

# Making Web-Based Training More Efficient

**Florian Albrecht, Thomas Tiller**

FAST e.V.  
Forschungsinstitut für Angewandte  
Softwaretechnologie  
Arabellastr. 17  
80925 München  
Germany  
{florian,tti}@fast.de

**Nora Koch**

Ludwig-Maximilians-Universität  
München  
Institut für Informatik  
Oettingenstr 67  
80538 München  
Germany  
kochn@informatik.uni-muenchen.de

## Abstract

Web-based learning is very advantageous for teachers and learners, specifically if the system adapts itself to the user's personal needs. Yet, many of the currently available systems suffer from the limitations of the underlying Web paradigm. We implemented the SmexWeb<sup>1</sup> framework for Web-based learning systems. This framework increases interactivity between learner and teaching system in two particular ways. The system transfers more valuable information from the client to the server by partially bypassing HTTP as underlying protocol. Furthermore we use a technique we call "Passive Navigation" to give the system a chance to take over control if the user remains inactive for a certain amount of time. To verify our framework we built an interactive course, a so-called SmexWeb application, that was used by first year computer science minor students.

## 1 Introduction

The WWW as a platform for training bears many positive aspects. Delivery of the training material is simple. Systems residing on a central server are easy to maintain and update. Learners can choose time and location of the lessons individually. Delivering content on basis of standard technologies guarantees platform independence and straightforward access to the learning material.

Web-based training and learning systems can be enhanced immensely by making them adaptive, i.e. letting the system estimate the personal characteristics of the user and adapting content and presentation to them. This is one of the reasons why many Web-based adaptive systems have been deployed in the last few years [Brusilovsky & Eklund 98, Kay & Kummerfield 94, Nakabayashi et al. 97, Weber & Specht 97].

Usually data transfer from the system to the client and vice versa is implemented using the hypertext transfer protocol (HTTP) and HTML forms. HTTP is a stateless protocol where any kind of data transfer has to be initiated by the client. When trying to build classical Intelligent Teaching Systems (ITS) on the Web two problems arise:

1. The system only gets information about the learner if he sends a request. This is not enough to simulate a real life tutor-learner situation. A tutor, for instance, is not only interested in a solution to a task the learner himself considers correct. The tutor acquires much more information by watching the learner's actions during the process of working out solutions. Misconceptions are detected earlier and appropriate reactions to them make learning more motivating.

---

<sup>1</sup> <http://pst1.pst.informatik.uni-muenchen.de:8080/>

2. The system is not able to start any kind of communication to the learner without him explicitly asking for it. A tutor in contrast to that will do this very often. If, for instance, he loses the learner's attention he will immediately react by changing the level of explanation or trying to find out the reason for the learner's lack of interest.

Vassileva solved these problem in her DCG system [Vassileva 97]. When a learner starts a session, a small application (Java-applet) with all the information available about the user's characteristics along with a plan for the path through the session is downloaded to the client. The applet observes and guides the learner. In case the plan seems inappropriate a new one is requested from the server. This approach transfers main parts of the "intelligence" of a system from the server to the client side. A higher interactivity and hence a closer adaptation of the presented material to the learner's needs is reached for the cost of transmitting more data than necessary, namely the whole inference mechanism for estimating the user's characteristics.

We implemented a framework, called SmexWeb [Albrecht, Koch & Tiller 99], for building adaptive Web-based training systems that increases interactivity without transferring more data from the server to the client than necessary (see Section 3).

## **2 SmexWeb, a framework for building Web-based adaptive hypermedia training systems**

The SmexWeb (Student Modelled Exercising on the Web) server is a collection of concrete and abstract Java classes, easily usable on any computer connected to the Internet. The classes resemble the architecture of a classical intelligent tutoring system. Amongst others they include a tutor component and a user model.

The user model component offers structure and functionality for maintaining the assumptions the system has about a user. A built-in and easy modifiable rule-based inference mechanism allows for straightforward estimation of the user's characteristics contingent upon his actions.

