CrossMark

ELM-ART – An Interactive and Intelligent Web-Based Electronic Textbook

Gerhard Weber[1] · **Peter Brusilovsky**[2]

Abstract This paper present provides a broader view on ELM-ART, one of the first Web-based Intelligent Educational systems that offered a creative combination of two different paradigms - Intelligent Tutoring and Adaptive Hypermedia technologies. The unique dual nature of ELM-ART contributed to its long life and research impact and was a result of collaboration of two researchers with complementary ideas supported by talented students and innovative Web software. The authors present a brief account of this collaborative work and its outcomes. We start with explaining the "roots" of ELM-ART, explain the emergence of the "intelligent textbook" paradigm behind the system, and discuss the follow-up and the impact of the original project.

Keywords Intelligent tutoring system · Adaptive learning environment · Adaptive hypermedia · Learning programming

Introduction

The middle of 1990 could be considered the dawn of research on Web-based education. At that time, the World Wide Web was still young and attempts to use the Web as an educational platform focused on delivering simple static educational content. Yet the Web potential for delivering education anytime, anywhere, has been recognized by a number of researchers and practitioners who worked on different kinds of educational technologies in a pre-Web context. Between 1995 and 2000, a number of teams reported interesting examples of "porting" or re-implementing their earlier approaches and systems in the new Web context. Among this generation of early advanced Web-based educational systems was

✉ Peter Brusilovsky
    peterb@pitt.edu

[1] University of Education Freiburg, Freiburg im Breisgau, Germany

[2] University of Pittsburgh, Pittsburgh, PA, USA

ELM-ART, one of the first Web-based Intelligent Educational systems. Unlike other systems of the early generation that usually focused on one innovative technology, ELM-ART offered a creative combination of two different paradigms - Intelligent Tutoring and Adaptive Hypermedia technologies. The unique dual nature of ELM-ART contributed to its long life and research impact and was a result of collaboration of two researchers with complementary ideas supported by talented students and innovative Web software. This paper presents a brief account of this collaborative work and its outcomes. We start with explaining the "roots" of ELM-ART, explain the emergence of the "intelligent textbook" paradigm behind the system, and discuss the follow-up and the impact of the original project.

The Roots of ELM-ART

The collaboration that lead to ELM-ART started when Peter Brusilovsky, supported by an Alexander von Humboldt Foundation postdoctoral award, moved to the University of Trier to work with Karl Wender and Gerhard Weber on intelligent tutoring systems for programming.

At that time, the University of Trier was home to one of the most advanced projects in this field that was a product of several years of work. The Episodic Learner Model (ELM) was developed to simulate the student's progress in learning the programming language LISP (Weber 1996a). It was the basis to provide individual feedback to the learner on programming errors and non-optimal code. ELM combines a case-based and a rule-based approach to represent the student's knowledge and to diagnose students' solutions to programming tasks. This model was included with a programming environment ELM-PE. The development of ELM started on a XEROX Lisp-machine and on workstations in Braunschweig. The structure editor STRUEDI (Köhne and Weber 1987) served as a programming interface for the learner. In order to use ELM-PE in regular classes, it was ported to MCL (Macintosh Common Lisp) on Macintosh computers. The case-based learning model ELM allowed for example-based programming. The learner could ask the system for help by providing an example. With knowledge about the particular user, especially how he or she already solved similar tasks, the system was able to show the user examples from his or her own learning history (Weber 1996b) and to explain similarities between the example and the current task (Burow and Weber 1996).

In an evaluation study it could be shown that learning first steps of programming recursive functions in LISP was easier and faster with the support of ELM-PE compared to learning with the same environment, but without help from the cognitive diagnosis and feedback. The publication of these results received the best paper award at ED-Media in 1994 (Weber and Möllenberg 1994).

Though the results of learning to program recursive functions in LISP with ELM-PE were very encouraging, the limitation of this approach to working with specific computers in a university lab only proved to become a bottleneck.

