling, which is a prerequisite for the sketch-like work of the architect. The limited computational capabilities and the lack of different interface settings also make these games unsuitable for any other purposes than what they are designed for.

There do not seem to be any tools available meeting the needs for a design aid that are formulated above posed by the furniture projects that are described here. As has been indicated, developing possibilities for an easy way of working involving sketching and experimentation and a use of GUI aimed at demonstrating the interplay between cause and effect in connection with structural behaviour is of particular relevance here. Characteristics such as these typically related to the early stages of the design process. Most available computer tools possess only a few of these desirable characteristics. Instead, they are suited primarily to the late stages of the design process. This indicates the need of a new computer-based tool, or an addition to existing tools, aimed at supporting the early stages of the design process.

## *3.2 Basic patterns in Structural Mechanics*

In a discussion of architecture and structural design in the Nordic countries, the two architects Bobert and Lund speak of the differences that are typical between engineering and architectural work and how the two professions could benefit from a common language richly provided with pictures.[32] They begin with a caricature of each of the two professions showing clearly how various difficulties in a collaborative design process can occur. The authors go on to describe how architects and engineers define their respective area of expertise today and how they can gradually share these areas with each other in order to improve their collaboration. Initially, the engineer and the architect have very few points of connection in the practice of their respective professions. At first the engineer

---

[30] Pontifex 2006
[31] BridgeIT 2006
[32] Bobert and Lund 1991

defines his/her territory as being that of forces and dimensions, whereas the architect feels his/her territory is that of space and geometry. Bobert and Lund continue by describing how the two professions could increase the bases they have in common in the design process. In a somewhat utopian description that is provided of a possible way of their collaboration the two territories appear to have merged to a considerable degree, but it is still clear that the particular area of expertise of the architect is that of space and the particular area of expertise of the engineer is that of forces. The areas of expertise that belong to both the architect and the engineer are those of *geometry* and *dimensions*. The authors conclude their discussion by saying that professional collaboration of this kind can be developed by use of analogies and pictures, their regarding this as the basis for a common professional language.

In *Conceptual Structural Design: bridging the gap between architects and engineers* Popovic discusses communication challenges between architects and engineers.[33] Using a holistic interdisciplinary approach in educational programmes Popovic argues that the gap between architects and engineers can be bridged. Similar to Bobert and Lunds suggestion, such an approach would inspire to new architectural and structural solutions being created.

In *Strukturmekanik & Arkitektur* (Structural Mechanics & Architecture), Olsson proposes two different bases for a common ground for a dialogue of design between the architect and the engineer.[34] These two bases express two opposite and complementary ways of understanding a structure: understanding it as a continuum, on the one hand, and as a set of discrete reference points, on the other. These two differing but related aspects of a structure are seen as providing patterns for an ongoing dialogue on design. In an effort to clarify these concepts, Olsson presents an example in which a sketch of a sheet of material in which there are randomly distributed holes is interpreted by use of each of those two bases for conceptualising it, see Figure 20.

[33] Popovic 2003

[34] Olsson 2005 (page 100)

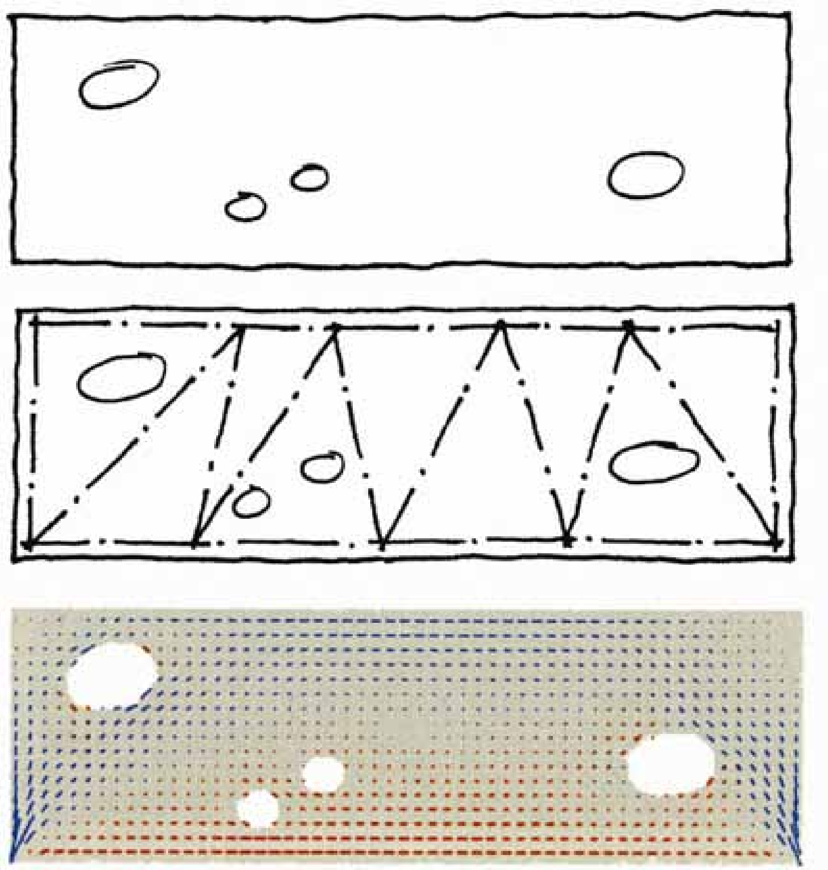

**Figure 20** A sheet of material with randomly distributed holes, interpreted as a set of discrete points connected to each other by system lines and as a continuum. (source: Karl-Gunnar Olsson)

In the second image in the figure the load-carrying 'structure' is thought of as a set of discrete points connected by straight one dimensional lines in the material. These *system lines*, as they are called, can represent bodies formed of bars that follow the lines so as to create a truss, or bodies that link the points but occupy space outside the lines, creating either a frame or something else. The discrete points that are connected by system lines represent a potential pattern for carrying loads, something which Olsson terms a 'structure'. In interpreting a shape in this manner, where the connecting points are located is of central importance in their determining the resulting structural behaviour. The picture shows where the material can be said to exert an effort to carry the load in an efficient way, where the holes can be expanded without appreciable loss in load-carrying capacity, and where the boundaries are for a potential 'structure'. These system lines are drawn with a particular shape in mind, this manner of interpreting them supporting the transition from a rough idea of a shape to a functional 'structure' embodying the final form this basic shape takes.

In the third image in the figure, the load-carrying structure is conceived of as a continuum which in contrast consists of continuously joined material. When a continuum is affected by a load, it creates a continuous network of "material chains" that carries the load from where it is applied to the supports. These material chains can be interpreted as 'structures' similarly to that of the system lines and can be identified in the third picture by the compression and tension lines there. In contrast to a system of discrete points, this manner of interpretation supports the transition from a set of prerequisites (loads and supports) to a shape, the material in effort telling one how it wants to carry the load. The shape can be developed, moulded and refined by identifying the material chains evident in the picture.

The computer tool ForcePAD was developed in connection with this latter possibility of imagining a structure as a continuum.[35] This program aims at enhancing the user's understanding of mechanical concepts, particularly in the sketching process. ForcePAD's graphical interface was developed with well-known image editing programs in mind, with the idea of making it easy for the user to recognise buttons, menus and functions, see Figure 21. The process of modelling in ForcePAD is similar to the use of drawing tools in image editing programs, use being made, for example, of pens, buckets and brushes. These tools facilitate the kind of unbound sketching most architects and designers usually employ in the early stages of the design process. The 'paint' applied here, however, represents stiffness, which means that the user is actually drawing with the use of stiffness as an element. The sketching is restricted a two-dimensional space, at the same time as stiffness is the material property that, alongside with shape, governs the load distribution within a body.

[35] ForcePAD 2006

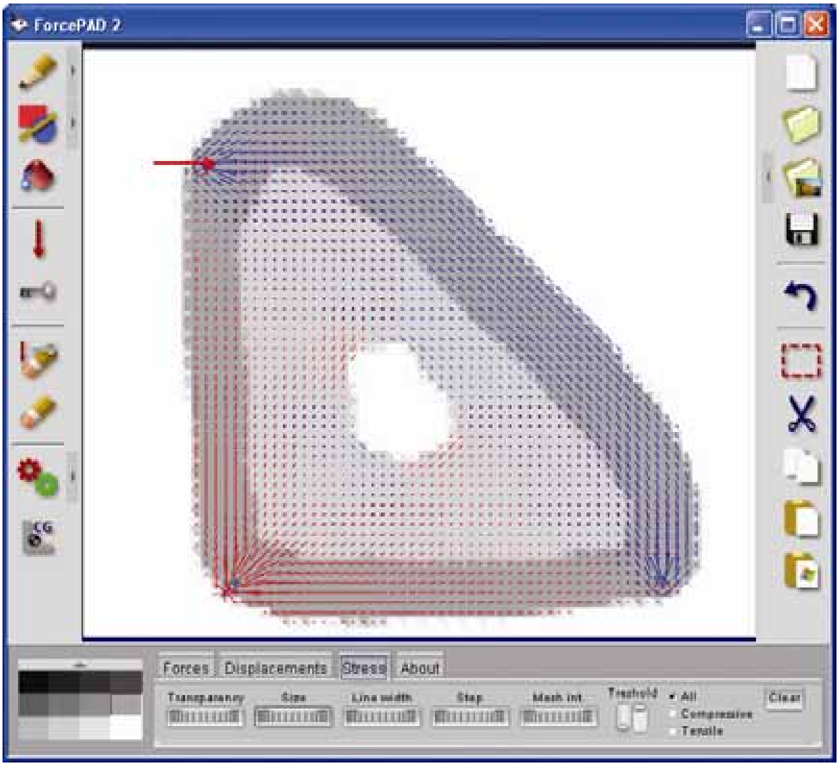


**Figure 21** The ForcePAD GUI.

From the beginning, the virtual canvas on which the sketching and painting is done is a homogenous continuum without any stiffness. By adding stiffness by use of pens and brushes, the continuum is given an inhomogeneous distribution of stiffness. In order for ForcePAD to run a computation, forces and supports needs to be applied. Further on in the process, the supports can be seen as a part of the sketching since they affect the total stiffness of the model. When a sketch has been made, the characteristics of the model as a whole are computed by use of FEM. The results of the computations are visualised and are superimposed on the model. The distribution of stiffness within the continuum determines how the loads that are applied are carried. It is visualised in terms of the distribution of the stresses present. The manner in which the results are visualised can be manipulated to a certain extent in line with the preferences of the user. Reviewing the results of computations a sketch can be modified and new computations be carried out that support fast response of the visualisation provided of the resulting structural behaviour to changes in the design.

ForcePAD facilitates a manner of working in which a particular shape is achieved by investigating the interaction between the forces applied and the supports available. Thus, there is no need of having an idea in advance of what the final shape will be in order to use ForcePAD. The user can let the investigation of the continuum and the prerequisites (loads and supports) be a source of inspiration in determining the final shape. In many respects ForcePAD encourages a study of materials, almost like the kneading of a piece of clay. To achieve the opposite – that of the architect or designer having a set idea in mind of a particular shape without knowing the loads or the supports to be involved – requires the use of a design tool of a different type. Here the approach of imagining a load-carrying structure as being a system of connected points has advantages.

In a workshop called *To fasten points* (Att fästa punkter), Olsson encouraged architectural students of his to suggest 'structures' by use of this latter approach. This assignment for his students was inspired by Ture Wester's *Structural order in space* and by discussions that he and Morten Lund, who co-authored the article "Kraft och rum" (Force and space) referred to earlier, had.[36] The students were first divided up into groups, each group being given one of five different concepts – those of jump, vibrate, rhythm, fall and abolish – and being instructed to give it a physical shape representing the boundaries of a possible 'structure'. In a second phase of the task, the students were instructed to define their shape by a limited number of points within a three dimensional space. In order to create a stable 'structure', each point had to be connected to three other rigid points, each point thus needing to have three supporting legs. If a point is made rigid in this way it can act as a support for another point. By following this simple rule or principle, the students could transform their "cloud of points" into a stable 'structure'. In the third and last phase of the task, they were instructed to use a given material of specified material properties – either steel, wood, masonry, concrete or glass – to create their final shape.

[36] Wester 1984

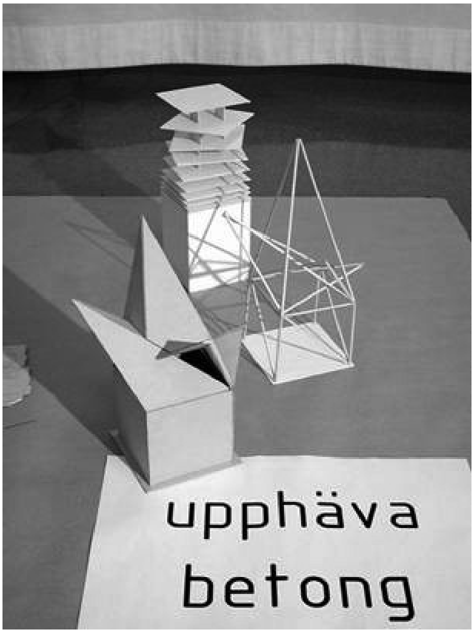


**Figure 22** From the workshop *To fasten points*. (source: Karl-Gunnar Olsson).

The approach used in this student assignment shows a possible way of translating the idea of a particular shape into a potential load-carrying ‘structure’. Since no computational tool was available that could effectively support this particular approach, a computer-based aid to conceiving a load carrying ‘structure’ as consisting of a system of connected points was needed, one similar to the computer program ForcePAD but supporting the concept of a continuous space as a load carrying ‘structure’. In order to facilitate the development of such a tool, a set of requirements to be placed on it was formulated, specifying how the design aid in question should be structured, as well as motivating and explaining this. These requirements were regarded as a specification for the development of the computer program and of its associated GUI.

# 4 Defining a concept – pointSketch

The different studies carried out in the Furniture IDS project referred to in chapter 2 allowed a set of qualities to be selected that appear important for a computer-based tool intended for use in the early sketching process with which a design task begins. These qualities were summarized in Olsson (2005) by use of the terms *associate*, *sketch*, *build*, *change*, *read* and *understand*, see chapter 3.

Four different types of pictures appear potentially useful for exploring how these qualities could best be made visible to the user of an appropriate computer program:

- Pictures showing the concept, metaphor or whatever basis to the idea of the shape or shapes to be created.
- Pictures showing the visualised shape and surfaces of a building in as true and accurate a way as possible.
- Pictures based on the computational model of an engineer showing the physical properties that affect the behaviour of a load-carrying ‘structure’.
- Pictures showing different actions (behaviours) of a building or whatever, such as forces present in the structural elements of it and displacements of the ‘structure’ when loaded.

