Lecture Notes in Computer Science 753

Leonard J. Bass Juri Gornostaev
Claus Unger (Eds.)

Human-Computer Interaction

Third International Conference, EWHCI '93
Moscow, Russia, August 1993
Selected Papers

Springer-Verlag

Towards an Adaptive Hypermedia Component for an Intelligent Learning Environment

Peter Brusilovsky, Leonid Pesin and Mikhail Zyryanov

International Centre for Scientific and Technical Information,
Kuusinen str. 21b, Moscow 125252, Russia
E-mail: plb@plb.icsti.su
and
Department of Applied Mathematics and Cybernetics,
Moscow State University, Moscow, Russia

Abstract. This paper discusses the problem of integration of hypermedia and Intelligent Learning Environments (ILEs) technologies and the problem of creating an adaptive hypermedia component for ILEs. Our experience of creating an adaptive on-line help facilities for ITEM/IP system is described. This experience forms a background for our hypermedia work and provides some good ideas for it. We also present our approach to integration of a hypermedia component into internal structure of ILE and illustrate it with two examples of adaptive hypermedia components for the most recent versions of our ILEs ISIS-Tutor and ITEM/PG. Finally, we summarize main features of this approach, provide some references to related works, and consider some issues of adaptive hypermedia in general.

1 Introduction

An intelligent learning environment is a relatively new kind of intelligent educational system which combines the features of traditional Intelligent Tutoring Systems (ITS) and learning environments. Some years ago many developers of educational systems considered ITS and learning environments as different and even contradictory ways of using computers in education. The recent success of well-known ILEs Smithtown [20], Sherlock [16], and Quest [23] showed that these approaches are not contradictory but rather complementary. ITSs are able to control learning adaptively on various levels, but generally do not provide tools to support free exploration. In comparison, learning environments and microworlds support exploratory learning, but they lack the control of an intelligent tutor. Without such control the student often works inefficiently and never discovers some important features of the subject.

The same situation happens now with ITSs and educational hypermedia systems. They are most often considered as two different approaches to using computers in education, while these approaches are really complementary. Recent research has demonstrated that hypermedia can provide the basis for an exploratory learning system but that by itself such a system is insufficient, needing to be supplemented by more directed guidance [13]. This guidance is often an important ingredient of effective learning from hypermedia, and it can be provided by an intelligent tutoring component. In comparison, hypermedia approach can add a new dimension to traditional ITS/ILE by providing a tool for student driven acquisition of domain knowledge.

We think that in many domains it is possible to achieve a good results by developing an educational computer system which integrates the capabilities of an intelligent tutor, a learning environment, and a hypermedia system. The problem is, however, to find an appropriate paradigm or basis for such integration. Our position is that an integrated system should be not just a sum but a real integration of its components. First, one component should be able to use the capabilities of another component as well as exchange or share data with it. Second, the results of students' work with any of the components during the session should be taken into account by other components to adapt their performance to the knowledge level and personal features of the particular student.

For a number of years we have been investigating the problem of creating an intelligent learning environment in the domain of teaching programming. We have designed and tested the Intelligent Tutor, Environment and Manual for Introductory Programming (ITEM/IP). ITEM/IP [4] integrates an ITS for programming, a programming environment, and adaptive on-line help facilities on the basis of the structural domain model and the overlay student model.

The results of students' work are reflected in the student model and used by other system components to adapt themselves to the knowledge level of the particular student. We have specially investigated the use of student models for adaptive curriculum sequencing by the tutoring component of ILE [3], and for developing an adaptive interface for the environment components of ILE [5].

More recently we have tried to apply our approach to some other domains. We have developed ILEs for mathematics [8], information retrieval [18], and geography [7]. First versions of these ILEs contained an intelligent tutor and a learning environment integrated together on the basis of the domain and student models, but did not contain hypermedia components. It was not an extension of our initial approach, but rather a validation of its applicability to another domains.

Now we are trying to extend our approach by integrating hypermedia technology into our existing ILEs. The problem here is not just to design a hypermedia component for one of the existing ILEs, but to find ways to integrate this component into a particular ILE. It means that the hypermedia component has to use the student model both to adapt its performance to the given student, as well as to update the student model to reflect the results of the student's work with the component.