In turn, Peter Brusilovsky brought to Trier his past experience with adaptive sequencing and adaptive navigation support. The work on adaptive navigation support was an extension of Peter's PhD thesis project ITEM/IP (Brusilovsky 1992), a

combination of an intelligent tutor, exploratory lab, and manual for introductory programming. The original ITEM/IP provided two distinctive ways to access learning content: system-driven (adaptive sequencing) and user-driven (hierarchical menus). Adaptive navigation support was suggested as a way to combine these approaches – i.e., provide user-driven access to learning content, yet support the user with adaptive guidance provided by the system. By the time of Peter's arrival to Trier, adaptive navigation support was explored in two systems: ITEM/PG (Brusilovsky and Zyryanov 1993) and ISIS-Tutor (Brusilovsky and Pesin 1994) and Peter's team in Moscow started to work on a new adaptive Web-based system for teaching UNIX (Brusilovsky 1995).

The Creation of ELM-ART

The idea of ELM-ART has grown from the attempts to bring together the benefits of adaptive navigation support explored in ITEM/PG and ISIS-Tutor and classic problem-solving support as offered by ELM-PE. It was not our first attempt to merge these technologies – before moving on with ELM-ART, we tried some other interesting ideas such as adaptive navigation support in the hyperspace of examples (Brusilovsky and Weber 1996). The main challenge on the way to ELM-ART was to find a paradigm for integrating two technologies that could seamlessly bring together the ideas and the experiences of both teams. Gradually, we arrived at a vision of an *interactive and intelligent Web-based programming textbook*. Just like traditional programming textbooks, our envisioned Web-based textbook should have sections of text with explanations of key concepts, program examples, and problems to solve. Yet, in contrast to traditional "static" books, all learning content in this book should be interactive, i.e., readers can answer questions in the text sections, explore program examples by running them with different parameters, interactively solve problems and receive system feedback. The textbook should also be intelligent in several senses, i.e., using several AI technologies to support readers – the intelligent program analysis, concept-based hyperspace organization with a set of domain concepts behind each content page, intelligent links between examples and problems, and, finally, adaptive navigation support that could guide students to the material that is most appropriate for the current knowledge. This vision provided a clear metaphor for the target system and allowed us to reuse earlier explored design features (concept-based hyperspace, adaptive navigation support) and assets (LISP textbook, program analysis and example recommendation). We also believed that the suggested paradigm offers a great demonstration of the new learning opportunities offered by the Web. A Web-based adaptive learning system allowed for classroom and platform independence. At that time, a system developed on a UNIX workstation could not easily be ported to a more common platform. A Web-based system could be accessed by many more learners compared to a platform-dependent system like ELM-PE.

An important tool for moving from the vision to the real system was CL-HTTP - a web server completely written in LISP and developed by John Mallery at MIT (Mallery 1994). As a MCL-version of CL-HTTP existed at that time, it was relatively easy to move the code of ELM to ELM-ART and to provide a rudimentary Web-based programming environment. We also ported the existing textbook of the introductory

LISP course to CL-HTTP. In the first version of ELM-ART, three lessons of our textbook introduced the necessary LISP functions and constructs that were needed to define recursive functions for manipulating lists. Lessons four to six provided recursive programming problems (the same as in ELM-PE) with the individual support of the ELM code diagnosis and feedback and storing the individual learner model. This first version led to the first publications of ELM-ART in 1996 (Brusilovsky et al. 1996a; Schwarz et al. 1996).

The next step was to add interactive questions to the textbook. Forced and multiple choice questions as well as questions that allowed for putting in results of computations (e.g., code evaluation) were evaluated by the ELM-ART server and provided further evidence on how well students already learned the concepts provided with the textbook. This allowed for generating the multi-layered overlay student model (Weber 1999). ELM-ART also provided an interface for the learner to inspect this "open" learner model and to modify it if necessary. This second version of ELM-ART was granted with the European Academic Software Award 1998 (EASA) in the section Computer Science.

Altogether, ELM-ART offered a good implementation of our original vision - an interactive, adaptive textbook with problem solving support. The main features of the first three lessons were adaptive navigation support and curriculum sequencing. In an overview of the sections and pages of the course, the traffic light metaphor was used to show students whether a page is ready to be worked on, already learned, or not ready to be worked on. The ELM-ART server computed the learning status at hand of previously visited pages and on whether exercises on these pages had been solved successfully. Additionally, this learning status was used to suggest to the learner the next best page to work on. In this way, individual curriculum sequencing was implemented. The main features of lessons four to six were example-based problem solving and cognitive diagnosis of solutions to programming problems, and detailed feedback and hints to improve the code. Missing from the former ELM-PE were the explanation of example similarities and the possibility to diagnose even incomplete solutions (because ELM-ART lacks a structure editor).

