

In: *Web-based Instruction*, B. H. Khan (ed) Educational Technology Publications, Englewood Cliffs, New Jersey. - pp. 255-261.

Electronic Textbooks on WWW: from Static Hypertext to Interactivity and Adaptivity

Peter Brusilovsky
School of Computer Science
Carnegie Mellon University
Pittsburgh, USA
E-mail: plb@cs.cmu.edu

Elmar Schwarz and Gerhard Weber

Department of Psychology, University of Trier
Trier, Germany
E-mail: {schwarz | weber }@cogpsy.uni-trier.de

Abstract. The paper discusses the problems of developing interactive and adaptive "textbooks" on WWW. We present the system ELM-ART which is an intelligent interactive textbook to support learning programming in LISP. ELM-ART demonstrates how interactivity and adaptivity can be implemented in WWW electronic textbooks. Since adaptivity is the most innovative feature of ELM-ART we center our presentation around adaptive features of ELM-ART and its implementation.

Introduction

World Wide Web (WWW) opens new ways of learning for many people. Now, educational programs and learning materials installed and supported in one place can be used by thousands of students from all over the world. However, most of the existing educational WWW applications use the simplest solutions and are much more weak and restricted than existing 'on-site' educational systems and tools. For many designers, the ideal form of educational WWW material seems to be a static electronic copy of a regular textbook: chapter by chapter, page by page, picture by picture. Such "static electronic textbooks" have two major shortcomings. First, they are not interactive enough, i.e., students can only passively read the educational materials. Second, they are non-adaptive, i.e., students with different abilities, knowledge, and background get the same educational material in the same form. At the same time, interactivity is very important for any educational application as a prerequisite of active learning. Adaptivity is especially important for educational applications on WWW which are expected to be used by very different classes of users without assistance of a human teacher (who usually can provide adaptivity in a normal classroom).

This paper discusses the problems of developing interactive and adaptive "textbooks" on WWW. We present the system ELM-ART (Brusilovsky, Schwarz & Weber, 1996) which is an intelligent interactive textbook to support learning programming in LISP. ELM-ART demonstrates how interactivity and adaptivity can be implemented in WWW electronic textbooks (ET).

ELM-ART: interactive intelligent LISP textbook on WWW

ELM-ART (ELM Adaptive Remote Tutor) is a WWW-based system to support learning programming in LISP. ELM-ART is based on ELM-PE (Weber & Möllenberg, 1995) an Intelligent Learning Environment that supports example-based programming, intelligent analysis of problem solutions, and advanced testing and debugging facilities. ELM-ART provides all the course materials (presentations of new concepts, test, examples, and problems) in hypermedia form. It can be considered as an on-line adaptive ET with an integrated problem solving environment (we call it I³ textbook, or intelligent interactive integrated textbook). ELM-ART differs from simple WWW ET in two major aspects.

First, ELM-ART is interactive: all examples and problems (which are important components of any textbook) in ELM-ART are not just a static text as in other textbooks, but "live experience": students can play with examples and solve problems interactively using WWW fill-out forms. The fields of these forms are used to provide several kinds of input, such as the text of a problem solution, and the buttons are used to get several kinds of help and feedback from the system while examining an example or working at a problem:

- to ask the system to evaluate or to step a LISP expression which can contain any LISP functions learned in the introductory course,
- to request the most relevant case from the personal history (example or problem solution),
- to request an intelligent diagnosis of the prepared solution.

Second, ELM-ART is adaptive. It "knows" the material it presents to a particular student and knows the current student level of knowledge about LISP. As a result, ELM-ART can help users to understand a particular section of learning material and supports them in learning and navigating the course material.

Next sections present interactive and adaptive features of ELM-ART in more details.

On-line course material

The course material in ELM-ART is provided on WWW in hypertext form. It consists of two main components: the textbook and the reference manual. The textbook (which is an on-line version of the normal printed LISP textbook used in the courses during the last years) is hierarchically structured into units of different level: chapters, sections, and subsections. Each unit can be presented to the student as a WWW "page" which shows the content of this unit (text and pictures) and various kinds of links from this unit to related elements of the course. All problem solving examples and problems are presented on separate "interactive" pages which use the possibilities of WWW fill-out forms. The reference manual provides the reference access to the course material. Each "page" of the manual contains a brief explanation of one of the course concepts (which can be a special term, a LISP function, a data type, or a higher-level concept) and the links to the related course units and to related manual pages. Thus, the manual in ELM-ART performs also the roles of glossary and index of traditional textbooks.

