

# **ELM-ART: An Intelligent Tutoring System on World Wide Web**

Peter Brusilovsky, Elmar Schwarz, and Gerhard Weber

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

**Abstract:** Making ITS available on the World Wide Web (WWW) is a way to integrate the flexibility and intelligence of ITS with world-wide availability of WWW applications. This paper discusses the problems of developing WWW-available ITS and, in particular, the problem of porting existing ITS to a WWW platform. We present the system ELM-ART which is a WWW-based ITS to support learning programming in Lisp. ELM-ART demonstrates how several known ITS technologies can be implemented in WWW context.

## **1 ITS Technologies and WWW Context**

WWW opens new ways of learning for many people. However, most of the existing educational WWW applications use simplest solutions and are much more weak and restricted than existing 'on-site' educational systems and tools. In particular, most WWW educational systems do not use powerful ITS technologies. A promising direction of research is to port these technologies to a WWW platform, thus joining the flexibility and intelligence of ITS with world-wide availability of WWW applications.

Most of traditional intelligent techniques applied in ITS can be roughly classified into three groups which we will name as technologies: curriculum sequencing, interactive problem solving support, and intelligent analysis of student solutions. All these technologies are aimed at supporting the "intelligent" duties of the human teacher which can not be supported by traditional non-intelligent tutoring systems. Curriculum sequencing and intelligent analysis of student solutions are the oldest and best-studied technologies in the domain of ITS. Most ITS developed during the first 10 years of ITS history belong to these groups. The technology of interactive problem solving support is a newer one, but it is more "intelligent" and supportive (it helps the student in the most difficult part of the learning process and provides the most valuable support for the teacher in the classroom). It is not surprising that it became a dominating technology during the last 15 years. The WWW context changes the attitudes to traditional ITS techniques [Brusilovsky, 1995]. For example, interactive problem solving support currently seems to be a less suitable technology for WWW-based ITS. Vice versa, the two older technologies seem to be very usable and helpful

in the WWW context. Intelligent analysis of solutions needs only one interaction between browser and server for a complete solution. It can provide intelligent feedback and perform student modeling when interactive problem solving support is impossible. Curriculum sequencing becomes very important to guide the student through the hyperspace of available information. In addition to traditional ITS technologies, some of more recent (and much less used) ITS technologies become important. Two examples are adaptive hypermedia [Beaumont & Brusilovsky, 1995] and example-based problem solving [Weber, 1995].

This paper discusses the problems of developing WWW-based ITS and, in particular, the problem of porting existing ITS to a WWW platform. We present the system ELM-ART which is an ITS to support learning programming in Lisp. ELM-ART is developed on the base of the system ELM-PE [Weber & Möllenberg, 1994] specially to be used on WWW. The presentation is centered around intelligent features of ELM-ART (a number of interesting non-intelligent features of ELM-ART are described elsewhere [Schwarz, Brusilovsky & Weber, 1996]). The goal of the paper is to demonstrate how several known ITS technologies can be implemented on WWW and what has to be added when porting a traditional ITS to WWW.

## **2 ELM-ART**

ELM-ART (ELM Adaptive Remote Tutor) is a WWW based ITS to support learning programming in Lisp. ELM-ART is based on ELM-PE [Weber & Möllenberg, 1994], an Intelligent Learning Environment that support example-based programming, intelligent analysis of problem solutions, and advanced testing and debugging facilities. For several years, ELM-PE was used in introductory Lisp courses at the University of Trier. The course materials were presented to students in lecture form (complemented with printed material) and ELM-PE was used to practice the obtained knowledge by problem-solving. ELM-ART, which is expected to be used for distance learning, provides on-line both the course materials and the problem solving support.

ELM-ART can be considered as an on-line intelligent textbook with an integrated problem solving environment (we call it I<sup>3</sup> textbook, or intelligent interactive integrated textbook). It provides all the course materials (presentations of new concepts, test, examples, and problems) in hypermedia form. ELM-ART differs from simple WWW "hyperbooks" in two major aspects. First, ELM-ART "knows" the material it presents to the students and supports them in learning and navigating the course material. Second, all examples and problems (which are important components of any textbook) in ELM-ART are not just a text as in other textbooks, but "live experience". Using ELM-ART, the student can investigate all examples and solve all problems on-line. For any textbook problem, ELM-ART provides the student with almost the same level of intelligent problem solving support as the original ELM-PE. All these features of ELM-ART are presented below in more detail.