The final version of ELM-ART also offered a lot of features besides the originally envisioned textbook with interactive exercises and problem solving support. It used the open learner model that can be inspected and changed by the learner. It offered the possibility to set and change preferences by the learner as an adaptable feature. It included an interactive LISP interpreter that allowed for evaluating LISP expressions and for stepping through the evaluation of code. We also added several communication tools: a discussion list, a chat tool, and the ability to send text or questions to a human tutor.

Hidden from the students were some other important functions of ELM-ART such as detailed analysis of the server interaction. All interactions with ELM-ART can be recorded, especially which link or button was clicked and the current navigation status of the visited pages. Different experimental versions of ELM-ART were used to explore the impact of different types of adaptive navigation support and curriculum sequencing. An investigation of the problem solving support showed that students performed better in the final programming tasks (without help from the cognitive diagnosis) compared to ELM-PE. Especially, students with lower levels of previous programming knowledge profited from learning with ELM-ART. Data analysis was used to explore how learners

navigate through the course, how often experienced users and users less familiar with the Web used the adaptive sequencing functionality and/or stopped learning during the first lessons. These findings along with the presentations of the final version of ELM-ART have been published 2001 in the well-known JAIED paper (Weber and Brusilovsky 2001).

Contributions and Impact

An important contribution of ELM-ART was providing a model for the design of intelligent Web-based educational systems. The suggested paradigm of an interactive and intelligent textbook appeared to be a winning approach to combine intelligent tutoring and adaptive hypermedia technologies into a consistent design that matches the hypertext nature of the Web. The success of this paradigm impacted our work in two directions that could be described as *cloning* and *generalizing* the ELM-ART design.

First, we immediately used ELM-ART as a blueprint for the development of several similar systems for different domains and with different intelligent technologies. These systems were also implemented in CL-HTTP. The statistics tutor AST (Specht et al. 1997) offered the learner different types of learning materials based on a probabilistic learner model. The AST learning materials consisted of examples, playing grounds, and tests. Learners can explore the curriculum or receive recommendations by AST for each next best unit and learning material to work with depending on their previous interactions. The ADI agent (Schöch et al. 1998), a learning system about narcotics, was based on the AST architecture. In addition to the *advisor* sequencing feature that recommended the next best page to read, ADI provided the user with a graphical representation of the agent's model ("knowledge barometer") and offered him or her the chance to dispute any wrong assessment. Performing some knowledge tests was used as an approach to improve the model on a specific topic.

Second, the experience with ELM-ART and our attempts to clone its design helped us to realize that the relatively complicated and error-prone process of creating adaptive textbooks calls for an *authoring system* to support authors in creating ELM-ART-style systems for an arbitrary domain. As a result, in parallel with our work on other adaptive systems, we produced a sequence of authoring frameworks and systems for developing intelligent textbooks and courses. First in the sequence of authoring tools was InterBook - an attempt to distill domain-independent features of ELM-ART. The first version of InterBook appeared immediately after the first version of ELM-ART in 1996 (Brusilovsky et al. 1996b). Later InterBook was expanded and used as a platform for delivering several adaptive electronic books (Brusilovsky and Anderson 1998) and user studies of adaptive hypermedia (Brusilovsky and Eklund 1998). Two years later, Marcus Specht generalized his experiences with AST in another courseware authoring environment ACE (Specht and Oppermann 1998). Finally, in parallel with porting the large tutor on cognitive psychology "incops" (Klein et al. 1999) to the ELM-ART architecture, Gerhard Weber developed a more extensive authoring tool ART-Web. In the following years, NetCoach (Weber et al. 2002), the successor of ART-Web, matured to a complete learning environment that organized groups or classes of learners in academies. The authoring tool of NetCoach provides more types of questions (gap-filling tests, multi-gap tests, flash-based tests, rating scales, and the possibility to send

answers to questions to tutors by email). Authors can set learning goals and the system will automatically select all pages necessary to reach these goals. Additionally, NetCoach offers inspection tools for tutors (Weber et al. 2002) to observe the learning state of users. Since the beginning of the current millennium, NetCoach has been used for blended learning courses in classes on introductory psychology at the University of Education Freiburg. Thousands of teacher-training students have used these courses over the years.