The framework is based on a hypertext structure that integrates all adaptive techniques [Brusilovsky 96]. The hyperspace represents the domain model. Any course is a structured collection of nodes and links between them. The nodes contain certain concepts of the domain-knowledge. An author builds a course with a closed knowledge corpus on top of this structure by creating adaptive pages using DHTML and JavaScript. These adaptive pages consist of page fragments of any granularity that are hidden or displayed dependent on the user model state. The author then links those pages and provides all necessary attributes for each link, allowing the system to provide the appropriate adaptive navigational support on basis of the user model information.

The framework supports the author of a course by taking care of the following tasks:

- Network communication: The author can state which information should be transferred at which time. He does not have to care about how the framework implements this.
- The inference mechanism: The author defines a set of simple rules (condition/action pairs) for adjusting the user model. Conditions can be any kind and combination of current and previously recorded user input. Actions are modifications of the values in the user model. The framework itself takes care of when and how to apply these rules effectively.
- The adaptation mechanism: Based on the information provided by the author, the framework automatically adapts material and navigation according to the current user model state.

The SmexWeb framework hooks up to a standard Web server running on the same machine and provides all functionality necessary to implement Web-based adaptive hypermedia training systems. It gives authors with only little technical knowledge the possibility to make their expert knowledge in any domain accessible to learners.

For a more detailed discussion of the framework's architecture and the underlying hypertext structure see [Albrecht 98, Tiller 98].

Courses based upon the SmexWeb framework are easy to use for a learner as nothing more than a standard Web browser is required.

### 3 Enriching the Web paradigm

As introduced above there are two fundamental issues when trying to simulate a tutor-learner situation on the Web. On the one hand, the tutor - in our case the training system - wants to "watch" the student while he is working on an exercise. For a Web-based system this means that more data than just the solutions to exercises have to be made available to the system's inference mechanisms. On the other hand a tutor might want to make a suggestion to the learner if he seems to need help. Again, for a Web-based training system this means that the system wants to present a different page of the hyperspace to the learner. Relying solely on HTTP, this would not be possible. The SmexWeb framework addresses these issues by enriching the basic Web paradigm in two ways: Direct Interaction and Passive Navigation.

#### 3.1 Direct Interaction

The pages of a training session are enhanced HTML pages that use JavaScript programs to adapt themselves to the user. Those pages may contain interactive exercises that are implemented using Java applets. When being displayed the applet opens a network connection to the SmexWeb server. While a learner solves an exercise a lot of potentially interesting information is available. It consists of all the atomic actions of a learner, which would not be recorded or transmitted when using HTML forms for giving exercises to the learner.