Most of the qualities referred to above are linked with some one of these types of picture rather than with all of them, yet the most essential quality – that of *understanding* – appears first in an interchange of content between two or more such pictures.

In the search for a computer-based tool aimed at exploring, in the early stages of a design process the form, material and forces involved, utilising the idea of these four parallel pictures helps create the initial basis for developing one’s design. These four different types of pictures will be referred to as the *visualisation modes*, a term that can be considered the first core concept of the computer tool proposed here.

It’s important in a collaborative situation that all members of the group in question remain focused on the task at hand. Accordingly, in working with a computer tool collaboratively, it is important that every action taken be transparent to all those involved. Both the symbols employed and the commands to be executed should be simple enough to be understood by each of the participants. Yet when the knowledge and skill of the group increases, or when planning of a more general sort needs to be supplemented by more precise results measured in mm (millimetres) and in kN (kiloNewton), it is desirable that one be able to perform more precise computations. Bearing this in mind leads one to the second core concept of the proposed computer tool, this being the ability to work at different *precision levels*, the level employed at any given time depending upon the knowledge the members of the group possess. When the precision level is changed, the amount of information utilised and the complexity of the functions one is able to execute change accordingly.

*Sketching* is an important tool for exploring the form and function of a 'structure' in the early stages of a design process. Architects have a long tradition of using sketching as a manner of working, their thus having considerable experience with it. The third core concept contained in the proposed computer tool then is the ability to engage in a sketching process of some sort.

In addition to these three core concepts, based on experience gained in the Furniture IDS project, two features that have been seen as functions that are desirable to include in the proposed computer tool have been added. In Olsson (2005) the concept of *canonical stiffness* was introduced. This concept allows the properties of a growing 'structure', such as mechanisms that make a 'structure' unstable or deformation modes that make an otherwise stable 'structure' weak, to be computed and visualised. In addition to the concept of canonical stiffness, a *link to CALFEM* has been created. The latter is an advanced computer-based feature offering the possibility of modifying specific aspects of the numerical (finite element) model. The code available for this can be seen as representing an additional "visualisation" mode, one visualising the algorithm that creates the mechanical model and executes analysis of it.

This computer tool has been given the name of *pointSketch*, which is derived from the "Fixed points in space" idea conceived of by Olsson, and refers to the process of sketching the outline of an imagined shape by the use of points which are eventually will be connected to each other to produce a stable 'structure'.

## *4.1 Visualisation modes*

The architect and the engineer can communicate knowledge of their respective professions to each other through use of different visualisation modes. The visualisation involved in each mode needs to be both simple and sufficiently complete to enable members of the other profession to understand it. Creating pictures in the visualisation mode most accessible in each case to the respective profession enables members of the two professions to come closer to each other their professional dialogue. The pointSketch concept is that of communication by way of these modes serving to create an area of common dialogue between the architect and engineer, as discussed in Bobert & Lund (1991).

Support for the initial part of the preliminary design process is provided by a *Sketch mode*, in which pens and brushes are used to create an initial drawing of the 'structure' conceived. Three additional modes – the *Appearance mode*, the *Physics mode* and the *Action mode* – modes which facilitate an understanding of the interplay between appearance and mechanical behaviour are provided. The process of switching back and forth between these four different modes in evaluating structural behaviour represents the basic core of the pointSketch concept. A fifth 'mode', the *numerical mode*, intended for use by engineers that provide a numerical interpretation of the 'structure' as designed as well as the possibility of carrying out numerical computations and manipulations, is to also be available. This 'mode' differs from the others in displaying a sequence of numerical operations rather than the information involved being presented in the form of a graphical picture.

### 4.1.1 Sketch mode

In the sketch mode the architect or designer draws the first tentative lines needed to visualise in a rough way the idea of the shape he/she has in mind. Such a drawing can be saved and be used as a starting point for further exploration carried out in the appearance, physics and action modes. Since switching between modes can be carried out at any time during the design process as a whole the deformation patterns visualised in the action mode, for example, can also be an inspiration for the designer or architect in making a new set of sketches in the sketch mode. It is an iterative process.

### 4.1.2 Appearance mode

The appearance mode represents the attempt to imagine the current 'structure' as it would look like in reality, i.e. in a photorealistic way whereas the sketch mode and the physics mode provide abstract views of the 'structure'. In contrast, the appearance mode provides a view to which both the architect or designer and the engineer can relate without having to compromise in linguistic terms. Viewing the 'structure' photorealistically also creates the possibility of placing the 'structure' in different environments, changing lighting, employing different textures, adding other elements such as people and furniture, and so forth in efforts to create an accurate representation of the imagined 'structure'. This is usually what architects refer to as *visualisation*, whereas *scientific visualisation* is the term what engineers use in referring to the making of pictures or animations in an effort to present a mechanical model or its behaviour in a readily understandable form. In this respect, the appearance mode facilitates *visualisation*, whereas the physics and the action modes facilitate a *scientific visualisation* of the object in question. Switching between the physics mode and the appearance mode indicates what consequences a change in the load carrying system would have on the appearance of the 'structure'. Moving a system line in the case of the physics mode could have a strong impact on the appearance of the 'structure' and various details of it, something which is hard to predict in working with only abstract mechanical models.

### 4.1.3 Physics mode

In contrast to the appearance mode, the physics mode visualises the 'structure' as an engineer would define it: as a mechanical model of a load-carrying system. In the physics mode, a 'structure' is built up by use of the essential components of a model of this sort – in terms of system lines (or surfaces) linked together at points (or lines) of connection. System lines are representatives then of a local one-dimensional mathematical model describing the behaviour of a single element in the 'structure', their having the advantage for design purposes of the final definition of a two- or three-dimensional shape being partly left to the future, when the final design is decided upon.

The visualized model used in the physics mode shows the causes of a particular structural action. The objects shown in this mode are the physical building blocks of which the mechanical model is constructed. Each object has parameters related to its characteristics, such as a bar with end points, an elastic modulus and a cross sectional area, and a support having a particular direction to it. The magnitude of the different parameters of an object indicates the degree to which the object affects the structural behaviour of the model. Adding, removing and editing the objects or components involved and their parameters provide the possibility of experimenting with the 'structures' behaviour. In pointSketch, there are four different objects available to the user: *nodes*, *bars*, *supports* and *loads*.

The *node* is the most basic object in the physics mode. Having a collection of 'points in space' allows an initial pattern from which a 'structure' can emerge to be established.

The nodes can be linked together by physical elements. Such elements have the potential of acting as intermediaries of the forces between the nodes. In terms of *truss theory*, the nodes are connected by straight *bars* that transmit normal forces acting perpendicular to their cross section. According to the pointSketch approach, an immediate and sufficient interpretation of this is that these elements represent simply bars, although a much broader interpretation is also possible, this serving to make pointSketch a design tool of considerable potential.

In the theory of bars, the governing equation is a one-dimensional differential equation describing the relationship between an applied load and the resulting deformation. Derivation of the equation requires the choice of an x-axis, or of a *system line*. To simplify expression of the equation, this system line is normally located at the centroid of the cross section. In one-dimensional theories, all entities – even those describing materials and areas at some distance from the axis – are presented as functions of the x-axis. Thus, the system line contained in the theory serves to represent a real bar with a three-dimensional volume. However, – and this is the point – a system line of this sort can also represent a broader group of elements, provided they in some sense fulfil the bar equation or can be related to it. Examples of such elements, structural in character, are shown in Figure 23.

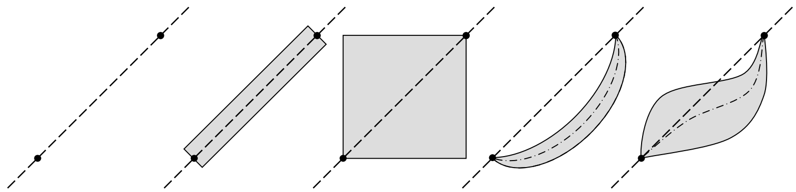


**Figure 23** System lines interpreted as load paths through structural elements of differing shape.

Assume that each of the elements shown in Figure 23 has a homogenous cross section, i.e. a constant material stiffness.[37] The first element, Figure 23b, is an ordinary bar in which the system line is located at the centroid of the cross section. Figure 23c shows an element with an expanded cross section, but with the centroid still located along the system line. This implies that here too the resultant of the normal stresses acts along this system line. Since the shape of the third element, that shown in Figure 23d, lies outside the system line except at its two end points, bending occurs there. There is also a second, curved system line located along the length of this element. This curved system line is at the centroid

[37] For an inhomogeneous cross section, i.e. one consisting of materials of differing stiffness, the location of the system lines in the cross section is moved to a point where the relation $\int EAy \cdot dA = 0$ is fulfilled along the two principal axes of inertia.

of a cross section. Figure 23e, finally, shows a freely shaped body connected to the two nodes of the major system line. This element too has a second, likewise curved system line. This curved element acts as the centroid of any cross sections.

Whereas the first two elements act in pure tension or compression, the second two also act in terms of bending. The latter require a more material-consuming cross sections since these elements also have to be provided with bending stiffness. The bending moment introduced is directly proportional to the distance between the two system lines.

The *bar* object could in initial use of pointSketch be interpreted as an ordinary bar element in a truss system. Yet the basic idea of pointSketch is that the *bar* object be seen as a system line able to represent a wide range of load-carrying bodies of differing spatial extension.

A *support* represents a connection between a 'structure' and rigid surroundings. In pointSketch, the most basic description of a support that is possible has been chosen, that of its preventing movement (translation) in a particular direction. In a two dimensional model, two supports acting in differing directions is required to describe a completely rigid node. Similarly, it takes three supports to describe a rigid node in a three-dimensional model.

These three basic objects are part of a graphical language through which the mechanical model can be described and be made legible. At the same time, the simple objects chosen allow mechanical behaviour of far more complex character to be described as well. For example, if a 'structure' rests on a support that can be deformed, it can be modelled as a semi-rigid (partly restrained) support by placing a *bar* of varying stiffness between an additional, rigid node and the node that is to be partly restrained. At higher *levels of precision*, see section 4.2, no such roundabout way is needed.

The concept of *stiffness* is of essential importance in the physics mode. Bars and supports are those elements that contribute to providing a 'structure' stiffness, yet a global stiffness of the entire 'structure' is first present when the nodes are defined and the individual elements are connected with them. Whereas bars and supports contribute to stiffness, the nodes organise the stiffness, resulting in the stiffness being distributed globally. In a model described in terms of only the three basic objects – node, bar and support – sufficient information is available to describe the model's structural properties in terms of the finite mechanisms of stability, canonical stiffness and the presence of certain load cases for which the 'structure' is weak. In a construction phase, i.e. before all the building blocks have been put in place, this also allows the mechanisms an unfinished 'structure' contains to be inspected.

To proceed in exploring the design of a ‘structure’ as a whole, a fourth object, that of *load* can be added. In many design tasks, particularly in the case of buildings, the load cases involved are known in at least an approximately way. The load-carrying ‘structures’ to be included can then be explored with these particular load cases in mind. The object load can be introduced in two forms, as a point load or a distributed load caused by gravity.

### 4.1.4 Action mode

Whereas the appearance mode shows the ‘structure’ in question as it would look in real life, and the physics mode show the ‘structure’ as it can be interpreted from a structural mechanical point of view, the action mode shows the structural actions that take place as a function of the stiffness distribution and the possible load. Each of these actions is expressed in two different ways – as *internal forces* and as *deformations.* If a load is applied to a stable ‘structure’, internal forces are developed in the bars. These are called normal forces and work in terms either of compression or tension. The distribution of normal forces in a ‘structure’ shows then how the ‘structure’ acts in carrying a load. If the ‘structure’ is statically determinate, there is only one possible distribution of the internal loads to which it is subjected. If the ‘structure’ is statically indeterminate, the internal load distribution depends on the stiffness distribution of the individual bars. The stiffer a bar is, the more load it can carry. Besides the normal forces, reaction forces which act at the supports can also be shown to be present. The magnitude of these forces indicates to what degree each of the supports “carries” the ‘structure’.

In pointSketch, the deformations to be expected can be revealed by two different types of computations: linear elastic analysis of deformations and forces and analysis of canonical stiffness (see section 4.3). The deformations computed by linear elastic analysis are responses to loads that are applied whereas canonical stiffness and the deformations connected with it as computed by eigenvalue analysis are responses to the organisation of the bars and supports of the ‘structure’. Analyses of the first type require that the ‘structure’ be stable, whereas analysis of the second type can also be performed when mechanisms are involved. Since the deformations a mechanism produces are of no generally defined magnitude, they can only show the deformation pattern the mechanism can produce. When such a pattern is visualised, it can be scaled so as to be expressive of the mechanism and show how the ‘structure’ might possible collapse unless bars and supports were provided. Pictures indicating these various effects and possibilities can be very useful in the early stages of the design process, as is taken up further in section 4.3. However, the scaling feature should be used with fact borne in mind that the linear elastic theory of structural mechanics is based on the assumption of small deformations. The pattern of deformations a given mechanism can be thought to produce is only reasonable as long as this assumption is fulfilled.

The visualising of deformations and forces found in the action mode is a way of helping the user to evaluate a design (or shape) in terms of structural mechanics. One type of feedback possible here is information concerning, or the visualisation of, zero energy modes. The number of zero energy modes is always less than or equal to the number of additional elements or supports needed to secure structural stability. The deformation modes involved enable one to determine at what locations additional elements could be needed. Other types of potential feedback concern the force and deformation patterns that can appear in a stable 'structure'. The relative magnitude of the normal forces that act on the 'structure' can be shown by colours of differing intensity. In general, feedback of this sort helps the user make his or her own decisions concerning what alterations in the design are needed in order to ensure that the structural behaviour it produces will be effective. The user, to be sure, needs to be familiar with the basic concepts of structural mechanics in order to glean adequate information from these visualisations. The required level of knowledge for basic use of pointSketch is similar to what is taught in this area at architectural and design schools. A trained engineer should have no problems in interpreting the visualisations provided in the action mode.

### 4.1.5 Graphical settings

The four visualisation modes described either show a sketch the user makes, indicate the appearance of the 'structure' at some stage, provide a mechanical model of it, or indicate the structural action that can take place. Included together with each picture are alternatives indicating choices that can be made of how the information to be shown should be represented enabling different parts or aspects of the information at hand to be emphasised.

Thus, the possibility of customising the verbal or numerical information and the visualisations one is provided with is a part of the pointSketch concept. In the physics mode it would be possible to customise the graphical appearance of the nodes, bars, supports and forces in order to maximise their legibility. If in the action mode only the compression forces are considered to be of importance for a specific investigation, the possibility would be available of omitting at the moment presentation of other types of information.

These customisation functions allow the levels of precision selected (see below) to be adjusted so as to not overwhelm a novice user, for example, with graphical settings that would disturb more than help. Default settings are provided for the case that no graphical customisation is chosen.