In addition to its impact on our own future work, the design of ELM-ART and its direct descendant systems influenced numerous subsequent attempts to design Web-based educational systems. The idea of ELM-ART as a Web-based interactive textbook that uses adaptive hypermedia to bring personalization to the Web appeared to be very productive and encouraged many similarly minded projects. The older generation of these projects such as AHA! (De Bra et al. 2006), MetaLinks (Murray 2003), KBS-HyperBook (Henze and Nejdl 2001), INSPIRE (Papanikolaou et al. 2003), SEATS (Davidovic et al. 2003), ALICE (Kavcic 2004), iClass (O'Keeffe et al. 2006) are now also highly cited as well. Among more specific ideas, the traffic light metaphor for prerequisite-based adaptive navigation support in educational hypermedia became very popular. Started by ELM-ART and propagated by InterBook, AST, and NetCoach, this metaphor has later been used in numerous adaptive educational hypermedia systems and frameworks such as PAA-S (El-Khouly 2000), KBS-HyperBook (Henze and Nejdl 2001), AHA! (De Bra et al. 2006), SIGUE (Carmona et al. 2002), NavEx (Yudelson et al. 2004), BITS (Butz et al. 2004), and GALE (Smits and De Bra 2011).

It is interesting to note that the majority of projects cited above focused on the idea of Web-based adaptive hypermedia book expanding it to other domains and exploring different kinds of personalization, while not necessary focusing on the interactive aspect of the original ELM-ART book, i.e., the ability to run live examples, or receive feedback on problem solutions. At the same time, the interactive nature of Web-based textbook introduced by ELM-ART appeared to be important for computer science and specifically programming education, the original ELM-ART domain. ELM-ART and other adaptive hypermedia programming textbooks such as KBS-HyperBook (Henze and Nejdl 2001) and SEATS (Davidovic et al. 2003) together with early works on active textbook from the field of program visualization (Brown and Najork 1997) helped to form the paradigm of interactive Web-based electronic textbooks for computer science education (Unanue et al. 2002) that gradually emerged as one of the dominated approaches to deliver online content in this domain (Korhonen et al. 2013).

In a broader context, the very idea of implementing an adaptive and intelligent educational system as a Web-based system appeared to be influential by itself. Together with other early adaptive and intelligent Web-based educational systems (Brusilovsky and Peylo 2003), ELM-ART encouraged a whole generation of more recent systems including such popular systems as SQL-Tutor (Mitrovic 2003) and Assistments (Koedinger et al. 2010). While switching from a conventional system to a Web-based application appears to be a technical rather than conceptual innovation, the ability to access an education system anytime, from anywhere, and without installation struggle is very important. In case of ELM-ART it appeared to be important for educational effectiveness, ELM-ART surpassed its locally available predecessor ELM-PE despite of smaller size of its episodic learner model. For other systems like SQL-Tutor and Assistments, the Web-based nature was critical to bring innovative educational systems

to a much broader community of learners. In turn, it allowed to collect large volume of student data and that is vital for developing and assessing new ideas. The process of "Web migration", however, is still ongoing. Even nowadays, many existing ITSs are not available online.

While serving as a source of ideas for future systems, ELM-ART also remained operation for many years and is still in use at the time of writing. Over many years of its service, more than 20,000 people from all over the world used ELM-ART. More than 1000 of them completed the whole course and gave interesting feedback. Many classes from different countries still use ELM-ART, some of them completing all lessons while others only use the first two or three lessons for working at the exercises. We have not performed large-scale controlled studies with the system; however, we continued to monitor it and learn from user data. In particular, we were interested in drop-outs, i.e., cases when users stopped learning with ELM-ART before finishing the complete course. Typical drop out happened at the end of a lesson. However, we discovered that some higher drop out rates at specific pages offer a hint to difficulties in the text. This information was used to revise the text at some pages, which, in turn, led to lower (normal) drop-out rates at these pages.

There is another important aspect of ELM-ART as a system still running and in use for more than 15 years, the longevity of an intelligent and adaptive learning system. No one will adopt a system to develop adaptive courses for when there only exist experimental adaptive systems developed in theses and short-living research projects. There are few systems existing for such a long time. Besides the direct followers Interbook and NetCoach an important system was AHA! (De Bra and Calvi 1998).