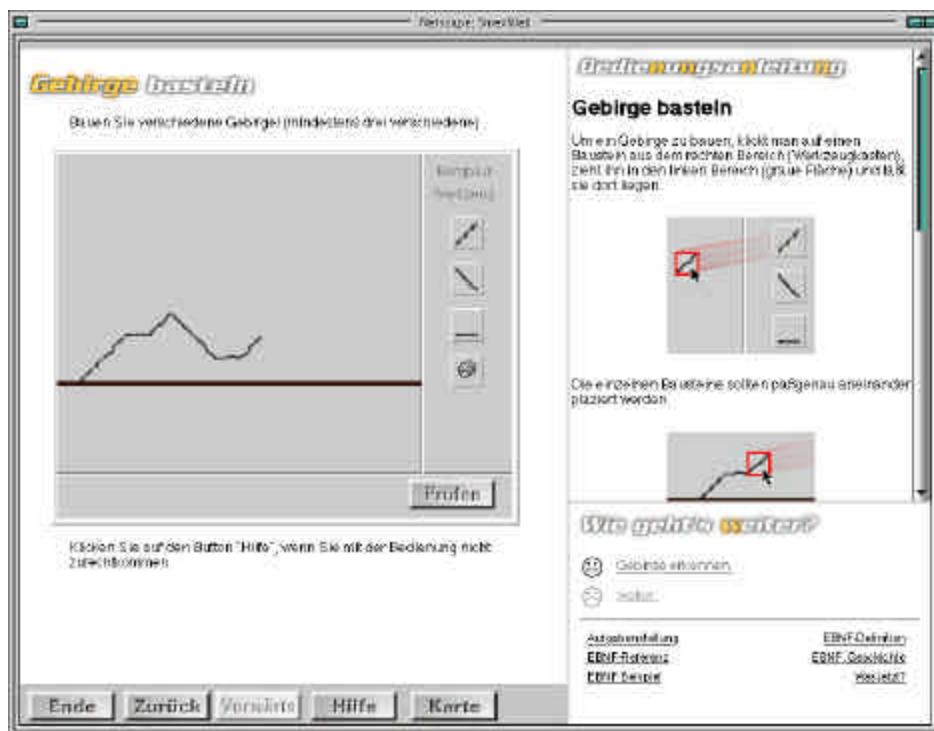


Figure 1: An interactive exercise in the EBNF application implemented with SmexWeb

The applet immediately transfers the data to the server using a protocol defined in the framework. The SmexWeb server uses inference rules to evaluate the incoming data and to adjust the user model accordingly. This fine grained observation allows the system to obtain a more precise picture of the learner than other Web based adaptive training systems.

The client directly interacts with the server and exchanges information when necessary - not just upon request by the learner. When the exercise is completed the connection shuts down along with the applet.

### 3.2 Passive Navigation

As written above, in certain situations the system needs a chance to present a new page to the learner, without him asking for it. Often time is an important aspect of real life learning situations. Part of the teacher's strategy is the decision, when to present certain material to the learner.

Common adaptive hypermedia applications have the ability to decide what to present to the user. The navigation paradigm for these systems only allows the presentation of new pages when the user requests them by following a link. Hence these systems have no chance to take actions without the user explicitly asking for them.

SmexWeb solves this shortcoming of classical hypermedia applications by giving the system the possibility to take control over navigation. The system decides when the user is being navigated to a different page. In opposite to the active navigation process of the user, being navigated by the system is called "Passive Navigation".

SmexWeb implements Passive Navigation using the Javascript "setTimeout" method and the JavaScript "location" attribute of a HTML document. For each node in the hypertext structure an author states under which conditions the learner should be transferred to a certain page. When a learner requests a page, SmexWeb reads the information from the hypertext structure and integrates a small JavaScript program into the page to be transmitted. This program checks the conditions and resets the URL of the page to be displayed in the browser.

Passive Navigation allows the shift of the locus of control [Vassileva & Wasson 96] from the user to the system if necessary. This is an important step for hypermedia systems towards intelligent teaching systems.

## 4 SmexWeb Application

To verify our framework we built a course on the topic EBNF (enhanced Backus-Naur formalism) for first year computer science minor students. For this heterogeneous group neither experience in working with computers and interactive systems nor the ability of understanding and applying abstract formalisms could be taken as prerequisite.

Figure 2 shows the structure of the EBNF course. An initial interview gives the system information about the student to set up the user model. In the next steps the student interactively solves exercises with the support of the system. In the last step the system tests the acquired knowledge and skills by asking the student to solve a similar exercise without support. Pages containing context sensitive help and reference material about the domain concepts are available for request at any time.

The system supports the learning process by adapting the content of the pages and using adaptive navigation techniques [De Bra 99, Brusilovsky 96]. The techniques implemented in the SmexWeb application for the EBNF course are link sorting, link annotation, link hiding and passive navigation.

- *Link sorting* orders the list of links according to the estimate significance for the user.

- *Link annotation* is used in two forms to establish a ranking of recommendations for the links. Three different icons as well as a fading anchor text are used to show relevance of the links.
- *Link hiding* is chosen by SmexWeb in case the type of exercise is not considered appropriated to the student knowledge and skills.
- *Passive navigation* allows the system to control navigation in case the student does not interact with the system for a predefined period of time.

In a classroom test the majority of students uttered the impression that they had acquired new knowledge and skills. Yet it is unclear if the positive results are due to the better observation of the learners on the basis of the increased interaction. Ongoing and more precise tests will give further insight.