### 4.1.6 Switching between modes

By switching between the four modes described, both a single user and a group of collaborating users can investigate the interplay between appearance and structural action, and obtain answers to questions of the “what if” type. Switching between the physics and the appearance mode, for example, can provide answers to questions of how a change in the mechanical model would affect the appearance of the ‘structure’ in question. Still more useful could be to switch between the physics and the action mode. Here the user can gain insight into the interplay between patterns of shape and structural behaviour. The functions associated with these modes that are available are designed to provide such guiding information during the entire sketching phase, from placement of the first node on the drawing to inserting the last bars to be incorporated into the design.

The nodes are one type of component which visualisation can help to clarify that can be made common to all four modes. In the sketch mode a design idea can be reduced to a set of points. In the appearance mode the points would represent geometrical locations. In the physics and the action modes the nodes would represents points of reference in the computational model. The computational model can be established or expanded by linking ends (or corners) of system lines (or surfaces) to the nodes. Forces can be applied to nodes. Supports can also be connected to the nodes. The nodes are the cornerstones to the idea of pointSketch, their also serving as the basis for the name this concept has been given.

### 4.1.7 Link to CALFEM

CALFEM is an interactive computer program developed for learning to use the finite element method (FEM).[38] It was created as a toolbox to the general mathematical computer program MATLAB.[39] Through a link being provided from the model used in pointSketch to the CALFEM code corresponding it, the user is able to gain insight into the workings of the computational model. Since the aim of CALFEM is to help the user gain working knowledge of the concepts

[38] CALFEM 2006
[39] MATLAB 2006

and the strategies available in FEM, the code has been designed to be as intuitive as possible. It has also been found to be useful in investigating rather complex mechanical actions. This feature of pointSketch is intended for the advanced user in particular.

Whereas pointSketch appears as a GUI, CALFEM has the appearance of a sequential computer code. Thus, CALFEM does not have the same safety features as pointSketch in terms of identifying mechanisms and the indicating of possible weak spots, although the programmer can introduce such features on his/her own. If the user tries to run a linear computation on an ordinary CALFEM program and it fails to work, the standard MATLAB error messages is provided, one pertaining in this case to numerical calculations rather than to structural mechanics per se. The link to CALFEM provides the advanced user valuable possibilities of developing and fine-tuning the computational model contained in pointSketch. In addition, the proficient CALFEM user can make use of pointSketch through the link in question enabling the basic geometry of a 'structure' to be outlined in pointSketch in a quick and easy way, a process that would be both tedious and time-consuming in CALFEM.

This link can be seen in a way as a fifth mode or a submode to the physics mode. Comparing the graphical representation of the computational model in the physics mode with its numerical representation in CALFEM gives the user insight into how the model is translated to the data structures and variables that are fed to the computer for the final computational process. This enhances the pedagogical value not only of pointSketch but also of CALFEM. When merged, pointSketch and CALFEM provide the user the possibility of following the idea of the shape of a 'structure' from the first tentative graphical sketch of it to its concrete numerical representation.

## 4.2 Precision levels

The possibility of adjusting the level of precision in pointSketch enables it to meet the needs of a wide variety of users, from those with only limited knowledge of structural mechanics to those with far more extensive knowledge. The level of precision need also be fixed for a session as a whole, but can be changed during a session to accommodate a user who feels comfortable with the current level as such but wants at some point to access to additional functions available only at higher precision levels. Accordingly, pointSketch facilitates learning at the same time as it supports the professional dialogue in the design process. The basic appearance of a visualisation provided does not change when the level of precision is changed, but remains the same so as to avoid confusion and misinterpretation. What differs between the different precision levels is the extent to which more advanced settings for the computational model in the physics mode and certain additional visualisation functions or settings in the physics and action modes are available. At the lowest level of precision visible interaction between the GUI and computational model is very limited. The majority of the more advanced settings are present as default values that are hidden from the user. The reason for this is to be able to inform the novice user in a basic way about mechanical actions that occur or may occur without burdening such a user with unnecessary mechanical precision.

At higher levels of precision, increasingly many settings are required in order to run a computation. Although the computational engine employed is the same. The difference lies in the GUI providing the user a greater number of possibilities of influencing the variables used in the computational model and of customising details of the model's increasing accuracy. Whereas the lower precision levels help the novice user to get started, the higher levels allow the advanced user to give number designations to entities and to introduce the more accurate mechanical modelling needed for improving the precision of the results.

The process of giving a bar the level of stiffness seen as optimal can serve as an example of how precision levels are implemented in the physics mode. At the lowest level only reference values employed. The bar stiffness – Young's modulus (E) times the cross-sectional area (A) – is given the reference value of 1. In the sketching stage, using reference values in this manner should suffice for determining and evaluating the most basic aspects of structural behaviour implied on the basis of the current sketch. At higher levels of precision the user is allowed to enter the parameters connected with the stiffness of each individual bar. This process is usually simplified by first defining the template material(s) to which reference is to be made whenever a new bar is added to the sketch. Sketching at this higher precision level is time-consuming and relatively complex, but it does provide the possibility of producing a more refined and detailed computational model. Note that the more detailed a computational model is, the easier is becomes to maintain consistency when switching between the appearance and the physics mode.

In the action mode the precision levels affect mainly the way in which the visualisation provided can be modified by the user. At the lower levels, the possibilities for modifying the visualisations are limited, only such simple things such as the deformation scale being available for this. This is also the default setting for visualisations in the action mode, regardless of precision level. At higher precision levels, more functions are available to the user, such as colour scales, toggle switches for certain parts of the visualisations provided, animated deformations and the browsing of eigenmodes. Accordingly, the novice user is given a fast and easy way of viewing the results of computations, whereas the advanced user is given the possibility of modifying the details and the appearance of the visualisation available.

## 4.3 Sketching

Three of the basic ideas included in the account above of how pointSketch is conceived are those of supporting the translation of the idea of a particular shape to a form in which the shape is effective in mechanical terms, working with different modes that can be interpreted as being varying expressions of the idea of this shape, and considering a set of discrete points in space – the nodes – as being the common denominator between different visualisation modes. The location of the nodes, common to all four modes, varies in the course of the process in the degree to which it is fixed, due in part to the preconditions for the real task at which the planning is directed, an in part to the connection their location has with the idea of the space or shape embodied in the design.

The first stage of sketching could be done simply on a piece of paper that is then scanned and imported into the *sketch mode*. It could also be done on a virtual canvas by use of drawing tools similar to those available in image editing programs such as Adobe Photoshop.[40] The next step of the process is the application of points at key locations of a sketch or of a representation of the design concept in some other form. In pointSketch this could be carried out in the sketch mode, or also in the appearance or the physics mode.

Sketching in the appearance mode involves creating and organising geometries in an effort to produce photorealistic pictures representing the imagined shape. This is the same as modelling in an architectural sense. In the two-dimensional case, the modelling can consist of creating drawings similar to those created in such programs as AutoCAD.[41] The aim of these drawing functions, similar to that which these more common programs have is to make it easier for the user to recognise functions, buttons and menus. The same argument holds true for the three-dimensional case, the functions there being similar to those in such solid modelling programs as Autodesk VIZ and SolidWorks.[42,43] To these modelling

[40] Adobe Photoshop 2006
[41] Autodesk AutoCAD 2006
[42] Autodesk VIZ 2006
[43] SolidWorks 2006

functions one can also add such capabilities as those of varying the lighting, textures, colours and so forth with the intention of enhancing the model's possibilities of exploring matters of physical appearance.

Going from the sketch or appearance mode to the physics mode involves the need of defining the properties of the mechanical model. These mechanical properties can include such matters as the location of system lines and specification of the support conditions. This connecting of the sketching process with mechanical realities is carried out in the physics mode, where the user is provided with a simple tool for evaluating design ideas. Certain limitations are introduced and guidelines for the decisions to be made for ensuring stability and safety are established.

The architect and the engineer, through making use of system lines rather than the actual structural elements involved, are able to share a common language that provides information essential to both of them. They can collaborate in determining a suitable set of global structural actions the building or whatever may show. They can then, in part individually and in part in collaboration with each other, proceed with details referring both architectural and structural needs. The latter part of the process involves the choice of appropriate shapes and suitable materials as well as the sizing of building components on the basis of the loads to be expected. This manner of working gives the architect considerable freedom in shaping the final form of the building or other object created. At the same time, the engineer is given the information needed to perform the computations required for ensuring stability and safety. Such an approach could be seen as contributing to the creation of a common ground for the architect and the engineer in the design process, one that Bobert & Lund consider extremely important.[44]

Since most of the structures involved in building design are three-dimensional, it should be possible to perform sketching concerned with three-dimensional space. At the same time, two-dimensional space offers a fast and often ideal way of testing an idea, which is pedagogically advantageous in collaborating in the analysis and shaping of structural behaviour in the early stages of the design process. PointSketch provides the possibility of working in both two and three dimensions.

---

[44] Bobert & Lund 1991

## *4.4 Analysis of canonical stiffness*

An important aspect of the pointSketch concept is the possibility it provides of determining those mechanisms to be considered in evaluating 'structures' at a stage of the construction one is planning when the 'structure' as a whole is not yet finished and thus has perhaps not yet been made stable. Another feature is the possibility it provides of evaluating deformation patterns for which an otherwise stable 'structure' would be weak under some load conditions. The mechanisms and deformation patterns that create weaknesses can be accessed by use of static eigenvalue analysis, as Olsson (2005) describes. The eigenvalues determined can be arranged in order of the degree of weakness involved, and be interpreted as being indicative of the inherent stiffness of the 'structure'. Olsson refers to this as the *canonical stiffness* of the 'structure'.

In terms of FEM, each *canonical stiffness* $\lambda$ (eigenvalue) is a scalar that links a deformation pattern **a** (an eigenmode) to a load distribution **f** (a load vector).

$$\mathbf{f} = \lambda\, \mathbf{a}$$

The eigenmode with the lowest degree of stiffness is called the first eigenmode. It represents how the 'structure' is most easily deformed. This also means that deforming the 'structure' in accordance with to this deformation pattern takes the least amount of force. Because of the direct connection found between eigenmodes and load cases, eigenvalues can be seen as a way of ranking different load cases in terms of how sensitive the 'structure' is to them.

Another way of using canonical stiffness values as design parameters is to measure the degree to which they decrease or increase when the design of the 'structure' is changed, i.e. how the structural stiffness changes when changes in the design occur. If the canonical stiffness value increases after change in design, this implies the change to have made the design more resistant to the load case currently being considered, more force than before being required for the 'structure' to be deformed by this load case. This same comparison can be made by switching from the physics mode to the action mode and assessing the canonical stiffness before and after the change in design.

A special case of canonical stiffness analysis concerns one or more canonical stiffness values being zero. In such a case the eigenvalue analysis leads to so-called *zero energy modes*, i.e. to no energy at all being needed for the 'structure' to be deformed. Zero energy modes show a distorted deformation pattern that can be interpreted as the manner in which the 'structure' would tend to collapse – which can be seen as a mechanism. The number of mechanisms of this sort is equal to the minimum number of bars or supports required for ensuring the stability of the ‘structure’. This can be helpful for the user in endeavouring to figure out how to make an unstable ‘structure’ stable. When it’s unstable, no normal forces that can be visualised exists.

Most structural mechanics analyses require a fully defined geometry in order for useful results to be obtained. The results of this eigenvalue analysis, however, can be just as useful when the first few nodes and bars have been are placed as at a far more advanced stage. Switching between the physics mode and the action mode during the construction process can help the user to in successive stages arrive at a stable ‘structure’. Building a large ‘structure’ which, though it may look stable, is in fact highly unstable, can be indicative of there being a large number of different mechanisms, making it difficult to determine where additional bars and supports should be placed. Under such conditions, it can be helpful to build the ‘structure’ step by step and to receive feedback continually on the stability of the ‘structure’, based on analysis of the canonical stiffness involved.

Canonical stiffness analysis represents a powerful tool for the creation of ‘structures’ and for evaluating changes in design. However, its use also requires good judgement. The deformation patterns encountered are potential patterns and not deformations that necessarily occur. They are thus similar in this way to those deformations shown to occur in response to applied loads in a stable ‘structure’. In the theory employed it is assumed that the deformations occurring here are small. This is important to bear in mind in considering deformation patterns of relatively large magnitude.

# 5 Tools and methods for development of the applications

Software development tools, external programming libraries and methods of interface design were needed for creating applications based on the pointSketch concept.

## *5.1 Programming language and software development tools*

This required use of development tools capable of handling numerical computations (FEM) and graphics (GUI). In order to reach as many users as possible and cater to their needs, the applications had to be designed to run on as many different computers as possible, i.e. to be *platform-independent*. Personal experience in developing similar applications suggested that GUIs having interactive graphical objects that were clearly defined (nodes, bars, supports and forces in this case) should be included in the programming structure. Organising the code in this manner involves every graphical object having a corresponding computational object, which facilitates connections between the GUI and the computational engine, making an *object oriented* programming language preferable.

C++ was found to be the most suitable programming language for creating the applications since it is object oriented and can be written as being platform-independent.[45] In contrast to most web-based applications, such as Java applets for example, C++ creates standalone applications that tend to make optimal use of available hardware resources, facilitating fast response from the computational engine. C++ is the most commonly used programming language in the context of advanced graphics and numerical computations. Creating the intended applications required external programming libraries since C++ itself does not support the creation of GUIs and various other computational operations. Such object libraries are bound to a particular programming language, C++ being firmly established in the programming community offered a vast array of possible libraries to choose from which facilitated the programming work.

A software development tool was needed for writing code, compiling code, debugging, and linking the external libraries. Visual Studio .NET permits a

[45] Stroustrup 2000

platform-independent programming code to be developed.[46] The programming code developed there can be used then as basis for the creation of files executable on arbitrary platforms, such as Windows or Linux, but for the purposes here Visual Studio only supported the Windows environment.[47,48] Since I have experience using this tool and since Windows was the most common platform for persons whose help was desired in testing the applications, this tool was selected.

## 5.2 External programming libraries

Although Visual Studio offers standard functions related to the C++ language, for developing programs utilising advanced graphics and FE computations external additions are needed. Fast Light Toolkit (FLTK) is a platform-independent GUI toolkit written in C++ that has been used in this connection.[49] Besides the basic building blocks for a GUI (buttons, menus, textboxes and so forth), FLTK also supports the two-dimensional graphics needed for creating the views related to the different visualisation modes.

A supplement to FLTK was required for the three-dimensional application of pointSketch. Interactive Visualisation Framework (Ivf++), developed at the Division of Structural Mechanics of Lunds University, was chosen for this purpose.[50] Ivf++ is written in C++ and utilises the graphic standard OpenGL, which makes it platform-independent, just as FLTK is.[51] This toolkit contains no complete visualisation library, consisting instead of a basic set of classes that are easy to use and can be extended in various ways, leaving it to the developer to organise and create composite graphical objects appropriate to the context at hand. Ivf++ has custom-made functions for interfacing with FLTK, which facilitated replacement of the two-dimensional graphics.