Two kinds of links are used in the course: normal hierarchy links and content-based links. The hierarchy links connect a higher level unit (like a chapter) with all its sub-units (like sections in this chapter) and in other direction each unit with its parent unit. Content-based links can be provided by ELM-ART because the system "knows" what is presented on each page of the course material, i.e., which concepts are introduced, presented, or summarized in each subsection, which concepts are demonstrated by each example, and which concepts are required to solve each problem. As a result, the system can provide cross-references between textbook and manual, for example, provide references from each textbook page

(including example and problem pages) to corresponding manual pages for each involved concept. The links within the manual are also content-based links. The system "knows" the pedagogical structure of the LISP domain and the relationships between various concepts. All these relationships are shown as links on the manual pages.

Adaptive navigation support

ELM-ART provides many more opportunities for browsing the course materials than traditional on-line textbooks. The negative side of it is that there is a higher risk for the student to get lost in this complex hyperspace. To support the student navigating through the course, the system uses *adaptive annotation* an efficient adaptive hypermedia technique (Brusilovsky, 1996). Adaptive annotation means that the system uses visual cues (icons, fonts, colors) to show the type and the educational state of each link. Using the student model, ELM-ART can distinguish several educational states for each page of material (including problem, example, and manual pages): the content of the page can be known to the student, ready to be learned, or not ready to be learned (the latter example means that some prerequisite knowledge is not yet learned). The icon and the font of each link presented to the student are computed dynamically from the individual student model (Fig. 1). They always inform the student about the type and the educational state of the node behind the link. Traffic lights are used as a metaphor. Red (italic typeface) means not ready to be learned, green (bold) means ready and recommended (part of the learning goal), yellow means ready but not recommended.



Fig. 1. Example of adaptive annotation of links at the beginning of the course.

The metaphor is traffic lights. Red (italic typeface) means not ready to be learned, green (bold) means ready and recommended, yellow means ready but not recommended.

Adaptive Prerequisite-Based Help

The system knowledge about the course material comprises knowledge about what the prerequisite concepts are for any page of course material including example and problem pages. As a rule, the prerequisites are not shown directly. But, when the student has problems with understanding some explanation or example or solving a problem, he or she can request help (using a special button) and, as an answer to help request, the system presents an adaptively generated help page which shows annotated links to all pages where the prerequisite knowledge is presented.

Interactive Examples

An example in a textbook does not help very much if the user does not pay enough attention to it. That's

why we want to provide an easy way for students to play with examples by integrating an enhanced evaluator into ELM-ART. This access is currently provided by transforming the text of any example into a link (Fig. 2). When the user activates this link, the system loads the evaluator page, executes the example, and responds with the result or an error message (Fig. 3). Employing the evaluator, the user can play with the given examples by modifying some pieces of code or by looking to step by step evaluation leading to a deeper understanding of the internal code evaluation strategy in LISP (Fig. 3). Any occurring run-time error is explained by ELM-ART in more detail and more suited to novices than common LISP interpreters do.

```

eine Liste erwartet und als Wert deren zweites Element liefert. Stellen
eine solche Funktion, die wir ZWEITES nennen. Dann liefert:

(ZWEITES ' (BROT KAFFEE MILCH ZUCKER))
KAFFEE

(ZWEITES ' (REST (BROT KAFFEE MILCH ZUCKER)))

(BROT KAFFEE MILCH ZUCKER)

(ZWEITES ' ()).
NIL