Discussion

Besides the positive results shown in several empirical studies and reported from user feedback, ELM-ART had several limitations. First, the system content was limited to the first few lessons in learning a programming language. This limitation to the relatively early stage of learning in a new domain can be found in most intelligent tutoring systems up to now. Many users asked for extending ELM-ART with more lessons on more complex programming concepts. However, the effort required to handle more complex programming problems was too high with respect to the advantages of the possible learning support. From a pedagogical point of view, supporting new learners in their first steps of learning in a new domain is more valuable than supporting advanced learners. Learning totally new concepts is the hardest and most frustrating stage that often causes new learners to give up learning at the early stage. An adaptive and intelligent learning system might carry novice learners through the most challenging first steps creating a solid foundation for mastering the next steps with a good textbook and well selected examples on one's own.

Another problem of ELM-ART that prevented its broader expansion is the limitation of the ELM diagnosis to functional programming in LISP. Generally, the ELM approach could be used in other domains, but developing the concepts and rules for another domain is very complex and time consuming. We do not see an easy way to develop an authoring tool that would be able to support developing episodic learner models for arbitrary domains. This motivated us to concentrate more on the domain

independent aspects of ELM-ART's adaptive textbook part and to develop the authoring tools like InterBook and NetCoach. With such a tool it is easy to create new adaptive learning courses with exercises and tests and with adaptive navigation support and individual curriculum sequencing.

Finally, the expandability of ELM-ART was limited because it was designed and built as a closed corpus system. In other words, only system authors could add new text content and new activities to the system. As a result, ELM-ART, while being a Web-based system, was not able to benefit from the availability of other Web content for learning Lisp. This expandability problem has been addressed to some extent in the authoring tools InterBook and NetCoach based on ELM-ART ideas. More extensively the closed corpus problem was addressed in subsequent research. For example, KBS-HyperBook (Henze and Nejdl 2001), SIGUE (Carmona et al. 2002) suggested an approach to integrate pre-existing open corpus textual content into the ELM-ART-like adaptive textbook. KnowledgeTree (Brusilovsky 2004) introduced a distributed architecture for adaptive E-learning that makes it possible to add new kinds of intelligent activities to an existing system. Another open corpus approach is AHA! (De Bra and Calvi 1998) and its successor GALE (Smits and De Bra 2011). These and other works opened the way to future generations of open adaptive and intelligent textbooks that could be gradually expanded integrating new content and new learning activities produced by a broad community of practice.

With several limitations of the ELM-ART approach discussed above, the system received relatively few expansions since its original version created in 1996. Instead, our work focused on refining and improving the systems making most of the innovative approach that it implemented. We believe that it contributed to the longevity of the system that is still running (as an extension to a NetCoach-Server) at the University of Education in Freiburg, Germany. Anyone can explore the system and learn programming in LISP for free with ELM-ART and we welcome the reader to try it. The current URL is http://art.ph-freiburg.de/Lisp-Course

References

Brown, M. H., & Najork, M. A. (1997). Collaborative active textbooks. *Journal of Visual Languages and Computing, 8*(4), 453–486.

Brusilovsky, P. (1992). Intelligent tutor, environment and manual for introductory programming. *Educational and Training Technology International, 29*(1), 26–34.

Brusilovsky, P. (1995). Adaptive learning with WWW: The Moscow State University Project. In: P. Held and W. F. Kugemann (eds.) Proceedings of Telematics for Education and Training Conference, Amsterdam, 24-26 November 1994, IOS, pp. 252–255.

Brusilovsky, P. (2004). KnowledgeTree: a distributed architecture for adaptive e-learning. In: Proceedings of 13th International World Wide Web Conference, WWW 2004, New York, NY, 17–22 May, 2004, ACM Press, pp. 104–113.

Brusilovsky, P. and Anderson, J. (1998). ACT-R electronic bookshelf: An adaptive system for learning cognitive psychology on the Web. In: H. Maurer and R. G. Olson (eds.) Proceedings of WebNet'98, World Conference of the WWW, Internet, and Intranet, Orlando, FL, November 7–12, 1998, AACE, pp. 92–97.

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.

Brusilovsky, P. & Pesin, L. (1994). ISIS-Tutor: An adaptive hypertext learning environment. In: H. Ueno and V. Stefanuk (eds.) Proceedings of JCKBSE'94, Japanese-CIS Symposium on knowledge-based software engineering, Pereslavl-Zalesski, Russia, May 10–13, 1994, EIC, pp. 83–87.

Brusilovsky, P., & Peylo, C. (2003). Adaptive and intelligent Web-based educational systems. *International Journal of Artificial Intelligence in Education, 13*(2–4), 159–172.