The computational engine of pointSketch was written from scratch, being inspired by the MATLAB toolbox CALFEM.[52,53]CALFEM gave the FEM model a clear and well-arranged structure. An essential part of a FEM-model is the matrix-oriented organisation of the data involved. Since standard C++ supplied by Visual Studio does not offer such data structures or numerical functions, Newmat C++ matrix library was used for creating the data structures and

---

[46] Microsoft Visual Studio .NET 2006
[47] Microsoft Windows 2006
[48] Linux 2006
[49] Fast Light Toolkit (FLTK) 2006
[50] Interactive Visualisation Framework – ivf++ 2006
[51] OpenGL 2006
[52] MATLAB 2006
[53] CALFEM 2006

numerical functions required.[54] Newmat, which is similar to FLTK and Ivf++, is based on C++ and is platform-independent.

## 5.3 Methods for interface design

Creating an intuitive graphical user interface is by no means an easy task since every person interprets such as interface in his/her own way. One of the most important books on this subject is "Designing Web Usability" by Jakob Nielsen.[55] Although the book concerns web design, the matters of usability taken up are applicable to most types of user interfaces. Nielsen also has a webpage with additional information useful in this context.[56] In Table 1, ten heuristics of user-interface design that Nielsen developed are presented. They are called heuristics because they are more in the nature of rules of thumb than specific guidelines for usability. Thus they can be used for evaluation purposes and also as aids in designing an interface. To avoid the most common pitfalls of usability in designing the pointSketch GUIs, these ten heuristics had to be kept in mind.

**Table 1** The ten heuristics for proper GUI design presented in Nielsen's webpage.

| |
|---|
| **Visibility of system status**<br>The system should always keep users informed about what is going on, through appropriate feedback within reasonable time. |
| **Match between system and the real world**<br>The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order. |
| **User control and freedom**<br>Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo. |

[54] Newmat C++ matrix library 2006
[55] Nielsen 2000
[56] http://www.useit.com

| |
|---|
| **Consistency and standards**<br>Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. |
| **Error prevention**<br>Even better than good error messages is a careful design which prevents a problem from occurring in the first place. |
| **Recognition rather than recall**<br>Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. |
| **Flexibility and efficiency of use**<br>Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. |
| **Aesthetic and minimalist design**<br>Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. |
| **Help users recognize, diagnose, and recover from errors**<br>Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. |
| **Help and documentation**<br>Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large. |

# 6 pointSketch2D and pointSketch3D

Both pointSketch2D and pointSketch3D are a manifestations of the general pointSketch approach. The difference between the two is in the appearance of the visualisation modes, represented by two-dimensional graphics in the former and by three-dimensional graphics in the latter case. The basics of the GUI are the same in both programs. It was pointSketch2D that was chosen for a more extensive implementation, aimed at demonstrating the form realisation of the pointSketch concept can take. In contrast, in pointSketch3D the functions implemented, except for the most basic ones that had to be included, where only those functions that clearly show the differences between the two programs due to the additional dimensions being included in the latter one. The present chapter contains a detailed presentation of the GUIs that were implemented.

Certain delimitations were made in the functions implemented in the two programs. Both of them are limited in terms of the structural behaviour considered to analysis of axial load paths along the system lines that are present, i.e. two-dimensional trusses in pointSketch2D and three-dimensional trusses in pointSketch3D. The possibility of analysing load paths created by transversal load (bending) and load acting to the side of the system line (torsion) has been left for future development of the system. Possibilities for *analyses of canonical stiffness*, on the other hand, are fully implemented.

As described here, the pointSketch programs have four *visualisation modes*. The main purpose of creating the programs was to implement a sufficient degree of functionality for the pointSketch approach to be able to show how it can be used to facilitate the design process generally. The process of switching between the different visualisation modes so as to facilitate the user's understanding of the interplay between architectural quality, 'structure', and the mechanical structural behaviour involved is a central aspect of the pointSketch approach. Two modes were chosen to be implemented, the physics mode and the action mode.

Because of its being visualisation modes generally that were chosen for implementation, *sketching* is done directly in the physics mode. This encourages users to sketch by use of points similar to the assignment of "fästa punkter" Olsson[57] proposed. Modelling of this sort would be a possibility and would continue to be a strength even when the pointSketch concept is fully

[57] Olsson 2005

implemented. A limitation of the present version is that the user is unable to create or to import sketches for use as a help in defining ‘structures’.

Emphasis in pointSketch2D and pointSketch3D is placed on functions relating to lower levels of precision, or on qualitative rather than quantitative aspects of modelling. For this reason, no full range of precision levels is implemented, but only enough to show how the precision levels resented make available or unavailable various program functions, the level of knowledge required in connection with them also being important.

The possibility of *customising views and visualisations* provided is likewise implemented in terms of the two visualisation modes (although only for pointSketch2D). In the physical mode, the user can modify the appearance of the building blocks used for creating the ‘structure’ aimed at. In the action mode, the user can also modify to a certain degree the appearance of the visualisation of the computational results which is provided depending on the current level of precision. A high level of precision being involved requires that the user’s knowledge of structural mechanics is at a high level since otherwise there can be considerable risk of the user’s misinterpreting the visualisations that appear when these changes in response to modifications the user makes in how these are to be presented.

## 6.1 Structure of the GUI

In both pointSketch2D and pointSketch3D the GUI consist of 4 parts; the menu bar, the action-and-tool bar, the virtual canvas, and the message prompt. The menu bar at the top of the GUI contains general program functions such as those of opening a saved file, saving a file, exiting the program, and making use of the full-screen mode. In addition, the menu bar contains certain functions specific to pointSketch generally such as precision levels, settings for customising the views provided, and a link being provided to CALFEM. The nodes, bars, supports, and forces provided, and how they can be added to the virtual canvas or be modified, are selected by the use of the action-and-tool bar located to the right of the virtual canvas. All of this applies to the physics mode. In the action mode, the bar contains settings for the visualisations being shown on the virtual canvas. The form of visualisation mode to be used currently is chosen by pressing the appropriate button in the group of buttons in the section of the GUI at the lower right. The modelling of the ‘structure’ as done on the virtual canvas located at the centre of GUI in both programs. The message prompt at the bottom of the GUI provides the user with feedback from the program.

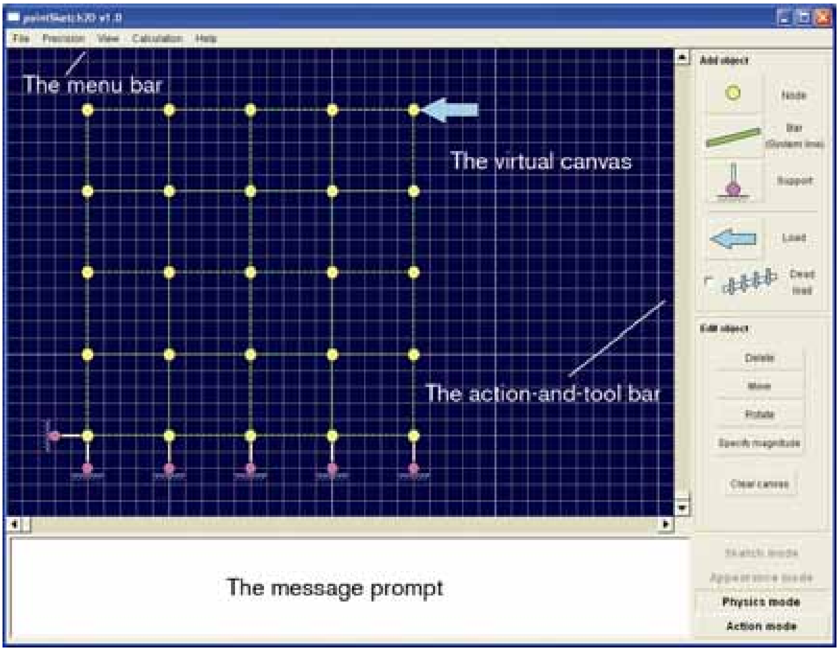


**Figure 24** The GUI in pointSketch2D, currently in the physics mode.

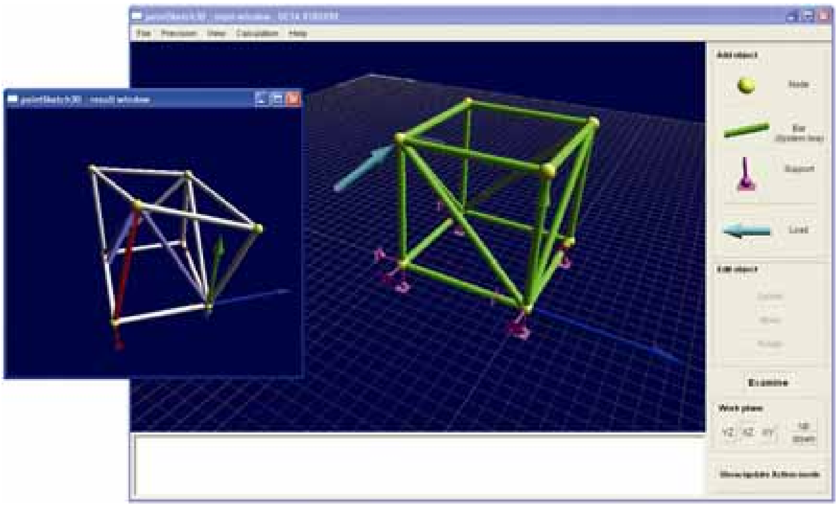

**Figure 25** The GUI in pointSketch3D. Note the separate window for the action mode.

The programs allow the user to design two-dimensional or three-dimensional ‘structures’ on the virtual canvas in the physics mode. Such a ‘structure’ is interpreted as being a truss and is analysed by the computational engine when the program is set to the action mode. If the computations made detect the ‘structure’ being unstable, the program displays appropriate visualisations that help the user make it stable. When the 'structure' is stable, the program shows the user visualisations that describe its structural behaviour. At all stages of the design process, switching between the various visualisation modes is possible, users also being provided continually with information to help them understand the relationship between cause and effect here in terms of design suggestions and the impact various changes would have on the structural behaviour involved.

## 6.2 The menu bar

The fourth of Nielsen's heuristics (Consistency and standards, see chapter 5.3) states that one should try whenever possible to follow platform conventions, since users tend to be familiar with them which in turn minimizes the risk of users becoming confused. The menu system most generally employed was created with this in mind. The separate menus tend there to be placed from the left to right near the upper edge of the window. Three of them at least found in most programs, specifically *File*, *View*, and *Help*, involve general settings. Menus in additional to these are often ones specific to the current application, in the case of pointSketch *Precision* and *Calculation*, resulting in there being five separate menus in pointSketch.

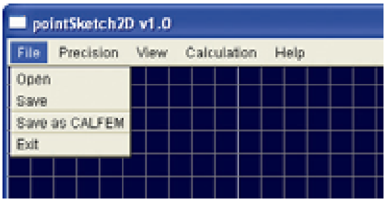


**Figure 26** The menu bar in pointSketch2D.

The File menu contains functions for such tasks as opening a saved model, saving a model that is open, and closing the program. The *Save as CALFEM* option is a function that is rather specific to pointSketch. Since it concerns file handling, it is placed in the File menu. The View menu contains functions that alter the appearance of the window, such as toggle functions (on/off) for the full screen-mode, as well as functions for displaying the grid on the virtual canvas and activating snap-to-grid function. The Help menu provides access to general information concerning the program, such as its official website and contact information.

The Precision menu contains a list of the precision levels the user can choose between. The higher the level is, the more advanced the interaction is that can take place between the program and the user, a higher level requiring greater knowledge of structural mechanics on the user's part. In pointSketch2D and pointSketch3D only the lowest two levels have been implemented thus far. This is basically to indicate how the level of precision can be set and how this affects the user's interaction with this menu. At the lowest level of precision the two pointSketch programs require little input by the user for running a computation, the possibilities for interaction there thus being very limited in terms of computational settings available in the model. For example, the lowest precision level uses predefined stiffness properties for all the bars in the truss, whereas the very highest level planned (not implemented here) would require use of certain unique materials, as well as cross-sections being created and being attached individually to each bar.

The Computation menu provides the possibility of viewing and of changing settings related to the computational engine. These settings, which can be regarded as rather advanced, should not be changed unless the user is well aware of the impact this can have on the results presented in the action mode. Entering unrealistic values in this menu can result in faulty behaviour in the course of the computational process, which in turn can result in a misleading interpretation of the visualisations presented. Since some situations may require a slight manipulation of the computational settings, however, this menu is made available, specifically for the high-end user.

## 6.3 The action-and-tool bar

Except for clicking on the virtual canvas with the mouse pointer, most of the interaction that takes place in connection with the pointSketch GUIs involves clicking on buttons of various kinds, such as push buttons and toggle buttons. A push button is a commonly used one which if pressed initiates an action, much like the Save or the Print button in a more conventional GUI. The difference between a push button and a toggle button is that the latter stays down until pushed up again. Accordingly, the action that the toggle button initiates when pushed down remains active until the button is pushed again so as to be lifted upwards again. The toggle buttons can also be placed in logical group in which only one button at a time can be pushed down.

The frame at the right hand side of the virtual canvas contains a set of push buttons used for modelling in the physics mode and for modifying the appearance of a visualisation presented in the action mode. The buttons at the top of the frame is the toolbar, its containing the visual building blocks for the model, specifically *nodes*, *bars*, *supports*, and *forces*, which will be taken up again later. The middle part of the frame is the action bar, which contains the actions available in the modelling process, namely *Delete*, *Move*, *Rotate*, and *Specify Magnitude*. Choosing Delete results in the next building block selected on the virtual canvas being deleted, provided other objects are not dependent upon it. A node, for example, cannot be deleted if it has bars attached to it. The Move action allows the user to move a node from one position to another on the virtual canvas by simple clicks of the mouse, the first click being used to choose a node and the second for placing it in a new position. The Rotate action allows the user to rotate either a support or a force, the first click choosing the building block to be involved and the second click determining its new direction. The numerical values of the parameters attached to each building block can be accessed and adjusted by choosing the action of Specify Magnitude and then clicking block on the virtual canvas on that particular building. The coordinates of nodes and the stiffness of bars can both be accessed, as the direction can be for supports and as the size and direction can be for forces. For pointSketch3D, additional buttons allowing the user to choose the current workplane or to move it in three-dimensional space are provided. A single button entitled “Examine” allows the user to rotate, zoom or pan the current view. While the view is being changed the modelling functions are temporarily disabled.