```

Fig. 2. Part of a page from the textbook concerning the problem "ZWEITES." Using the links, the examples are put into an interactive evaluator page where they can be modified and re-executed.

```

? (ZWEITES ' (BROT KAFFEE MILCH ZUCKER))
(ZWEITES ' (BROT KAFFEE MILCH ZUCKER))
' (BROT KAFFEE MILCH ZUCKER)
' (BROT KAFFEE MILCH ZUCKER) --> (BROT KAFFEE MILCH ZUCKER)
LIST --> (BROT KAFFEE MILCH ZUCKER)
  (REST (FIRST LIST))
  (FIRST LIST)
  LIST --> (BROT KAFFEE MILCH ZUCKER)
  (FIRST LIST) --> BROT
  (REST (FIRST LIST)) --> *ERROR*
(ZWEITES ' (BROT KAFFEE MILCH ZUCKER)) --> *ERROR*
> Der Funktionsaufruf
> (REST BROT)
> ist fehlerhaft.
>
> Das REST-Konstrukt erwartet eine Liste als Argument.
>
> Das Argument ist aber ein Atom
> BROT
? (FIRST (REST ' (BROT KAFFEE MILCH ZUCKER)))

```

Fig. 3: Evaluation of the first example of Fig. 2.

Interactive and intelligent problem solving support

Any page describing a programming problem supports the user interactively in different ways. All problem pages provide an interface to the stepper to let the user to test the developed solution. If testing of a developed solution reports that it is still erroneous and the user is unable to find the error, he or she can try two forms of intelligent problem solving support: request a helpful example or invoke an intelligent program analyzer.

As ELM-ART is a system that supports example-based programming, it encourages the students to re-use the code of previously analyzed examples when solving a new problem. The hypermedia form of the course and, especially, similarity links between examples help the student to find the relevant example from his or her previous experience. An important feature of ELM-ART is that the system can predict the student way of solving a particular problem and find the most relevant example from the individual learning history. This kind of problem solving support is very important for students who have problems with finding relevant examples. Answering the help request ("show example" on Figure 4), ELM-ART selects the most helpful examples, sorts them according to their relevance, and presents them to the student as an ordered list of hypertext links. The most relevant example is always presented first, but, if the student is not happy with this example for some reasons, he or she can try the second and the following suggested examples.

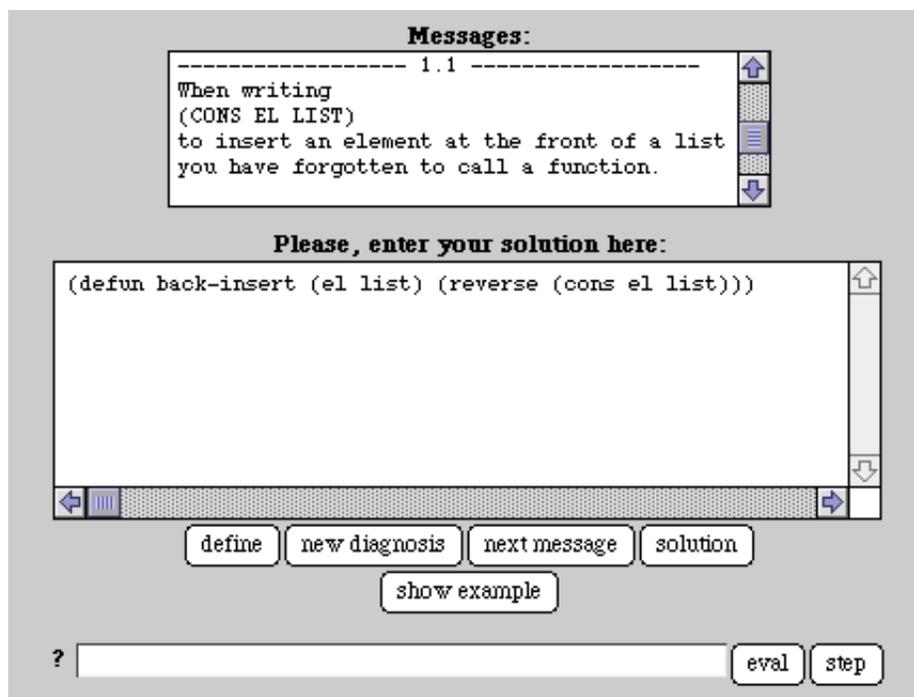


Fig. 4. Form-based interface for problem solving support

If the student failed to complete the solution of the problem, or if the student can not find an error which was reported by the testing component, he or she can ask the system to diagnose the code of the solution in its current state. As an answer, the system provides a sequence of help messages with more and more detailed explanation of an error (Fig. 4). The sequence starts with a very vague hint what is wrong and ends with a code-level suggestion of how to correct the error or how to complete the solution. In many cases, the student can understand where the error is or what can be the next step from the very first messages and do not need more explanations. The solution can be corrected or completed, checked again, and so forth. The student can use this kind of help as many times as required to solve the problem correctly. In this context, the possibility to provide the code-level suggestion is a very important feature of ELM-ART as a distance learning system. It ensures that all students will finally solve the problem without the help of a human teacher.