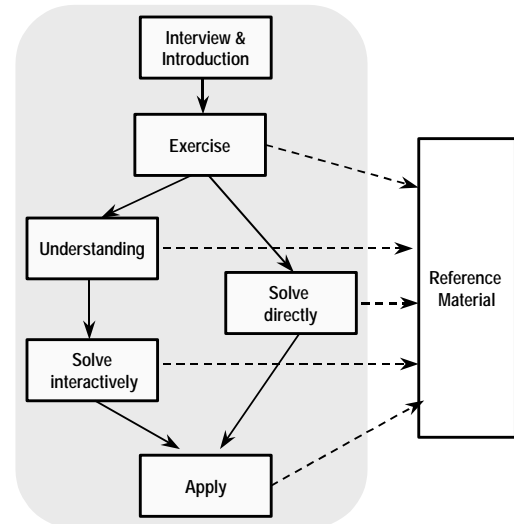


Figure 2: Structure of the EBNF course

## 5 Conclusions and Future Work

SmexWeb is an adaptive Web-based learning system that enables a higher amount of user-system interactivity than most Web-based learning environments. We described how SmexWeb enriches the Web paradigm through *direct interaction* and *passive navigation*. The first one is performed through interactive exercises implemented as Java-applets and allows to record more incoming data of the user. This data is evaluated and used to update the user model. The second one, passive navigation, gives the system the possibility to take control over navigation when predefined conditions take place. These conditions have to be established by the author.

Current and future work include:

- The implementation of a graphical authoring tool on top of the SmexWeb framework for building Web-based adaptive hypermedia training systems.
- A generic protocol for the direct communication of applets on the client side and the user model component on the server.
- A discussion forum to allow collaborative learning.

## References

- Albrecht F. (1998). SmexWeb: Ein adaptives Web-basiertes Übungssystem. Diplomarbeit, Ludwig-Maximilians-Universität München, <http://pst1.pst.informatik.uni-muenchen.de:8080/DA/>
- Albrecht F., Koch N. & Tiller T. (1999) SmexWeb: An adaptive Web-based Hypermedia Teaching System. International Journal of Continuing Engineering and Life-Long Learning, in print.
- Brusilovsky P. (1996). Adaptive Hypermedia: An Attempt to Analyze and Generalize. Proceedings of Multimedia, Hypermedia and Virtual Reality, P. Brusilovsky, P. Kommers, N. Streitz (Eds.). Lecture Notes in Computer Science 1077, 288-304, Springer Verlag.
- Brusilovsky P., & Eklund, J. (1998). A study of User Model Based Link Annotation in Educational Hypermedia. Journal of Universal Computer Science, 4 (4), 429-448, Springer Science Online.

- De Bra P. (1999). Design Issues in Adaptive Web-Site Development. Proceedings of the 2<sup>nd</sup> Workshop on adaptive systems and User Modeling on the WWW.
- Kay J. & Kummerfield R. J. (1994). An Individualised Course for the C Programming Language. Proceedings of the Second International WWW Conference '94, Mosaic and the Web.
- Nakabayashi K. et al. (1997). Architecture of an Intelligent Tutoring System on the WWW. Proceedings of the Eighth World Conference of the AIED Society, Kobe, Japan, 18-22 August 1997
- Tiller T. (1998). Eine Hypertext-Struktur für abgeschlossene adaptive Systeme. Diplomarbeit, Ludwig-Maximilians-Universität München.  
<http://pst1.pst.informatik.uni-muenchen.de:8080/DA/>
- Vassileva J. & Wasson B. (1996). Instructional Planning Approaches: from Tutoring towards Free Learning. Proceedings of EuroAIED'96, 1-8, Lisbon, Portugal.
- Vassileva, J. (1997). Dynamic Course Generation on the WWW. Proceedings of Workshop Intelligent Educational Systems on the World Wide Web in AI-ED 97: Eighth World Conference on Artificial Intelligence in Education.
- Weber G. & Specht M. (1997). User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems. User Modeling: Proceedings of the Sixth International Conference, UM'97, Jameson A., Paris C. and Tasso C. (Eds.). Springer Verlag Wien.