When pointSketch2D is in the action mode, the action-and-tool bar is changed to provide the visualisation settings required for this particular mode. The top and the middle part of the action-and-tool bar are divided into three frames containing visualisation settings for forces, deformations, and eigenmodes, respectively. The settings for forces and deformations are similar, both of them containing possibilities for adjusting the visibility and the scale of the building block involved. For an unstable ‘structure’ the frame associated with the eigenmodes shows the number of mechanisms that are present, allowing the user to investigate the nature of these mechanisms by dragging slider which is provided back and forth. For a stable ‘structure’ the frame contains settings for investigating the eigenmodes that are computed.

For pointSketch2D, the buttons at the bottom of the action-and-tool bar are used for switching between the different visualisation modes, whereas for pointSketch3D a button allowing the user to open or to close the window dedicated to the action mode is provided.

## 6.4 The virtual canvas

The virtual canvas is the part of the GUI on which all the modelling is carried out. It is the heart of the pointSketch programs since it contains all functions made available by the GUI. Clicking on the virtual canvas sends a signal to the underlying program to which relevant information is attached, such as the number of clicks that have been given and to which coordinates they relate. The virtual canvas can determine then what answer the user is to be given, such as to place a new node at a point defined by the coordinates that have just been given, deleting the bar under the mouse pointer, or picking up a node at the coordinates it has and placing it at the coordinates communicated by a second mouse click. The program's is also dependent upon settings given by the tool-and-action bar, a matter to be taken up again later.

In the physics mode the virtual canvas receives input by the user and creates a model in accordance with it. In order for it to facilitate modelling, the virtual canvas is equipped with a grid, making it easier to maintain the appropriate proportions. The grid can be used as a guide to placing the nodes on the virtual canvas. There is also a snap-to-grid function, which enables more precise modelling to be carried out if this is preferred. These settings are available in the menu bar. In the action mode, the virtual canvas is limited to displaying only visualisations, its not reacting to mouse clicks. For both visualisation modes, the size of the virtual canvas can be set to a larger size than is possible to be shown in the GUI, sliders being provided at the edges of the virtual canvas to facilitate navigation.

## 6.5 The message prompt

The message prompt provides the user with feedback while using the pointSketch program. For example, it can help the user complete an action requiring more than one step to complete. An example is that of adding a support to a 'structure', where the message prompt first instructs the user to choose a node, when that has been done its instructs the user to rotate the support to the desired angle. The message prompt indicates, on the basis of a computation has been carried out, whether or not the 'structure' is stable. Still other tips and hints helping the user decide what step to take the next in creating the 'structure' which is carried out are also made available. Thus, the user should always keep an eye on the message prompt while using the pointSketch programs.

# 6.6 Physics mode - visual representation of the 'structure'

In the physics mode a 'structure' is visualised according to how it is interpreted from the standpoint of physics rather than of how it would look in real life. Thus, the visual building blocks in this mode reflect the building blocks used to create the physics model, i.e. nodes, bars, supports and forces. All objects shown in the physics mode have a direct connection with the structural behaviour of the 'structure' itself, no objects being regarded as having only a visual function. This refinement gives the user the possibility of experimenting freely with the decisive parameters of the computational model that determine the behaviour of the 'structure'.

The sections that follow present a detailed description of each graphical object, its visual representation and the motives behind the choice of a particular representation, i.e. a design rationale[58].

## 6.6.1 Nodes

Since the nodes have no geometrical "body", direction or size, the visual representation of them should be sparse and convey characteristics of *connection* and *position*. In the physical model involved, a node also represents a *frictionless joint*. Nonsymmetrical graphical objects or lines would be inappropriate as visual representations of nodes. Since a circle, or a sphere in the three-dimensional case, is easily associated with position, with frictionless joints as well as with connection generally (as on a map showing the stations and hubs in a subway system), it was chosen for the graphic representation of nodes. Circles, dots or points are often used in other computational programs for representing nodes. According to Nielsen's heuristics, using such conventions reduces the likelihood of the user being confused in interpreting the symbol employed.

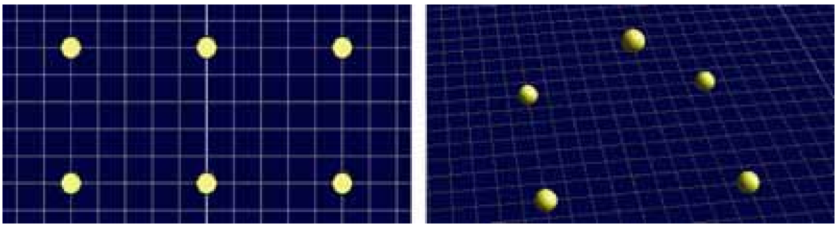

**Figure 27** The node as visualised in pointSketch2D and pointSketch3D.

[58] Record of design decisions, Preece 1994.

### 6.6.2 Bars

Bars are structural elements representing an ideal *load path* between two nodes. The distinction between the starting node and the end one has no meaning for the FE model since the direction of the structural elements in two- and three-dimensional trusses does not affect the structural behaviour involved. The directional distinction is only made to facilitate modelling and visualisation concerned with the GUI.

Besides representing an ideal load path, a bar is also a retainer of *stiffness*. Accordingly, the graphical representation selected should convey a sense of a bar's being able to keep nodes connected or of its representing an ideal load path through an arbitrary shape, and also a sense of its being a tangible 'body' that possesses stiffness. There is a danger, however, in visualising the 'body' aspect of it too much during the sketching stage, since this can divert one's attention from considering a bar as representing an ideal load path to seeing it as representing an actual bar regarded as a structural element. At lower levels of precision in which no settings for stiffness were available, a simple trace line was chosen as the visual representation of the ideal load path. This will be the initial and default version of the visual representation of an ideal load path.

As the precision levels increases there may be a need (depending on whether or not the user chooses to deviate from the default settings) of visualising the impact of bars that differ in their level of stiffness. In such a case the trace line can be replaced by a thick continuous line that varies in thickness and colour, depending upon the stiffness involved. In the levels of precision implemented in pointSketch2D, stiffness is entered as a relative value that is 0-10 times the initial stiffness. The colour of the bars is saturated in accordance with this multiplier. When the multiplier is less than 1, the colour of a bar becomes increasingly whitened, whereas when the multiplier is greater than 1 the colour of the bar becomes increasingly blackened. When the multiplier is exactly 0, the bar is completely white in colour whereas when it is exactly 10 the bar colour is completely black. The visualisation of the stiffness of a bar when a 'structure' is statically indeterminate is of particular interest since it determines the distribution of forces to be shown in the action mode.

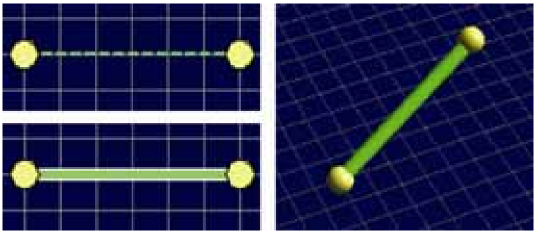

**Figure 28** A bar, visualised as a trace line in pointSketch2D and as a solid line in pointSketch2D and pointSketch3D.

## 6.6.3 Supports

A support restrains a node from moving in a particular direction. In the two-dimensional case, two restraints on movement are required in order for the position of a node to be fixed. In the three-dimensional case, three restraints on movement are required for this. Since both pointSketch2D and pointSketch3D concern trusses for computational models and not frames (as they are in a system of beams), no restraints on rotation are needed.

The visual representation of supports needs to convey the impression of a node being prevented from moving in a given direction. In order to achieve this, the representation employed needs to invoke symbols pertaining both to *rigidity* and to *direction*. Various possibilities for symbols of this sort were considered, a composite symbol consisting of three component symbols, a 'wall' symbol, a pin, and a circle, being selected. The wall symbol denotes rigidity, the pin denotes the direction of the restraint, and the circle denotes a frictionless joint between the wall and the pin, indicating the restraint to be translational rather than rotational.

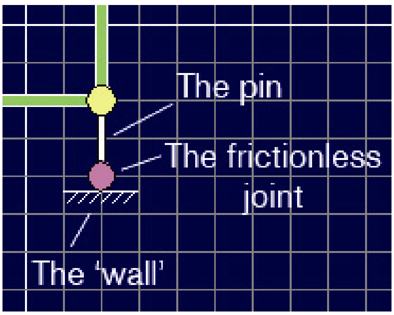


**Figure 29** Support as visualised in pointSketch2D, consisting of a 'wall', a pin and a frictionless joint.

An advantage of such a composite symbol is that it relates to only one direction, forcing the user to think in terms of number of the one-dimensional supports needed for ensuring stability at each individual node, whereas in most traditional symbol systems a single symbol can be used to include more than one direction.

The same arguments concerning the visual representation of supports apply to pointSketch3D as to pointSketch2D. Just as in the two-dimensional case. The type of symbol used is composed of three parts: a shaft represented by a thin cylinder, a frictionless joint represented by a sphere, and a 'wall' represented by a plane. The wall denotes *rigidity*, the shaft the *direction* referred to, and the joint the type of restraint involved.

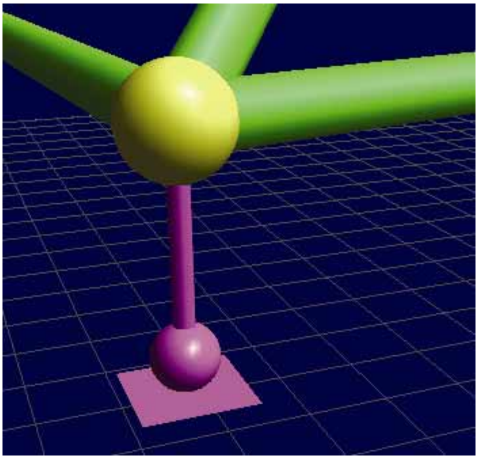

**Figure 30** The support as visualised as in pointSketch3D.

### 6.6.4 Forces

In considering the symbol to be used for forces, not many symbols other than the classical arrow one came to mind. In the context of structural mechanics, forces tend to always have been denoted by arrows of different shape, size or quantity. When a well-established symbol is available, there is usually no reason to create a new symbol. Also, as already mentioned, in accordance with Nielsen's heuristics one should always conform to conventions if failing to do so can lead to ambiguous interpretations. Since there was no reason to consider an arrow as being an inadequate symbol here or as creating ambiguous interpretations, it was chosen for the denoting of forces, both in pointSketch2D and in pointSketch3D.

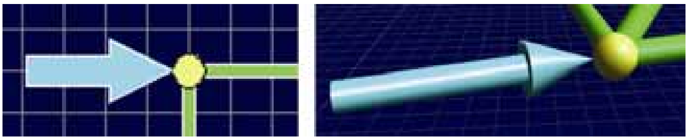

**Figure 31** Forces, as visualised in pointSketch2D and pointSketch3D.

## 6.7 Action mode - visual representation of structural behaviour

By pressing the appropriate button in the lower section on the right-hand side of the GUI, the program can out into the action mode where the structural behaviour of the current ‘structure’ is visualised. The same objects are shown on the virtual canvas here as they are in the physics mode, although the colour and position of them differs in accordance with the deformations and normal forces present as computed by the computational engine.

The reaction forces found are the only new objects added in the action mode. In both the physics mode and the action mode, efforts were made to keep the two programs as similar as possible with respect to the visualisation of objects and of the behaviour these showed so as to make it easy for the user to recognise the symbols involved and what they stand for when shifting between the two programs. Accordingly, the graphical representations in pointSketch2D and in pointSketch3D were created with the aim of their being simply two and three-dimensional representations, respectively, of the same objects and phenomena.

### 6.7.1 Forces

When a ‘structure’ is subjected to a load, the load is carried, by the bars involved, from the node where the load it is applied to the rigid supports. If the ‘structure’ is statically determined, there is only one possible load path, whereas if the ‘structure’ is statically undetermined there are many possible load paths, the load being carried by the load path offering the greatest amount of support, i.e. by the load path having the highest stiffness value. The manner in which the load is carried by the bars the structure possesses can be interpreted on the basis of the visualisation of the normal forces acting in the axial direction of the bars, both those of compressional and of those of tensional character. The normal forces are the only forces pertaining to the bars that from the computations in pointSketch provide.

The graphical representation of the normal forces present involves changes in the colour of the bars in accordance to the size of the normal forces found and with the question of whether these are compressive or tensional. A bar that carries no load at all in the ‘structure’ is coloured white, one that is compressed (subjected to a negative normal force) is coloured blue and one that is under tension (subjected to a positive normal force) is coloured red. In order that the compressed bars can be readily compared in terms of their rate of compression, the blue colour they have is saturated in accordance with the absolute value of the normal force involved. The highest computed compression force possible in the ‘structure’ is 100% blue. As the compression force decreases from this level it becomes decreasingly blue and increasingly white, its becoming 100% white if it reaches the state in which the normal force is zero. If the bar is subjected to tension, the same basic procedure is followed, the bar with the highest tension value possible under the circumstances being 100% red, the bars becoming decreasingly red and increasingly white as the tension to which they are subjected decreases.

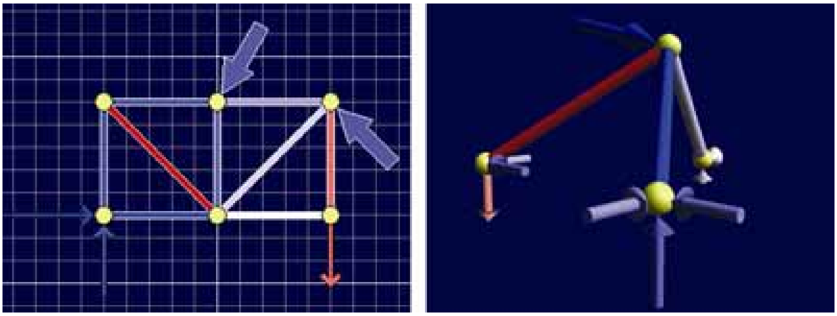

**Figure 32** The normal forces and the reaction forces as visualised in pointSketch2D and pointSketch3D.

Both the normal forces and the reactions forces are of interest in analysing structural behaviour in pointSketch2D and pointSketch3D. Reaction forces that develop in the supports resist the loads applied to the ‘structure’ or those developed within the ‘structure’ (e.g. dead load), ensuring that a state of equilibrium is maintained. In pointSketch the reaction forces act in the direction of the support and can be either compressive or tensional in character, just as in the case of the normal forces. The reaction forces are symbolised by arrows, just as forces in the form of applied loads are in the physics mode.

The arrows used to represent of the reaction forces are thinner than the force arrows employed in the physics mode and they lack the solid arrowhead that the latter arrows possess. Both the length and colour of the reaction force arrows is related to the value of the reaction force as obtained in the computations, their being coloured in accordance with the degree of compression or tension to which they are subjected, in the same manner as for the normal forces. For purposes of ready interpretation, the same colour scale as applied to the normal forces is used providing the possibility of comparing directly the magnitude of the forces of these two differing types. To likewise facilitate comparison of the forces of the two types, the length of the reaction force arrows is mapped to the size of the reaction force involved.