## 2.1 On-line Course Material

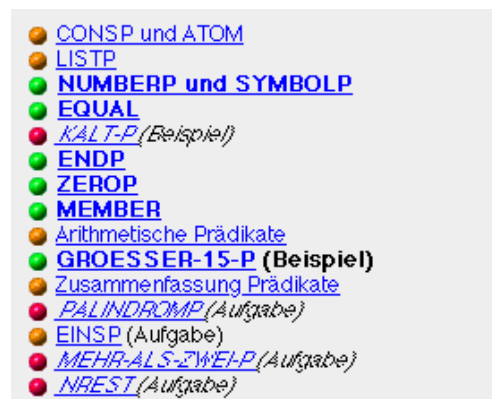
The course material in ELM-ART is provided on WWW in hypertext form. It consists of two main components: the textbook and the reference manual. The textbook (which is an on-line version of the normal printed Lisp textbook used in the courses during the last years) is hierarchically structured into units of different level: chapters, sections, and subsections. Each of these units 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 and the links to the related course units and to related manual pages. Thus, the manual in ELM-ART performs also the role 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 subunits (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 references from each textbook page (including example and problem pages) to corresponding manual pages for each involved concept. Vice versa, from each manual page describing a concept (for example, a Lisp-function) the system can provide references to all textbook pages where this concept is introduced, explained, or summarized, to all example pages where it is demonstrated, and to all problem pages which can be used to master this 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. Finally, the system "knows" the structure of any problem solving example. Thus, it can provide "similarity" links from each example to other most similar examples. This feature is based on the EBR method [Weber, 1995].

## 2.2 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 two adaptive hypermedia techniques - adaptive annotation and adaptive sorting of links. *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 (an individual permanent student model is maintained for each registered student), 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 case 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 (Figure 1). They always inform the student about the type and the educational state of the node behind the link. *Adaptive sorting* is used to present similarity links between cases. Since the system can measure the similarity between each two cases, it can also sort all cases related to the current one according to the similarity values. Links are presented in sorted order - the most relevant first - so the student always knows what the most similar cases are.



**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.

### 2.3 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, they can be shown to the students in the following two cases. First, when a student enters a page which is not yet ready to be learned, the system warns the student that this material has unlearned prerequisites and shows additional links to textbook and manual pages where the unlearned prerequisite concepts are presented. This feature adds intelligent help and implicit guidance to hypermedia freedom of navigation. Second, 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 will show the links to all pages where the prerequisite knowledge is presented.

## 2.4 Intelligent Problem Solving Support

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 2), 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.

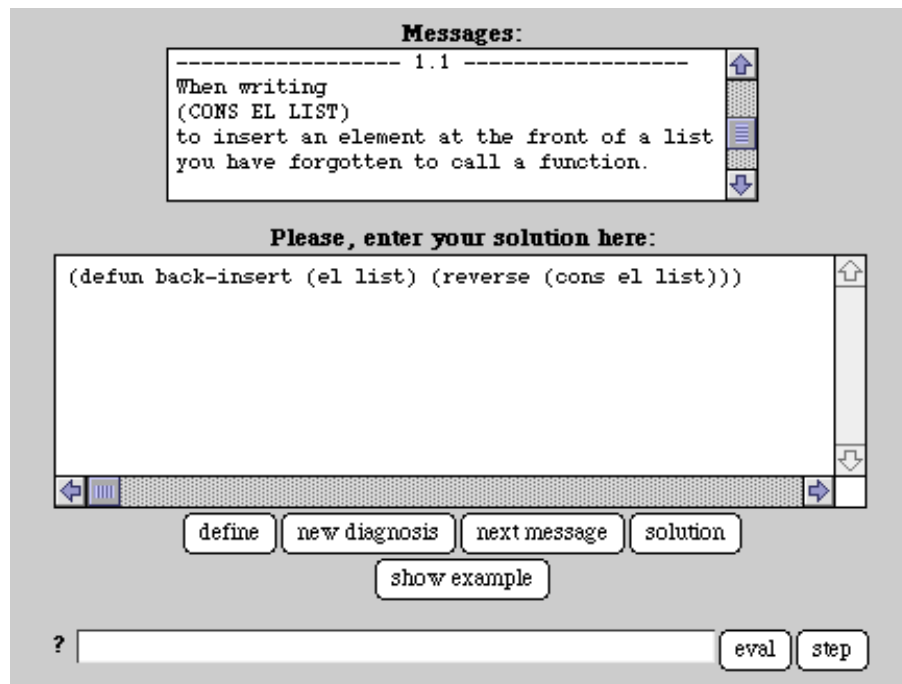


Fig. 2. Form-based interface for problem solving support

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 (Figure 2). 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.