This paper discusses the problem of integration of hypermedia and ILE technologies and the problem of creating an adaptive hypermedia component for ILEs. First, we present our experience of creating an adaptive on-line help facility for ITEM/IP system. This experience forms a background for our hypermedia work and provides some good ideas for it. Then we present our approach to the integration of a hypermedia component into the structure of ILE and illustrate it with two examples of adaptive hypermedia components for the most recent versions of our ILEs: ISIS-Tutor and ITEM/PG. Finally, we summarize the main features of the suggested approach, provide some references to related works, and discuss some problems of adaptive hypermedia in general.

2 Adaptive access to teaching material in ITEM/IP: Lessons learned

ITEM/IP stands for Intelligent Tutor, Environment and Manual for Introductory Programming. ITEM/IP is an intelligent learning environment for teaching and learning introductory programming by the mini-language approach [2]. In the mini-language approach beginners learn what programming is, while learning how to use a simple mini-language to control a robot which acts in a microworld environment. ITEM/IP [4] was designed in 1985 to support an introductory programming course based upon the educational mini-language Turingal for first-year students at Moscow State University.

The student can use ITEM/IP in the following modes:

1. *Novice Programming Environment.* To work with Turingal programs the student can make use of a complete set of tools for program design and debugging. One of the functions of the environment is to display visualizations of the student's programs. With the help of the environment, the student can observe various programs "at work", experiment with them, and gradually learn from experience, observations and mistakes.

2. *Instruction.* At any moment of work the student can ask the teaching component to generate the next teaching operation: a new concept or construct, an example construct, a test, an example for problem solving, or a programming problem to solve. The teaching component analyses the domain knowledge and the student model and offers the student both the optimal teaching operation and a list of all relevant (i.e. ready to be applied) teaching operations. The student who is not satisfied with the optimal operation suggested by the system can choose any relevant teaching operation using adaptive hierarchical menus.

3. *Repetition and On-Line Manual.* At any moment of work the student has menu-based access to all previous teaching material: presentation of any previously learned concept, demonstration of all learned examples, and analysis of any explained or solved problem. This mode offers access to the learned material of the course as reference, thus supporting on-line help and example-based programming.

ITEM/IP provides both instruction and reference access to the teaching material. Generally, these two modes of access require different presentation and structuring of the material, so many good books on programming languages consist of two independent parts: the tutorial and the reference manual. In the ITEM/IP project, we have tried to make an integrated tutorial and manual by providing adaptive presentation of the same teaching material on the base of domain and student models.

The domain model in ITEM/IP is a net with nodes corresponding to programming concepts and mini-language constructs (i.e. domain knowledge elements) and with links reflecting several kinds of relations between nodes. The overlay student model reflects by a set of integer counters the extent to which the student has mastered the concept or the construct. The student model is always kept up to date and supports adaptive work of all modules.

All the teaching material is stored in the base of teaching operations in a form of frames. There are three kinds of frames: concept/construct frames, example frames and problem frames. The presentation module uses these frames in both instruction and repetition modes to generate the five kinds of teaching operations mentioned above. Since the presentation of teaching material is adapted to the current level of student knowledge, a repeated explanation is usually more concise and complete than the original one.

For example, presentation of a concept or a construct consist of some textual information about it followed by a presentation of the relations which link the presented element with other domain knowledge elements. The textual information which is stored for the given concept or construct can be divided into a sequence of text fragments. Each fragment has a condition which addresses the knowledge level of the given and related constructs. While producing a description of the concept, the presentation module presents only the fragments whose condition is true. The more the concept or construct is learned, the more concise its presented description. The information about the relations of the given construct is not stored as text, but is generated using simple templates. To avoid confusion, only the relations with already-known nodes are presented.

The method of adaptive presentation used in ITEM/IP provides a smooth transition from learning to on-line help access and supports adaptive student-driven access to teaching material. This method was tested in a real classroom. We were quite satisfied with the adaptive presentation itself, but we were not satisfied with using hierarchical menus as the only access to teaching material in instruction and repetition modes. The idea of student-driven access to teaching material was to support exploratory learning and student-driven acquisition of conceptual knowledge. However, the mechanism used for it in ITEM/IP appears to be too complicated for novice programmers.