Brusilovsky, P., & Weber, G. (1996). Collaborative example selection in an intelligent example-based programming environment. In D. C. Edelson & E. A. Domeshek (Eds.), *Proceedings of international conference on learning sciences, ICLS'96* (pp. 357–362). Evanston: AACE.

Brusilovsky, P. & Zyryanov, M. (1993). Intelligent tutor, environment and manual for physical geography. In: T. Conlon (ed.) Proceedings of Seventh International PEG Conference, Edinburgh, Scotland, 2–4 July, 1993, pp. 63–73.

Brusilovsky, P., Schwarz, E., Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. In: C. Frasson, G. Gauthier and A. Lesgold (eds.) Proceedings of Third International Conference on Intelligent Tutoring Systems, ITS-96, Montreal, Canada, June 12–14, 1996, Springer Verlag, pp. 261–269.

Brusilovsky, P., Schwarz, E., & Weber, G. (1996). A tool for developing adaptive electronic textbooks on WWW. In: H. Maurer (ed.) Proceedings of WebNet'96, World Conference of the Web Society, San Francisco, CA, October 15–19, 1996, AACE, pp. 64–69.

Burow, R. & Weber, G. (1996). Example explanation in learning environments. In: C. Frasson, G. Gauthier and A. Lesgold (eds.) Proceedings of Third International Conference on Intelligent Tutoring Systems, ITS-96, Berlin, June 12–14, 1996, Springer Verlag, pp. 457–465.

Butz, C.J., Hua, S., & Maguire, R.B. (2004). A Web-Based Intelligent Tutoring System for Computer Programming. In: Proceedings of Web Intelligence, 2004. WI 2004. Proceedings. IEEE/WIC/ACM International Conference on, IEEE, pp. 159–165.

Carmona, C., Bueno, D., Guzmán, E., Conejo, R. (2002). SIGUE: Making Web Courses Adaptive. In: P. De Bra, P. Brusilovsky and R. Conejo (eds.) Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002), Málaga, Spain, May 29–31, 2002, Springer-Verlag, pp. 376–379.

Davidovic, A., Warren, J., & Trichina, E. (2003). Learning benefits of structural example-based adaptive tutoring systems. *IEEE Transactions on Education, 46*(2), 241–251.

De Bra, P., & Calvi, L. (1998). AHA! An open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia, 4*, 115–139.

De Bra, P., Smits, D., & Stash, N. (2006). The design of AHA! In: U. K. Wiil, P. J. Nürnberg and J. Rubart (eds.) Proceedings of Seventeenth ACM Conference on Hypertext and Hypermedia (Hypertext 2006), Odense, Denmark, August 25–26, 2006, ACM Press, pp. 133.

El-Khouly, M. (2000). Expert tutoring system for teaching computer programming languages. *Expert Systems with Applications, 18*(1), 27–32.

Henze, N., & Nejdl, W. (2001). Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education, 12*(4), 325–350.

Kavcic, A. (2004). Fuzzy user modeling for adaptation in educational hypermedia. *IEEE Transactions on Systems, Man, and Cybernetics, 34*(4), 439–449.

Klein, B., Dörr, G., & Weber, G. (1999). Multimediale lernumgebungen in der hochschullehre: Das hypermediale lernprogramm "incops" zur einführung in die kognitionspsychologie. *GMW-Forum, 2–3*, 6–14.

Koedinger, K., McLaughlin, E., & Heffernan, N. (2010). A quasi-experimental evaluation of an on-line formative assessment and tutoring system. *Journal of Educational Computing Research, 43*(4), 489–510.

Köhne, A. & Weber, G. (1987). STRUEDI: A LISP-structure editor for novice programmers. In: H. J. Bullinger and B. Schackel (eds.) Proceedings of Human-Computer Interaction INTERACT '87, North-Holland, pp. 125–129.

Korhonen, A., Naps, T., Boisvert, C., Crescenzi, P., Karavirta, V., Mannila, L., Miller, B., Morrison, B., Rodger, S., Ross, R., & Shaffer, C. (2013). Requirements and Design Strategies for Open Source Interactive Computer Science eBooks. In: Proceedings of Proceedings of the ITiCSE Working Group Reports Conference on Innovation and Technology in Computer Science Education-working Group Reports, Canterbury, England, United Kingdom, ACM, pp. 53–72.

Mallery, J. C. (1994). A Common LISP hypermedia server. In: Proceedings of the First International Conference on the World-Wide Web, May 25, 1994.