### **3 Implementation of ELM-ART**

#### **3.1 Knowledge**

The key to the intelligent behavior performance of ELM-ART is the knowledge about the subject domain and about the student which is represented in the system in several forms. The major part of the ELM-ART knowledge base consists of the knowledge about problem solving in Lisp which is represented as a network of concepts, plans, and rules. Most of this knowledge, as well as tools to work with it were inherited by ELM-ART from its mother, ELM-PE [Weber, 1996].

To support all intelligent techniques that are used to work with the course material we have enhanced ELM-ART knowledge base with the conceptual knowledge about Lisp and with the knowledge about the course. We have applied an approach described in [Brusilovsky, 1992] to represent this knowledge. The core of this knowledge is the Lisp conceptual network (LCN) which represents all important concepts used in the course and all relationships between them. The kinds of represented concepts include several kinds of Lisp objects (functions, data objects, data types) and several kinds of high level concepts related with Lisp and programming. Each represented concept has a corresponding page in the ELM-ART reference manual. Two main types of relationships in LCN are the traditional "part-of" and "is-a". The prerequisite relationships between concepts are not represented directly, but the system is able to compute them from part-of and is-a relationships using several heuristics. To represent the conceptual knowledge of individual students the system uses a traditional weighted overlay student model. In particular, this model is used by ELM-ART to compute the current educational state for each concept that are shown to the student by adaptive visual cues.

To represent the knowledge about the course, all terminal units of the learning material (explanations, examples, and problems) were indexed by LCN concepts. It means that for each unit we have provided a list of concepts related with this unit (we

call this list *spectrum* of the unit). For each involved concept, the spectrum of the unit represents also the kind of the relationships or the role of the concept. For example, a concept can be *introduced*, *presented*, *summarized* in an explanation unit, or be a *prerequisite* concept for this unit.

### 3.2 Platform

The WWW implementation of ELM-ART<sup>1</sup> is based on the Common Lisp Hypermedia Server CL-HTTP [Mallery, 1994]. CL-HTTP<sup>2</sup> 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<sup>3</sup> 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. The received URL and enclosed form values that may contain an arbitrary amount of incoming data are submitted as function parameters. The function generates an HTML page as an adaptive response. To do that, the function can use a special set of HTML-generating functions. 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.

### 3.3 Functionality

*Adapting Pages.* An important feature of the system is that all pages presented to the user are generated adaptively on the fly when the user requests them. To generate pages, the system uses the text of the course which is stored as an annotated HTML file, and knowledge about the structure of the course. When assembling a page of the course, ELM-ART extracts the text of the requested unit from the HTML file and generates the rest of the page (header, footer, hierarchy links and content based links) from the knowledge base. The situation with reference manual pages is even more flexible, because for most of them not only links, but also the content itself is generated from the knowledge base. With this approach, all adaptive features of ELM-ART presented above such as additional headers for not-ready-to-be-learned pages or adaptive annotation of links according to their educational state can be easily implemented. An information about methods applied for adaptive annotation of links can be found elsewhere [Beaumont & Brusilovsky, 1995].

