AVOCAAD EXERCISES
EXPANDING ON INTERACTIVE OPERATIONS

AUTHORS
Martin Sprekelsen
neue Rezepte
Föhrenstr 13a
D- 82194 Gröbenzell
martin@neueRezepte.de

Gernot Pittioni
Ingenieurbüro Pittioni
Pippinger Strasse 102
D- 81247 München
pittioni@pittioni.de
AVOCAAD EXERCISES
EXPANDING ON INTERACTIVE OPERATIONS

Contents
1. Introduction
2. Advantages of interactive web-pages for AVOCAAD
3. Technical survey
4. An example
5. Outlook
6. References

1. Introduction
The web is a vital element for realising the AVOCAAD project. The web's features and functionality present a splendid platform. The following will discuss multiple advantageous options available through this new media as they relate to the AVOCAAD project.

All data are permanently available on a central server, accessible to an unlimited number of clients anytime, anywhere in the world. Clients access the centrally stored information and work locally with the material, thus using the common server-to-client publishing set-up. Dynamic database functions available to the general user are able to control various aspects of data flow. This procedure is used by the AVOCAAD-web-system.

Recent developments in the web are going to enable an even more sophisticated use, thus widening the range of application. The online material may present interactive properties, meaning that the user is able to observe changes of processes in relation to the influence he actually exerts on the material within his subsystem.

We will focus on this material in our paper, exploring the possible impact on AVOCAAD-exercises.

2. Advantages of interactive web-sites for AVOCAAD
The AVOCAAD-exercises offer a professional aid to students and architects looking for training material. The structure was shown by help of some examples 1998 by Verbeke [1]. In general, AVOCAAD instructions are provided in a problem-solution or knowledge-transfer format.

As in all systematically performing processes, in special cases, e.g. structural design, the need arises to show or observe the solution process. The learning process can be made more efficient by interactively demonstrating the behaviour of systematic or procedural exercises.

Thus the user experiences solution possibilities by determining a varying range of control options. Solutions will rarely be stringent as to the way they are achieved; this is even true for engineering problems. For example, one can improve the energy-behaviour of a building to a given extent by insulating either the roof or the walls. The process of developing optimal potential solutions would be explained most clearly to the user by allowing him or her to
alter specific parameters and observe the corresponding changes in numeric process behaviours and outcomes.
In addition there are cases where the common "description – goal – exercise"-scheme does not work efficiently, for there are a number of sub-solutions, each exerting their own specific influence on the remaining solution process. Imagine an exercise with a result consisting of different steps, and a non-linear structure; the goal defined as the correlation between these working steps. A meaningful instructional exercise exhibiting this level of complexity will be highly dependent on the interactive functionality of the web page. Thus an individualised solution can be achieved by guiding the user interactively. Finally an exercise might present a general concept within more relaxed limits which enable the user to work with details according to the individual level of knowledge and/or interest.

3. Technical survey
The tools to provide web-sites with interactive properties are currently in development. As web-technologies (for example the document language HTML) are in a constant process of evolution, new developments arise, thus expanding possibilities. One problem in this evolution-process is, that usage patterns of different browsers vary widely; those familiar with web-browsers will confirm that the support of the latest web-features is not guaranteed with all browsers. Sometimes browser graphic presentation differs substantially from the intention of the web-site-developer. This difficulty will grow even more as tool sophistication increases. Therefore it is generally highly recommended to use only those language-definitions which are commonly accepted by different browsers; in particular this refers to language definitions standardised by the www-consortium 1. In the context of these restrictions, the following options to develop a web page exist currently:

• The Document Language Enhancements DHTML (dynamic-HTML), DOM (document object model) and XML (extensible markup language)
• The Script language ECMA-Script, the former JAVA-Script
• JAVA, the Web-programming language
All these systems have in common that they are supported by the two most widely used browsers, Netscape Communicator (Navigator) and Microsoft Internet Explorer (at this time only beta versions of DOM and XML exist). The Document Language Enhancements aim at basic definitions of how to handle a certain page on-screen. The web-page is handled as an object with properties and functions – not as mere document presentation. Thus the web-page offers the functionality to be handled like a program, so mouse-operations and similar input can be interpreted interactively. User-initiated change of parameters is easily accessible on-screen.

JavaScript exists as a script-language, which was originally developed by Netscape. In the meantime it is officially acknowledged as ECMA Script by the W3C-Consortium. It was integrated (more or less completely) by Microsoft into its Internet Explorer as JScript.

1 An international commission, also called W3C. Since 1994 it oversees the development of the www-standards; its decisions are used as guidelines by developers all over the world.
ECMA Script resp. JavaScript resp. JScript offer an object-oriented structure – more or less completely – and thus handle

- mathematical operations
- string-operations
- parametrical processes
- user in- and output and
- browser-related functions.

A conceptual weakness of the Script-language is its inability to change web page source code interactively, once the page is built-up. That means that we have very restricted output-properties, except for HTML-forms, graphic objects, or browser-owned elements. In connection with DHTML or DOM this weakness is omitted, the web-page is entirely controllable.

Script-languages have the benefit that they can be written directly into the document-source-code. This makes programming easy and processing-time is enhanced considerably, as no additional modules or libraries have to be loaded.

In certain cases it will be inevitable to use a genuine programming-language as Java, to avoid the mentioned weaknesses. Java is a completely object-oriented language, which is closely related to the programming language C++. The former – Java - was developed by SUN Microsystems, Inc., for use in the world-wide-web.

Currently Java source code is freely accessible, which guarantees its further existence and development. By choosing Java the programmer has definitely all properties of a highly sophisticated programming-language at his disposal.

The transmission of lean precompiled source-code over the net ensures satisfactory performance on all systems in most cases, assuming availability of a runtime-version of a compiler. This independence of the platform in actual use is both an advantage and a disadvantage, because the applets - the loaded and processed Java-programmes – considerably slow the performance of the web-page.

Assuming those “teething troubles” will be omitted in the near future by use of optimised compilers and by general progress, the programming of highly-sophisticated interactive web-pages will become more and more attractive. This will enhance the added value of AVOCAAD-exercises considerably.

4. An Example

In order to demonstrate the theoretical background we will proceed with a demonstration of the features by selecting a very simple topic.

In Figure 1 we can see an interactive Exercise, where a student can visually experience the influences on the energy consumption of residential houses by changing different parameters, like the thickness of the walls or the size of a window or the conductivity value of the insulation. It’s a great help in understanding the energy behaviour by interactively testing the correlation of parameters.
Figure 1: AVOCAAD Exercise sheet: “Energy consumption of residential houses” (G.Pittioni).

Another interactive exercise which is going to be shown in the course of this conference is the facility management diagram included in the paper of Oswald, Pittioni [2]. These exercises are written in JavaScript, which controls the in- and output of the text-fields and naturally also performs the correct mathematical calculations.

5. Outlook
Looking at the future, it is likely that AVOCAAD-exercises will increasingly take advantage of interactive web-page features. The combination of server system functionality with client system functionality, as discussed above, will likely become highly attractive.
The guiding of the user - during the processing of his exercises - and the further processing of the results by the central server-system will increasingly operate together. For the end-user web-page interactive functionality will become ever more important.

6. References