Mitrovic, A. (2003). An intelligent SQL tutor on the Web. *International Journal of Artificial Intelligence in Education, 13*(2–4), 173–197.

Murray, T. (2003). MetaLinks: authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. *International Journal of Artificial Intelligence in Education, 13*(2–4), 199–233.

O'Keeffe, I., Brady, A., Conlan, O., & Wade, V. (2006). Just-in-time generation of pedagogically sound, context sensitive personalized learning experiences. *International Journal on E-Learning, 5*(1), 113–127.

Papanikolaou, K. A., Grigoriadou, M., Kornilakis, H., & Magoulas, G. D. (2003). Personalising the interaction in a Web-based educational hypermedia system: the case of INSPIRE. *User Modeling and User Adapted Interaction, 13*(3), 213–267.

Schöch, V., Specht, M., & Weber, G. (1998). "ADI" - an empirical evaluation of a tutorial agent. In: T. Ottmann and I. Tomek (eds.) Proceedings of ED-MEDIA/ED-TELECOM'98 - 10th World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications, Freiburg, Germany, June, 20–25, 1998, AACE, pp. 1242–1247.

Schwarz, E., Brusilovsky, P., & Weber, G. (1996). World-wide intelligent textbooks. In: Proceedings of ED-TELECOM'96 - World Conference on Educational Telecommunications, Boston, MA, June 17–22, 1996, AACE, pp. 302–307.

Smits, D. & De Bra, P. (2011). GALE: A Highly Extensible Adaptive Hypermedia Engine. In: Proceedings of Proceedings of the 22Nd ACM Conference on Hypertext and Hypermedia, Eindhoven, The Netherlands, ACM, pp. 63–72.

Specht, M., & Oppermann, R. (1998). ACE - Adaptive Courseware Environment. *The New Review of Hypermedia and Multimedia, 4*, 141–161.

Specht, M., Weber, G., Heitmeyer, S., & Schöch, V. (1997). AST: Adaptive WWW-Courseware for Statistics. In: P. Brusilovsky, J. Fink and J. Kay (eds.) Proceedings of Workshop "Adaptive Systems and User Modeling on the World Wide Web" at 6th International Conference on User Modeling, UM97, Chia Laguna, Sardinia, Italy, June 2, 1997, pp. 91–95, also available at http://www.contrib.andrew.cmu.edu/~plb/UM97_workshop/Specht.html.

Unanue, M. ı., Velasco, P., Flores, P., Fuentes, U., & Velázquez, I. (2002). Electronic books for programming education: a review and future prospects. In: Proceedings of ITiCSE '02: Proceedings of the 7th annual conference on Innovation and technology in computer science education, Aarhus, Denmark, ACM, pp. 34–38.

Weber, G. (1996a). Episodic learner modeling. *Cognitive Science, 20*(2), 195–236.

Weber, G. (1996b). Individual selection of examples in an intelligent learning environment. *Journal of Artificial Intelligence in Education, 7*(1), 3–31.

Weber, G. (1999). Adaptive learning systems in the World Wide Web. In: J. Kay (ed.) Proceedings of 7th International Conference on User Modeling, UM99, Wien, June 20–24, 1999, SpringerWienNewYork, pp. 371–378.

Weber, G., & Brusilovsky, P. (2001). ELM-ART: an adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education, 12*(4), 351–384.

Weber, G. & Möllenberg, A. (1994). ELM-PE: A knowledge-based programming environment for learning LISP. In: T. Ottmann and I. Tomek (eds.) Proceedings of ED-MEDIA'94 - 6th World Conference on Educational Multimedia and Hypermedia, Vancouver, Canada, June, 25–29, 1994, AACE, pp. 557–562.

Weber, G., Kuhl, H.-C., & Weibelzahl, S. (2002). Developing adaptive internet based courses with the authoring system NetCoach. In S. Reich, M. M. Tzagarakis, & P. M. E. De Bra (Eds.), *Hypermedia: openness, structural awareness, and aptivity* (pp. 226–238). Berlin: Springer.

Yudelson, M., Brusilovsky, P., & Sosnovsky, S. (2004). Accessing interactive examples with adaptive navigation support. In: Proceedings of IEEE International Conference on Advanced Learning Technologies, ICALT 2004, Joensuu, Finland, August 30 - September 1, 2004, AACE, pp. 842–843.