### 6.7.2 Deformations

In pointSketch2D, the visualisation of deformations is activated by clicking on the "Show deformations" box on the action-and-tool bar in the action mode. When this setting is marked the nodes change from their initial position to assume a deformed position. The deformations are always scaled in such a way as to produce a readily discerned visualisation of the deformation pattern. By default, the scale is recalculated after each switch between the physics and the action mode. One can prevent this from occurring by locking the scale when in the action mode. This option is used for comparing the magnitude of the deformations in alternative structural designs or for comparing alternative load cases. The scale can also be changed in line with the user's preferences. This is done by entering a scalar value in the appropriate text box in the menu.

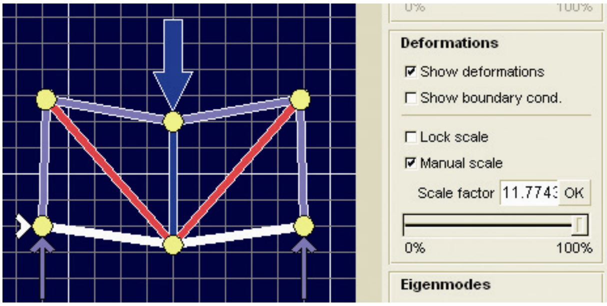


**Figure 33** Settings for the visualised deformations being adjusted.

If the 'structure' is complex, it can be difficult to interpret the deformation behaviour on basis of only static visualisation. A slider is made available to remedy this allowing the user to manually change the scale of the deformations in real-time by dragging the slider from 0% to 100%. Dragging the slider back and forth results in the deformation pattern becoming "animated manually" making it easier to see how the 'structure' is deformed.

In pointSketch3D the option of showing the deformations present is available at all times. The user can view either the deformations associated with a stable 'structure' as produced by the current load case or the deformations patterns associated with the mechanism in question if the 'structure' is unstable.

### 6.7.3 Canonical stiffness

As mentioned in chapter 4.4, eigenmodes can be interpreted in this context as deformation patterns that can be arranged in terms of their canonical stiffness. Visualising the eigenmodes follows basically the same procedure as the visualisation of deformations. The visualisation can be shown in two different ways, depending upon whether the ‘structure’ is stable or not. If the ‘structure’ in its current form is unstable, analysis of the canonical stiffness which is found can be used to identify one or more mechanisms. These mechanisms are visualised and, if there are two or more of them, they are superimposed upon each other on the virtual canvas to form a single deformation pattern. The number of mechanisms identified in connection with the current ‘structure’ is shown in the lower section of the menu to the right of the GUI. Below that number is a slider with a scale ranging from -100% to 100%. By dragging the slider back and forth, the eigenmodes can be scaled accordingly and provide an animation of the mechanisms involved, i.e. the deformations patterns.

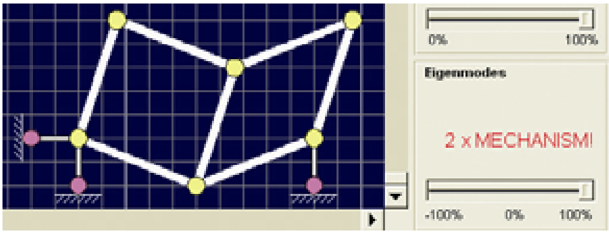


**Figure 34** The number of mechanisms associated with the unstable ‘structure’ at hand.

By adding more bars and supports to an unstable ‘structure’, the number of mechanisms can be reduced, the reduction achieved being indicated in the menu to the right in the GUI. Studying and identifying the extreme deformation behaviour the mechanisms represent suggests to the user where additional bars and supports can best be placed so as to create stability within the ‘structure’. Adding a single bar or support to an unstable structure usually means eliminating one of the mechanisms. When the last mechanism has been neutralised in this way the ‘structure’ becomes statically determined. At this point, analysis of the canonical stiffness of the ‘structure’ no longer reveals any mechanisms. When the ‘structure’ becomes stable, the visualisation of the eigenmodes on the virtual canvas remains unchanged but the menu changes in terms of the information it provides. Instead of indicating the number of mechanisms present, it shows the ranking of the current eigenmode and the canonical stiffness value associated with it. Two buttons are added then to the menu, providing the possibility of browsing the eigenmodes.

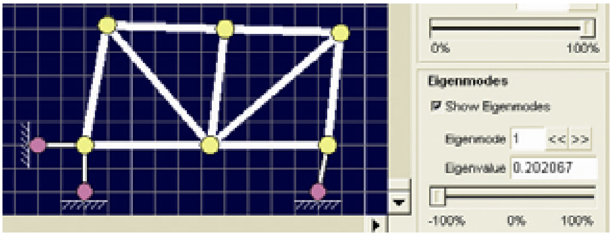


**Figure 35** The settings associated with the stable ‘structure’ at hand.

## *6.8 Application structure and programming*

The GUIs attached to the two pointSketch programs were created using FLTK, see chapter 5.2. A special editor, termed FLUID, is provided by FLTK. It allows the developer to construct GUIs using an extensive palette of such components as buttons, menus, sliders, and frames. In a programming context such objects are called *widgets*. FLUID was used for organising the widgets of interest in appropriate groups and for creating the connections between the widgets themselves. Thus far, these groups of widgets and the system connecting them is little more than an empty framework, functions for such tasks as preparing and executing computations, processing results and creating appropriate visualisations still being required before the programs can be considered complete. Since FLUID does not provide such widgets, new widgets had to be created, ones specific to pointSketch.

Two such widgets were developed, the one for pointSketch2D and the other for pointSketch3D. The virtual canvas provides both the visual representation of the respective widget and the means of interaction between the user and the functions programmed into the widget for the respective pointSketch program. In terms of programming structure, two systems of objects were defined for the pointSketch widgets, the one for graphical representation and interaction and the other for computations. The first of these contains all information and connections pertaining to the visual representation of the different objects both the programs employ (nodes, bars, supports and forces), organised as a scenegraph[59]. In addition to the object structure, the scenegraph contains functions for managing the drawing of objects on the virtual canvas, its serving here as a self-sufficient graphics engine. Visual representation of all the objects contained in the scenegraph are included in both the physics mode and the action mode.

[59] A scenegraph is an object-oriented structure that arranges the logical and often (but not necessarily) spatial representation of a graphical scene.

The second system of objects contains all information related to the computations. Included in this system are the same objects as in the scenegraph but they are interpreted here from a structural mechanical point of view rather than as a graphical system. Just as in the scenegraph for the graphical engine, the objects are organised here in terms of a finite element model. The model provides an input into a computational engine that consists of a set of numerical functions able to compute deformations, forces, eigenmodes and canonical stiffness. When the program is switched from the physics to the action mode, a finite element model is created, one based on information from the current scenegraph concerning such matters as coordinates, bar thicknesses, and directions of forces. After a set of computations has been carried out, the information available is updated in accordance with the forces and deformations that have been computed and is sent back to the scenegraph to provide input for visualisation of it in the action mode. The scenegraph, the FE model, and the computational engine are contained in the widget of the respective pointSketch program.

The major part of the work required for creating pointSketch2D and pointSketch3D was spent on creating these two widgets. When all the grouping of widgets (the pointSketch ones and the ones provided by FLTK) and appropriate connections between them had been created, FLUID was used to create a source code (C++). This source code, together with a few additions to it were then processed in Visual Studio so as to create the executable files required for running the pointSketch programs.

The program pointSketch2D has a single window structure, which means that, in contrast to such programs as Adobe Photoshop[60] and Microsoft Word[61], it cannot have more than one model open at a time. The pointSketch3D program has a slightly different window structure, the physics mode and the action mode each having its own window, obviating the need of the user actively switching between modes. The window for the action mode is updated after any new object is added to the virtual canvas in the physics mode, there thus being real-time interaction between the two modes. This window structure also facilitates the use of dual monitors, since the two windows can be placed apart at locations separate from each other. Creating such a window structure required more extensive programming for implementing all the intended functions than implementing the single window structure in pointSketch2D did. Since the aim in connection with pointSketch3D was simply to implement a sufficient number of program

---

[60] Adobe Photoshop 2006

[61] Microsoft Word 2006

functions to show users the basic differences between modelling and viewing in three dimensions, it was decided to adopt a dual-window approach here. Implementing few functions in pointSketch3D made it easier to create a dual-window structure there.

# 7 Evaluation

Evaluations were needed to determine whether pointSketch was properly designed and facilitated the design dialogue intended. GUIs can be evaluated in many different ways, such as by means of case studies, experiments, interviews or questionnaires. The scale employed also determines the outcome of an evaluation to a certain extent. Evaluations in small and informal groups tend to result in comments on the overall functions and intentions of the program, partly because of the developer being able to participate in the discussion and explain those functions and structures of the GUI the group experiences as being indistinct. The developer can also steer the discussion towards questions concerning the parts of the GUI that are still under development and for which comments are called for in particular. A risk associated with such evaluations is that of test subjects learning from the developer how the GUI should be properly navigated, making it difficult for them to assess how intuitive the GUI would be for first-time users. For determining intuitiveness of a GUI in an adequate way, a rather large test group not previously familiar with the GUI is needed.

In the present study it was decided to divide the evaluation into four steps. The first step was that of *self evaluation* carried out during the process of developing the GUI, the design ideas employed being checked against well established guidelines for GUI design. The second step was to introduce the first usable version of pointSketch to a limited *test panel*, small informal evaluations in which the developer was able to take part in the discussion being carried out. The third step in evaluation was that of a *workshop*, involving a slightly larger test panel to which the program was introduced for the first time and which was asked to comment on how intuitive they perceived the GUI as being. In the fourth and final step in the in the evaluation process, the pointSketch programs developed were made available generally as a shareware, users worldwide being encouraged to download and evaluate it.

## *7.1 Self evaluation*

Creating pointSketch2D/3D was a process of writing a programming code combined with the graphical design of the GUI. This part of the work, which was carried out solely by the developer called for a self-evaluation strategy of some sort so as to eliminate ideas that were ill suited to the program and would otherwise have been kept as a part of pointSketch until the next step in evaluation. The ten heuristics for proper GUI design developed by Nielsen, see chapter 5.3, was used for this evaluation. Comparing the GUI with these heuristics after each step in the development process in terms both of structure and appearance, enabled the most common pitfalls in GUI design to be avoided.

Evaluations made at this stage lead both to a possible redesigning of the GUI and to inspiration for the developing of new functions. For example, the message prompt found at the bottom of the pointSketch2D GUI was a result of the first heuristic (Visibility of system status). In checking the GUI against this heuristic it became obvious that something was missing, specifically that there was no way for the user to know whether a computation was running or whether instead the program was waiting for additional input before being able to start the computation. For that reason a message prompt was added to the GUI to make sure that the system status was visible at all times.

Some of the heuristics had a stronger impact on the design of the GUI than others. The second heuristic (Match between the system and the real world) was very important to bear in mind in creating pointSketch2D/3D, since it concerns the availability and intuitiveness of the GUI, which is one of the major aspects of the whole idea of pointSketch. Despite pointSketch clearly being related to structural mechanics as such, it was always considered important to refrain from using excessive amount of technical language related to the engineering profession in the pointSketch GUI. The whole idea of pointSketch was to create a common language understandable for whatever profession might be involved.

Although placing some of the GUI functions deep within submenus would be preferable in endeavouring to create a logical system for finding certain functions, after reviewing early proposals for the GUI in light of the heuristics that applied (especially the sixth heuristic, Recognition rather than recall) it was decided that certain functions should be made available directly through buttons and sliders that appeared in the main window frame. This lead to the creation of the action-and-tool bar.

In considering the seventh heuristic (Flexibility and efficiency of use) the arguments for giving the advanced user additional possibilities for managing the computational process gained support. This was one of the reasons for creating a *Link to CALFEM*.

## 7.2 Test panel

When a functional version of the program had been developed evaluation on a slightly larger scale was undertaken. The major difference between this process of evaluation and the self-evaluation process was that here more people than the developer alone were involved. However, a full-scale evaluation encompassing a large number of testers can easily result in information overload rather than useful suggestions being presented. To avoid this problem, it was decided that only a small number of people would be invited to this step in evaluation. The fact that everyone involved here was familiar with the project’s aim was conducive to the discussion being held on a general level and not focusing too much on small details. Being concerned above all with general ideas seemed appropriate since at this point it was still possible to make rather considerable changes in the GUI.

The test panel consisted of four persons, not counting the developer. They were representatives of both the engineering and the architectural professions. The developer had the role of discussion leader during these evaluations. This helped ensure that the comments would deal above all with those aspects of pointSketch that the developer wanted to have discussed. Through taking part in the discussions, the developer had a chance to argue for the current design, and to make sure that the comments the group made did not reflect any misinterpretation of the GUI. It was important, however, that the developer be open to the fact that a comment that seemed irrelevant in the sense of a misunderstanding of the intention of the program being involved might at the same time point to a weakness in the design. This part of the evaluation process was iterated several times, with an interval of roughly a month between meetings. During these intervening periods the developer was able to redesign the program on the basis of the test panel‘s comments preparing a new version that could be discussed in the next meeting.

During this phase of the evaluation process, the most common topic of discussion was how the physics mode should be designed and structured. The action mode underwent fewer alterations than the physics mode did, possibly due to the fact that the action mode can be used without as many buttons needing to be pressed, the default view being quite enough in most cases. However, it is in the physics mode that most of the modelling is carried out. This makes it highly important that the GUI is as easy and intuitive to work with as possible. The most common topic discussed was how the modelling process should be carried out. This involved questions both of how the action-and-tool bar should be structured and of how the mouse should be used to drag, drop, and click on the virtual canvas in creating the model.

Both the modelling process and the structure of the functions controlled by the action-and-tool bar were aspects of the GUI discussed considerably by members of the test panel. Just as in the self-evaluation process, the discussions here became more than simply an evaluation, their providing panel members the opportunity to consider different design ideas carefully, to develop them further and to incorporate them into the program. This part of the evaluation process had the strongest impact on the design of the program. Discussions were not concerned primarily with programming details such as how the computational code should interface with the GUI code, but dealt more instead with how the GUI should the structured and the functions that should be available in the program. As more and more meetings were held, the suggestions made became fewer, indicating that the pointSketch programs were ready for the next step in the evaluation process.

## 7.3 Workshop

It appeared appropriate, when the first full version of the program had been completed, that an evaluation using a larger group be conducted. In order to ensure that the comments of this group would not be coloured by it being familiar with the project, the group was one that was new to the program, although it was important that the group be relatively familiar with the context in which the pointSketch programs were to be used, so that the comments its members made would be of relevance.