Implementation of ELM-ART

The WWW implementation of ELM-ART[1] is based on the Common LISP Hypermedia Server CL-HTTP (Mallery, 1994). CL-HTTP[2] is a fully featured HTTP server completely implemented in Common LISP. Since the original ELM-PE system was also implemented in Common LISP, CL-HTTP

appears to be an optimal platform for our purposes. CL-HTTP offers a Common Gateway Interface^[3] to handle incoming URLs from all over the world via the Internet. To enable the server to respond to a particular URL, this URL has to be associated to a response function implemented in LISP. Answering an incoming request, the server recognizes a URL, calls an associated function, and returns an HTML page which is generated by this function. With such an architecture, CL-HTTP is a very flexible and powerful tool for implementing intelligent applications on WWW. Since a LISP function is called to handle the request, any interactive or intelligent tool written in LISP can be connected to WWW with the help of CL-HTTP.

Conclusions and future work

The system ELM-ART described in this paper is an example of an interactive and adaptive ET implemented on WWW. ELM-ART provides remote access to hypermedia-structured learning material which includes explanations, tests, examples, and problems. Unlike traditional electronic textbooks, ELM-ART provides the learner with an adaptive interface, intelligent navigation support, adaptive help, and possibilities to play with examples, to solve problems, and to get intelligent problem-solving support. ELM-ART integrates the features of an electronic textbook, of a learning environment, and of an intelligent tutoring system.

We hope that methods we used in ELM-ART to represent knowledge about the subject, the interface, and the user, as well as adaptation techniques based in this knowledge can be re-used for developing ET for different subjects on WWW (e.g., our team plans to use the same framework for implementing a WWW textbook on statistics). We also want to investigate some advanced ways (i.e., JAVA) of adding interactivity to our ET. A first project is to port the adaptive LISP structure editor from ELM-PE to JAVA.

Acknowledgments

Part of this work is supported by a Grant from "Alexander von Humboldt Foundation" to the first author and by a Grant from "Stiftung Rheinland-Pfalz für Innovation" to the third author.

References

- Brusilovsky, P. (1996) Methods and techniques of adaptive hypermedia. In P. Brusilovsky and J. Vassileva (eds.), Spec. Iss. on Adaptive Hypertext and Hypermedia, *User Modeling and User-Adapted Interaction* 6 (2-3), 87-129.
- Brusilovsky, P., Schwarz, E., and Weber, G. (1996) ELM-ART: An intelligent tutoring system on World Wide Web. In: C. Frasson, G. Gauthier and A. Lesgold (eds.) *Intelligent Tutoring Systems*. Lecture Notes in Computer Science, Vol. 1086, (Proceedings of Third International Conference on Intelligent Tutoring Systems, ITS-96, Montreal, June 12-14, 1996) Berlin: Springer Verlag, pp. 261-269.
- Mallery, J. C. (1994) A Common LISP hypermedia server. In: Proceedings of the First International Conference on the World-Wide Web, May 25, 1994. Weber, G. and Möllenberg, A. (1995) ELM-Programming-Environment: A Tutoring System for LISP Beginners. In: K. F. Wender, F. Schmalhofer and H.-D. Böcker (eds.): *Cognition and Computer Programming*. Norwood, NJ: Ablex, pp. 373-408.

[1] ELM-ART URL: <http://www.psychologie.uni-trier.de:8000/elmart>

[2] <http://www.ai.mit.edu/projects/iiip/doc/cl-http/home-page.html>

[\[3\]http://hoohoo.ncsa.uiuc.edu/cgi/](http://hoohoo.ncsa.uiuc.edu/cgi/)