---

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

<sup>2</sup><http://www.ai.mit.edu/projects/iip/doc/cl-http/home-page.html>

<sup>3</sup><http://hoohoo.ncsa.uiuc.edu/cgi/>

*Problem solving support.* To integrate various forms of problem solving support, we provide a form-based interface by which the user can send a LISP expression to ELM-ART for evaluation or send a problem solution for analysis. A page with a problem to solve is implemented as a HTML fill-out-form<sup>4</sup> with slots to be filled with LISP code and several buttons to request different feedback from the system (Figure 2). For example, the slot "solution" has to be filled with the code of the problem solution and the associated buttons are used for defining, testing, and diagnosing the inserted solution. When the user "pushes" one of these buttons, a HTTP request with an attached content of the form is sent to the server. The server calls the responsible LISP function to process the solution. The processing functions provide the interface to different components of ELM-ART inherited from ELM-PE (evaluator, tester, diagnoser, etc.): they prepare the data for these components and transform their output into HTML form. The functionality of all these components is described elsewhere [Weber & Möllenberg, 1994]. The most advanced case here is an interface to the diagnoser. As we mentioned above, the diagnoser generates several levels of help on from hints to the code-level suggestions and this help should be presented to the user not in one shoot, but step by step. Currently, this functionality is implemented by presenting the diagnostic feedback in a scrollable window of the fill-out form so that only one message is visible at once. The user can browse these messages step by step until he or she understands the problem.

## 4 Conclusion

The system ELM-ART described in this paper is an example of an ITS implemented on WWW. ELM-ART provides remote access to a hypermedia-structured learning material which includes explanations, tests, examples, and problems. Unlike traditional electronic textbooks, ELM-ART provides the learner with intelligent navigation support and possibilities to play with examples, to solve the problems, and to get intelligent problem-solving support which usually can be provided only by a human teacher. ELM-ART integrates the features of an electronic textbook, of a learning environment, and of an intelligent tutoring system.

ELM-ART can also serve as a good example to discuss the problems of implementing and porting ITS systems on WWW. It demonstrates that some ITS technologies, such as intelligent analysis of solutions and example-based problem solving can be ported relatively easily to a WWW context. It shows the importance of on-line course material and the technologies of adaptive hypermedia and curriculum sequencing which help students to navigate through this material. From the practical side, ELM-ART demonstrates an effective way of implementing ITS on WWW using the Common Lisp Hypermedia Server CL-HTTP.

---

<sup>4</sup><http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/fill-out-forms/overview.html>



We consider the problems of implementing ITS on WWW as an important direction of ITS research. WWW can help ITS to move from laboratories to real classrooms. In the WWW context, ITS can be located on HTTP servers in the research laboratories which have powerful equipment to run intelligent systems and ITS professionals to support and update the systems. At the same time, learners from over the world can access these ITS using any WWW browser. These browsers require relatively cheap hardware and can run on almost any platform. It gives ITS really world-wide audience and unlimited source of data for testing and improving their functionality.

### **Acknowledgments**

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

### **References**

- Beaumont, I. and Brusilovsky, P. (1995) 'Educational applications of adaptive hypermedia'. In: K. Nordby, P. Helmersen, D. J. Gilmore and S. A. Arnesen (eds.): *Human-Computer Interaction*. London: Chapman & Hall, pp. 410-414.
- Brusilovsky, P. (1995) 'Intelligent tutoring systems for World-Wide Web'. *Proceedings of Third International WWW Conference*, Darmstadt, pp. 42-45.
- Brusilovsky, P. L. (1992) 'A framework for intelligent knowledge sequencing and task sequencing'. In: C. Frasson, G. Gauthier and G. I. McCalla (eds.): *Intelligent Tutoring Systems*. Lecture Notes in Computer Science, Berlin: Springer-Verlag, pp. 499-506.
- Mallery, J. C. (1994) 'A Common LISP hypermedia server'. *Proceedings of the First International Conference on the World-Wide Web*.
- Schwarz, E., Brusilovsky, P., and Weber, G. (1996) 'World-wide intelligent textbooks'. *Proceedings of ED-MEDIA'95 - World conference on educational multimedia and hypermedia*, Boston, MA.
- Weber, G. (1995) 'Examples and reminders in a case-based help system'. In: J.-P. Haton, M. Keane and M. Manago (eds.): *Topics in case-based reasoning. Selected Papers of the Second European Workshop, EWCBR-94, Chantilly, France*. Berlin: Springer-Verlag, pp. 165-177.
- Weber, G. (1996) 'Episodic learner modeling'. *Cognitive Science* .
- Weber, G. and Möllenberg, A. (1994) 'ELM-PE: A knowledge-based programming environment for learning LISP'. *Proceedings of ED-MEDIA '94*, Vancouver, Canada, pp. 557-562.