The opportunity arose of conducting an evaluation by user testing in the context of a conference held at the Royal Academy of Fine Art at the School of Architecture in Copenhagen, Denmark. The conference involved discussions between participants regarding their experiences during the past semester in teaching structural mechanics to architectural students. One topic dealt with concerned computer-based tools available that were suitable for such purposes. This was seen as an opportunity to introduce pointSketch2D/3D and to organize a workshop in which the participants were given a chance to test the programs. The eight participants in this workshop were all of them involved in teaching structural mechanics at some one of the architectural schools in Scandinavia. This was a very fortunate constellation of testers since they were highly knowledgeable of both engineering and architecture and had definite ideas of how a pedagogical design tool should be designed and structured in this context. Some of them had experience earlier with programs similar to pointSketch and two of them had tested very briefly an early version of pointSketch2D.

After a brief presentation of the basic ideas embodied in pointSketch and of the two pointSketch programs, the workshop began. The intention was to give the participants a number of tasks to be carried out while the developer was available for answering questions. The tasks aimed at introducing the participants to the most important features of pointSketch2D. Since at the time of the conference pointSketch3D was at only an early stage of development, it was not possible for the workshop participants to carry out similar tasks with use of that program. The tasks concerned a predesigned ‘structure’ in which a few bars were intentionally removed, making it unstable. It was left to the participants to analyse the ‘structure’ and suggest different ways in which it could be stabilised and evaluate them.

**Table 2** The tasks for the workshop.

| | |
|---|---|
| 1. | Open the file truss.txt and test the stability of the ‘structure’ in the action mode. |
| 2. | Create an inner stability in the ‘structure’, i.e. create a rigid body. |
| 3. | Add external supports until the ‘structure’ is stabilised. |
| 4. | Investigate the distribution of normal forces. |
| 5. | Add one or more point loads and investigate how this affects the distribution of normal forces. |
| 6. | Activate the visualisation of the deformations and investigate the deformation pattern involved. |
| 7. | Add further external supports and remove certain internal supports (bars). Experiment with different combinations of external and internal supports while keeping the ‘structure’ stable. |
| 8. | Make the ‘structure’ statically indeterminate, giving the bars differing degrees of stiffness and investigating how this affectes the distribution of forces and the deformation pattern. |
| 9. | Optional study: Save the ‘structure’ as a CALFEM file and study the MATLAB code produced. |

The ‘structure’ used in the tasks was a well-known example taken from a book by Hans Friis Mathiasen and Erik Reitzel, *Grundtræk af bærende konstruktioner i arkitekturen*.[62] The ‘structure’ presented to the participants, except for the supports and the point load, can be seen in Figure 24. It turned out that three of the participants had used the very same example in their lectures. They had even used it with basically the same set of tasks as those the developer proposed. These participants were thus very eager to see whether pointSkecth2D would give the

[62] Mathiasen and Reitzel 1999

same results as those they had taken up in their lectures. Rather than their being given these tasks as though they were new ones, these participants simply reproduced the tasks they had used in their lecturers, using pointSketch2D instead of the static PowerPoint[63] presentations they had employed earlier. From the standpoint of evaluation, this turn of events was considered a very positive one since these participants were able to carry out again in pointSketch2D the basic tasks they had preformed for their students earlier, a good incentive indeed for exploring a new computer program. In so daring, they could discover how the various functions available in pointSketch2D could help them reproduce the material they were familiar with from earlier. After these three participants had successfully reproduced their lecture material in pointSketch2D and the other five workshop participants had carried out these same tasks for the first time, likewise using pointSketch2D, the developer gave all of the eight workshop participants a brief presentation of pointSketch3D, together with instructions on how to navigate the three-dimensional workplanes. The participants were given the opportunity then of testing the functions implemented in the current version at the time: navigating the 3D space, building a model, viewing results (deformations and normal forces), and altering the geometry followed by recomputation, although no specific tasks were presented. A predefined model was provided that allowed participants to test visualisations in the action mode without having to create a model from scratch.

At the end of the workshop, each of the eight participants was given the opportunity to comment on the pointSketch programs. This provided the developer useful feedback concerning pointSketch generally and the question of whether if it was perceived as intuitive. The following is an account of the comments made by the workshop participants:

*Participant 1* regarded the graphics in pointSketch2D as being very legible and its simplicity to be a major advantage. He considered pointSketch2D to be readily accessible to users, but not pointSketch3D in the form that was demonstrated. He also expressed this three-dimensional version as being much more difficult to use than the two-dimensional version in terms of ease of navigating on the virtual canvas or the workplane. Prior to the workshop, he had used ForcePAD in teaching his students. He was of the general opinion that not be particularly much input into such a programs should be needed in order to obtain output of interest to the user. He considered pointSketch2D to be well designed in this respect. He also felt the awareness the program provides of what is required for creating and

---

[63] Microsoft PowerPoint 2006

carrying out complete set of computations in a structural mechanical context to be useful for his students.

*Participant 2* felt that being able as pointSketch2D to experiment freely with the model in is a clear advantage when one wants to evaluate structural behaviour, his regarding this as being one of the most positive characteristics of pointSketch2D. He also considered that working with the concept of a ‘structure’ (in the physics mode) rather than with a photorealistic representation of it was advantageous, particularly since this also meant there being no need to produce complete drawings of the ‘structure’ in order to carry out an analysis of it. He found it to be easy in pointSketch2D to show the basics of how a ‘structure’ carries its load and the sketching involved to be an advantage from a pedagogical point of view. This participant, who was a first-time user, felt personally inspired by pointSketch2D.

*Participant 3* was one of the lecturers who had used the ‘structure’ presented in Mathiasen and Reitzel’s book in his classes. He could easily imagine pointSketch replacing his PowerPoint presentations, even in its current version and its being a help in his lectures. He considered pointSketch2D to be able to help him in the type of lectures he wanted to give to his students. He noted that it gave them the possibility of analysing on their own the examples used in the course work. He regarded the possibility students had of experimenting freely while using pointSketch2D to be a very positive aspect of it. This participant felt he could easily put pointSketch2D in the hands of his students and give them tasks to carry out with it. He was convinced the students would have no problems in using pointSketch2D on their own.

*Participant 4* considered it to be a limitation that frame analysis could not be used in the pointSketch programs. Although he felt it was good that truss analysis was provided, he considered that the addition of frame analysis would greatly enhance the pointSketch programs. He saw there being parallels between ForcePAD and pointSketch2D and suggested it would be an advantage to integrate them, noting that this would involve the students only having to learn how to use one GUI instead of two. This participant also underlined the importance of being careful in interpreting the visualisations considered when testing different design ideas. He pointed out that a subtle change in the visualisation that resulted could signal a very significant change in the structural behaviour involved, a fact that could easily be overlooked if the user was not observant enough.

*Participant 5* valued very much the pedagogical simplicity found in pointSketch2D, considering this to be a marked advantage when using it to teach principles of structural mechanics. He felt the same degree of simplicity needed to be created in pointSketch3D.

*Participant 6* was one of those who had used an earlier version of pointSketch2D in one of his lectures. In addition to pointSketch2D, his students had also been using forcePAD. He assured the other participants in the workshop of his students having no problems in learning to use both of these GUIs and he did not consider there to be any need of integrating the two programs. He remarked that by using a program such as pointSketch2D he could now easily compare physical (tangible) models their digital counterparts which he considered to be a considerable advantage pedagogically. He noted also that pointSketch2D made it possible to create and analyse many different versions of the same basic structure. He also regarded the file management possibilities as being an important aspect of pointSketch2D due to its allowing users to save their models and open them again later to continue working with them. He said that some of his students had difficulties, when modelling in the physics mode, in adding an appropriate number of supports to a model, its not being obvious to them that they should add the minimum number of supports needed in order to make the model statically determined, their tending instead to add too many supports, making the structure statically indeterminate, an approach that could lead to unnecessary tensions being introduced.

*Participant 7* considered the graphics in the pointSketch programs to be appropriately designed and easily comprehensible. He approved of the notion of analysing stability in a structural mechanical context as a means of investigating the structural behaviour involved. He especially liked the idea of pointSketch2D telling the user how many additional bars would be needed to achieve stability, without the user being told exactly where these bars should be added, which he regarded as a nice pedagogical twist. He pointed out, however, the fact that bars appear extended when one analyses the visualisations of mechanisms, an extension which in reality is impossible since there are no forces present that can produce it[64].

*Participant 8* was involved in the same series of lectures as participant 6 and thus had a certain familiarity with pointSketch2D. He as well found it to be a limitation that there was no possibility in pointSketch2D of analysing frames. He experienced problems too in comparing different ‘structures’ in terms of the

[64] This is a problem related to the fact that for purposes of legibility, pointSketch2D enlarges the deformations that the computational engine has found to occur, its being assumed that only small deformations in the ‘structure’ take place. When the deformations are scaled upwards to a high degree in this way, they fail to show the true deformation pattern. The scaling factor can be adjusted either upwards or downwards by the user.

distribution and magnitude of the normal forces within them. He tended to perceive the magnitude of the forces as being constant, regardless of the geometry and the load case, which he felt made it unnecessarily difficult to compare 'structures' that differed from each other.[65]

The following are a number of additional comments and suggestions that were made in the workshop:

- It was suggested that the possibilities that were available to the user be made more obvious by improved design of the "specify magnitude" function. None of the participants had intuitively understood how this function the program had could be used.
- It was felt that it would be a welcome addition to the palette of building blocks for some element other than a bar being included that carries load through tension only (such as a wire for example)
- In general, pointSkecth2D was perceived as being a visualisation rather than a computational program.
- The workshop participants felt that a version of pointSketch2D available for the Macintosh platform would be greatly appreciated. Currently, pointSketch2D/3D is only available for the Windows platform.
- Many of the participants indicated that they would like to see the sketch mode implemented in the pointSketch programs. They felt it would appeal very much to architects in general if the possibility for sketching on a background supplied by the user were provided.

## 7.4 Shareware

The fourth and final step in the evaluation process is that of the pointSketch programs being made available for download from a website[66] free of charge, and of further distribution of it being allowed, its thus becoming a *shareware*. Registration of the person download it is required, allowing the manager of the website to keep track of how the programs are spread throughout the world. Having a platform-independent program facilitates such a distribution process. Users providing feedback after use of the programs is encouraged and can be done by use of the website. The feedback will be reviewed continuously and if the principles involved are considered to be consistent with the spirit of pointSketch they are incorporated into the programs. One thus allows the community of users

[65] This confusion was resolved by an explanation being provided of the function that locks a given scale to a particular value.

[66] http://www.chalmers.se/arch/SV/forskning/pointsketch

of pointSketch to influence its further development. This helps ensure that it remains current.

# 8 Discussion

The need of a computer-based tool to support collaboration of architects and designers with engineers during the early stages of a design process was indicated. A general conception of such a tool based on experience gained in the Material and Design Studio at Chalmers, as reported in Chapter 2, as well as on a survey of various computer tools and ideas in this area presented in Chapter 3, was taken up in Chapter 4. Six characteristics of capabilities an appropriate computer tool of this sort should have were formulated as guiding principles for the conceptual design process. These were designated by the verbs *associate*, *sketch*, *build*, *change*, *read*, and *understand*.

The two computer programs, pointSketch2D and pointSketch3D, were created by use of development tools, as reported in Chapter 5. Their GUIs were designed, as reported in Chapter 6, to correspond to the characteristics just referred to. The visualisation modes plays a central role in both programs. The sketch mode provides the possibility of investigating and testing different shapes by use of *sketching*. In the appearance mode a sketch can then be visualised to show how the 'structure' in question would look in reality. In order to ensure the stability of the 'structure', a model of it needs to be *built* then in the physics mode by use of nodes, bars, and supports. *Changing* the magnitudes associated with the properties these building elements possess affects the structural behaviour concerned. In the action mode, the structural behaviour is *read* in pictures showing eigenmodes, normal forces, reaction forces and deformations. Switching between the different visualisation modes provides an *understanding* of how a change in the sketch alters the structural behaviour of the 'structure' to which it refers. The process as a whole of sketching, modelling and switching between modes allows the user to *associate* an abstract conception of a 'structure' with its tangible structural behaviour.

The pointSketch programs were evaluated finally in three separate steps, as dealt with in Chapter 7. In the following, a number of concluding reflections on the comments participants made in the evaluations they provided are presented, the order in which these are taken up being the same as used in the account given of the pointSketch approach in Chapter 4. This, in turn, is followed by a general discussion of the pointSketch approach, the contributions it can make to research in this area, and proposals for future work.

## 8.1 Comments from the evaluations

The overall responses of participants in the workshop as well as in the test panel meetings strongly indicate pointSketch2D to be an intuitive and pedagogical computer-based design aid. The possibility it provides for experimenting freely with the structural behaviour a design involved appeared to be regarded as the foremost characteristic the program. The perceived shortcomings that were noted, such as the indistinct animations of various mechanisms, are being worked on. The lack of either frame analysis or support for the Macintosh platform are limitations that were obvious from the start, but that had little or no influence on the evaluation of the programs in their present form. The fact that the architectural school lecturers who participated in the workshop, have since introduced pointSketch2D as a tool in their teaching gives evidence of pointSketch being a useful tool and having a place in the design environment that architectural education represents.

The form in which the programs were to be made available to users as a design aid was discussed thoroughly in the test panel meetings. A central decision made was to create four different visualisation modes, each of them facilitating the translation of an abstract conception of a 'structure' into one of tangible shape and form. In the sketch mode the architect or designer sketches the first tentative lines required for visualising the rough idea of the shape he/she has in mind. The appearance mode represents in turn, the attempt to imagine the current 'structure' as it would look in reality. The physics mode visualises the 'structure' as an engineer would define it, as a mechanical model of a load-carrying system. The action mode, finally shows the structural actions that take place as a function of the stiffness distribution and the loads that are or can be involved, each of these actions being expressed in two different ways – as internal forces and as deformations. By switching between the four modes described, both a single user and a group of collaborating users can investigate the interplay between the basic idea that has been conceived, the appearance the 'structure' in question would have, structural model of it, and the structural actions that occur, obtaining answers to questions of the "what if" type. Neither the sketch nor the appearance mode is implemented as yet in pointSketch2D, but both are described as being part of the pointSketch approach. Of these two modes, the sketch mode was the one that was primarily requested during the workshop. It is reasonable to assume that in the evaluations, the physics mode was considered to provide sufficient freedom and opportunity for creating 'structures', no far-reaching conclusions regarding this can be drawn before all the visualisation modes has been implemented in the programs and been tried out for the uses intended.

During the self-evaluation process and the test panel meetings, the manner in which the information in each visualisation mode should best be presented in order to for legibility to be maintained was discussed. This lead to graphical settings being provided aimed at enabling the user to adjust the appearance of the building elements provided in the physics mode as well as of the visualised deformations and forces dealt with in the action mode. During the workshop, first-time users misinterpreted to a certain extent the functions related to the scaling of deformation and colour, in particular the locking of these scales. This indicates that the functions in question should be made more legible and comprehensible, that additional information should be provided in connection with them, and that they should possibly be moved to a higher level of precision in the GUI. On the other hand, when the participants were provided an explanation of these functions by the developer, they regarded them as being logical, despite the initial confusion they had. Certain other functions as well were misinterpreted initially but were subsequently understood with little effort when explained. This raises the question of whether it is reasonable for at least a limited learning process to be required in connection with the use of a program such as pointSketch2D. A goal in GUI design generally is that no process of learning be required of the user. In connection with pointSketch2D, it was adjudged however, that the GUI presenting the deformation and colour scale would lose too much of its intended usefulness if it were simplified to that point. At the same time, use of majority of the functions available in the pointSketch2D GUI did not need to be explained to users more than once, indicating these functions to be intuitive in character, or at least to be easy to learn. The workshop in which questions of this sort arose lasted roughly an hour, during which time the all of them appeared to have learned the basics of pointSketch2D reasonably well.

The evaluation and development process the thesis work involved contributed to reflections on the current state of the programs as well as being an inspiration for new ideas and introduction of additional features. One feature that emerged from the development process was that of linking the model used in the physics mode with CALFEM. CALFEM is an addition to the high-level computer language of MATLAB.[67,68] It is a tool for creating numerical algorithms used here

[67] CALFEM 2006
[68] MATLAB 2006

for analysing structural behaviour by use of the Finite Element Method (FEM). Such a link implemented in pointSketch provides the user the possibility of comparing the graphical representation of the computational model with its numerical representation. A link of this sort can be regarded either as a fifth mode or as a submode to the physics mode. When merged, pointSketch and CALFEM provide the user the possibility of following the idea of the shape of a 'structure' from the first tentative graphical sketch of it to its concrete numerical representation.

Particular emphasis was directed in the first two steps of the evaluation process, at enabling pointSketch to provide computational results appropriate to variations allowed in the precision of input. The reasons for this were twofold, first that of accommodating users who differed in their knowledge of structural mechanics, and secondly providing appropriate computational results in the early stages of the design process, when search for and comparison between 'structures' in a more global sense is the primary goal, and in later stages, in which there is more intensive concern with structural details. The idea of users being able to set their own level of precision in the program was conceived of during the test panel meetings. In the physics mode, at the lowest level of precision the user is provided with only basic building elements and actions, whereas at the highest levels of precision complete and detailed settings concerning the material and geometric properties of the 'structure' are provided. The user is able at any time to adjust the level of precision employed in the use of the programs. This promotes a learning process.

Since pointSketch is aimed at supporting the architect or designer in the conceptual stages of the design process sketching was assigned particular importance both in the evaluation process and generally. Thus sketching had a strong impact on how the interaction between the programs and the user were conceived for the different visualisation modes. Sketching in the sketch mode resembles traditional sketching, that of pencil or pen and paper. In the appearance mode, sketching consists of making a picture of the 'structure' as it would appear in real life. The transition from the appearance or the sketch mode to the physics mode entails the creation of nodes and of system lines used to represent the 'structure'. This is a process that can likewise be carried out in a sketch-like manner.

During the workshop considerable use was made of the program functions that allowed participants to study the eigenmodes related to different mechanisms. There were also a few of the participants who made use of the possibility of investigating the canonical stiffness of the stable 'structures' that were present and of the eigenmode attached to that particular canonical stiffness. When pointSketch2D identifies an unstable ‘structure’, the action mode automatically shows the structural movements that are connected with the mechanisms that are obtained. These modes of movement are the only tangible results obtained for the ‘structure’ in question. In contrast, when a ‘structure’ is stable, pointSketch has the capacity of producing eigenmodes, canonical stiffness, normal forces, and deformations. For such a ‘structure’ the default visualisation setting is that of showing the normal forces that are produced in response to the loads that are applied, and not to the deformations involved. This is a conscious choice that was made, aimed at indicating the deformations associated with a stable ‘structure’ to be very small compared with those associated with an unstable ‘structure’. The eigenmodes obtained for stable ‘structures’ are adjudged to be of only secondary interest, except for one case, that of an unloaded stable structure. The choices of default settings to be used can be discussed. An observation made in working with architects is that deformations can be easier to grasp than forces. If deformations are used as the default visualisation, the degree of representation of structural action that appears is about the same for unstable ‘structures’ generally as for either loaded or unloaded ones that are stable. Forces have been chosen here nevertheless as the default visualisation to be shown for stable ‘structures’, with the aim of focusing attention not on the movements shown by an stable ‘structure’ but rather on its the task of being a carrier of loads, which can be expressed by visualisation of the pattern of tension and compression produced.

It was felt that pointSketch3D was a welcome supplement to pointSketch2D, although there was consensus of its being necessary that pointSketch3D be given the same simplistic intuitiveness as pointSketch2D in order for it to be regarded a worthy supplement to it. Difficulties in managing the GUI attached to pointSketch3D were pointed out during the workshop. After the evaluation had taken place, efforts were made to improve the program’s three-dimensional navigation, the aspect of this GUI that was considered to be its major weak point. At the same time, it is more difficult generally to navigate in a three-dimensional than in a two-dimensional environment. The same tools, the traditional mouse and keyboard, were chosen here for navigation in both two and three dimensions. This was decided upon because of the majority of users being equipped with these tools only. A general comment made was that pointSketch3D should never be regarded as a replacement for pointSketch2D. Sketching in two dimensions has just as much importance from a pedagogical point of view as sketching in the three dimensions.

In the workshop, the use of different computer tools was compared. Some of the participants had experience in using ForcePAD prior to the workshop. None of them spoke of the two pointSketch programs competing with ForcePAD in any way. Rather, they were regarded as complementing each other in terms of sketching, modelling and visualisation. The other programs the participants had been using in their lecturers during the past year were similar to commercial CAE programs. They considered pointSketch2D to be completely different in character than these and to fit their educational aims better than the CAE tools that were considered did.

In the workshop, pointSketch2D was regarded as a welcome addition to the computer aids available for lecturers in teaching structural design for architects. Many felt that pointSketch2D should be given international recognition. However, the ease of using pointSketch2D led to certain concern on the part of the participants that users might become over-confident of it as a self-instructional tool. It was felt there should be clear incentives for students to reflect while using a computer program of this sort that provides quick responses to inputs. There was considered to be a risk of important information being overlooked when a user clicks back and forth quickly between different visualisations. The view nevertheless was that the structure of pointSketch encourages the user to consider different approaches to solving a given task. One example was that of pictures of different mechanisms guiding the user to the location where additional bars (system lines) should be placed. In using the computer programs users should study and take careful note of how the different clicks they make on the GUI can affect the 'structure' they are analysing, since users need to be able to make their own design decisions on the basis of information they glean from the visualisations.

## 8.2 Contributions to the research field and future work

The use of structural analyses is generally concerned with a limited goal within engineering design, that of computing the strength of proposed structures in order to decide whether they will fail structurally or will hold. In pointSketch the elements found in most such analyses are used but are formulated and presented in a different way than usual, the possibilities and solutions that structural mechanics provides being organised as a GUI in a manner aimed at supporting the early stages of the design process of the architect. This can be seen as promoting a new and more cognizant attitude towards structural mechanics and how it supports design work on a conceptual level. I consider this aspect of pointSketch to be the most important contribution of the present research.

Collaborative design work of all kinds involves an element of education. If a participant in a collective design process possesses the knowledge required for the design task at hand, this knowledge needs to be conveyed in a manner making it readily comprehensible to all those involved so that appropriate design decisions can be made. The pointSketch approach is designed with this in mind. As a welcome side effect, the pointSketch programs can also be of use in variety of educational settings, such as those of courses for architects and engineers.

Both additional functions that can be incorporated into the pointSketch programs and further development of the programs in other respects should be aimed at. Efforts might be made initially to achieve the following goals:

- Full implementation of all the visualisation modes.
- Implementation of all the precision levels.
- Providing support for frame analysis.
- Providing support for computations within the second-order theory involved.
- Porting of pointSketch to the Linux and Macintosh platforms.
- Evaluation of the pointSketch programs in a professional setting.

# References


*Active Statics* (2006), MIT Media Lab, Massachusetts Institute of Technology, http://acg.media.mit.edu/people/simong/statics/Start.html

*Adobe Photoshop* (2006), Adobe Systems Inc., http://www.adobe.com

Anderson, O. (2003) *Nya designmetoder för möbelbranschen – Re-design av karmstol KS 263, Publication 2003:2.* Göteborg, Studio of Material and Design, School of Architecture, Chalmers University of Technology. (in Swedish)

*Ansys* (2006), Ansys Inc., http://www.ansys.com

*Autodesk AutoCAD* (2006), Autodesk Inc., http://www.autodesk.com

*Autodesk VIZ* (2006), Autodesk Inc., http://www.autodesk.com

Balmond C. (2002) *Informal*. London, Prestel Publishing

Billger M., Heldal I., Stahre B. and Renström K. (2004) "Perception of Colour and Space in Virtual reality: A Comparison Between a Real Room and Virtual Reality Models", in *Human Vision and Electronic Imaging IX,* edited by Bernice E. Rogowitz, Thrasyvoulos N. Pappas, Proceedings of SPIE-IS&T Electronic Imaging, SPIE Vol. 5291, pages 90-98.

Bobert R. and Lund M. (1991) "Kraft och rum". *Arkitektur*, vol. 5.

*Bridge Builder* (2006), Alex Austin, http://www.bridgebuilder-game.com

*Bridge Designer* (2006), Whiting School of Engineering, John Hopkins University, http://www.jhu.edu/virtlab/bridge-designer

*Bridge IT* (2006), Chronic Logic, http://www.chroniclogic.com

*CALFEM : a finite element toolbox to MATLAB : version 3.3* (2006), Division of Structural Mechanics, Lund Institute of Technology, http://www.byggmek.lth.se/Calfem/index.html

*Dr. Frame3D* (2006), Dr. Software, http://www.drsoftware-home.com

*Fast Light Toolkit (FLTK)* (2006), Spitzak, B., http://www.fltk.org

*ForcePAD* (2006), Division of Structural Mechanics, Lund University of Technology, http://www.byggmek.lth.se/resources/forcepad/forcepad.htm

*Interactive Visualisation Framework – ivf++* (2006), Division of Structural Mechanics, http://www.gorkon.byggmek.lth.se/ivfweb

Larsson P., Västfjäll D., Olsson P. and Kleiner M. (2005) “When What You Hear is What You See: Presence and Auditory-Visual Integration in Virtual Environments” (Paper D). *Virtually Hearing, Seeing, and Being: Room Acoustics, Presence, and Audiovisual Environments*. Göteborg, Department of Civil and Environmental Engineering, Chalmers University of Technology.

*Linux* (2006), http://www.linux.org

*MATLAB : the language of technical computing* (2006), Natick, Mass., The Math Works Incorporated, http://www.mathworks.com

Mathiasen H. and Reitzel E. (1999) *Grundtræk af bærende konstruktioner i arkitekturen*. København, Kunstakademiets Arkitektskoles Førlag.

Meriam J.L. and L.G. Kraige. (1993) *Engineering mechanics. Vol. 1, Statics, Third Edition.* New York, John Wiley & Sons, Inc.

*Microsoft PowerPoint* (2006), Microsoft Corporation, http://www.microsoft.com/office/powerpoint

*Microsoft Visual Studio .NET* (2006), Microsoft Corporation, http://msdn.microsoft.com/vstudio

*Microsoft Windows* (2006), Microsoft Corporation, http://www.microsoft.com/windows

*Microsoft Word* (2006), Microsoft Corporation, http://www.microsoft.com/office/word

*Multiframe* (2006), Formation Design Systems, http://www.formsys.com

*Newmat C++ matrix library* (2006), Davis R., http://www.robertnz.net/nm_intro.htm

Nielsen J. *Designing Web Usability* (2000). Indianapolis, Ind., New Riders.

*NX* (2006), UGS, http://www.ugs.com

Olsson K.-G. (2005) *Strukturmekanik & Arkitektur – Om strukturmekanisk förståelse i gestaltningsprocessen.* Göteborg, Department of Architecture, Chalmers University of Technology. (in Swedish)

Olsson, P. (2003) *FEM and Scientific Visualisation as Tools in Furniture Design.* Göteborg, School of Architecture, Chalmers University of Technology.

Olsson P., Eriksson P. and Olsson K.-G. (2004) “Computer-supported Furniture Design at an Early Conceptual Stage”. *International Journal of Design Computing.* Volume 7.

Olsson P. and Olsson K.-G. (2003) *Applied Visualization of Structural Behaviour in Furniture Design*. Göteborg, Studio of Material and Design, School of Architecture, Chalmers University of Technology.

*OpenGL* (2006), http://www.opengl.org

*Pontifex* (2006), Chronic Logic, http://www.chroniclogic.com

Popovic, O. (2003) *Conceptual structural design : bridging the gap between architects and engineers*. London, Thomas Telford.

Preece J. et.al. (1994) *Human-Computer Interaction*. Wokingham, Addison-Wesley.

*Pro/ENGINEER* (2006), PTC, http://www.ptc.com

Saabye Ottosen, N. and Petersson, H. (1992) *Introduction to the finite element method.* New York, Prentice Hall.

Sandaker B. (2000) *Reflections on span and space : towards a theory of criticism of architectural structures.* Oslo, Arkitekthøgskolen.

Slater M. (2000) *A Note on Virtual Light Fields.* Department of Computer Science Research Note RN/00/26/ April 5th, University College London.

*SolidWorks* (2006), SolidWorks Corporation, http://www.solidworks.com

Stroustrup B. (2000) *The C++ programming language.* Boston, Addison-Wesley.

Wester T. (1984) *Structural Order in Space – the plate-lattice dualism*. Köpenhamn, Plate laboratory, Royal acad. of arts, School of architecture.

Zalewski W. and Allen E. (1997) *Shaping Structures: Statics*. New York, John Wiley & Sons, Inc.

# Appendices

The appendices are provided on the appended CD.

I. pointSketch2D version 1.0
II. pointSketch3D BETA version
III. Olsson P., Eriksson P. and Olsson K.-G. *Computer-supported Furniture Design at an Early Conceptual Stage*
Also available on the web:
http://www.arch.usyd.edu.au/kcdc/journal/vol7/papers/olsson/index.htm
IV. Olsson P. and Olsson K.-G. *Applied Visualisation of Structural Behaviour in Furniture Design*
Also available on the web:
http://www.arch.chalmers.se/staff/pierre.olsson/dissertation/applied_vis.pdf