For example, when learning or repeating a description of a construct, the student was provided with the references to all previously learned related concepts and constructs. However, to refresh knowledge about it, the student needed to re-enter the repetition mode from the top and access the related information by a three-level menu. An even bigger problem was that the menus for the instruction and repetition modes provided two different views of the domain knowledge, making understanding of the domain knowledge structure much more difficult for the students.

To overcome these problems, we decided to redesign the high level architecture of the system and to apply hypermedia technology to support unified access to all of the teaching material. Our goal was to develop an adaptive educational hypermedia, using our general student model based approach to adaptation in ILEs, in a way that was similar to the way we used it to create the adaptive presentation. The rest of this paper reports on our approach and recent experience in creating adaptive hypermedia components for intelligent learning environments. We briefly describe two systems - ISIS-Tutor and ITEM/PG - which employ adaptive hypermedia, and discuss our work in the context of related works.

3 Towards an adaptive educational hypermedia

The first important step in integrating the hypermedia network into an intelligent learning environment is to take the domain model network of an ILE as a basis for the hypermedia network. More exactly, we suggest that the hypermedia network be designed precisely as a visualized (and externalized) domain network. Each node of the domain network should be represented by a node of the hyperspace, while the links between domain network nodes constitute main paths between hyperspace nodes. Thus the structure of the overall hyperspace resembles the pedagogic structure of the domain knowledge. Each domain network node will have a hypermedia 'page' as an external representation, and a frame as an internal representation. We suggest also that these hypermedia pages be constructed from the corresponding frames by a special program rather than being stored directly in presentation format. This saves page design time and provides space for adaptation.

We used the above approach in our recent work on adaptive hypermedia components for ILEs. The following two subsections report on how the domain and student models were used in ISIS Tutor and ITEM/PG to generate the 'adaptive pages' for hypermedia nodes.

3.1 The adaptive hypermedia manual in ISIS Tutor

ISIS-Tutor [18] is an ILE to support learning of the print formatting language of a well-known information retrieval system: CDS/ISIS. This system is supplied by UNESCO and used widely in ICSTI and other information centres in the world. The print formatting language is the key to many CDS/ISIS operations, and mastering this language is important for effective use of the system.

The architecture of ISIS-Tutor resembles the architecture of ITEM/IP in most details. It contains an environment for experimenting with the language and a tutoring component which deals with three kinds of teaching operations: concept presentations, examples and problems. However, the domain model - which is a network of 69 concepts and constructs - is twice as complex as the one in ITEM/IP. The overall 'space' of teaching material is larger as well.

ISIS-Tutor applies the same ideas of student model based adaptation as ITEM/IP, but adds some important improvements. When presenting a concept or construct, ITEM/IP and ISIS-Tutor provide an adaptive list of related concepts and examples. To reduce the student's cognitive load, ITEM/IP puts only learned related concepts in the list, and only those examples which do not use unknown constructs. ISIS-Tutor lists both learned and ready-to-be-learned concepts (a concept is ready to be learned if all of the preceding concepts are known to the student). To distinguish old and new concepts, two different colours are used. Another new feature of ISIS-Tutor is hypermedia-like access to related concepts and examples. The student can select any related concept or example from the generated list to move to a related page of teaching material. The selected concept or example is presented adaptively to the student, who can then read information about the concept or experiment with an example and then use its links to navigate to other related concepts or examples.

Thus ISIS-Tutor provides a hypermedia-like way to investigate the teaching material. Navigating along the links to related concepts and examples, students can repeat learned pieces of knowledge as well as learn new material. To support navigation and learning, all links to new material are visibly marked. ISIS-Tutor unifies the instruction and repetition modes of ITEM/IP and provides a unified adaptive view of the structured teaching material. Note that ISIS-Tutor and ITEM/IP both use two similar menu-based entries into the teaching material (for new and repeating material), but in ISIS-Tutor, these menus are a secondary way of access, a kind of index.

3.2 The adaptive hypermedia component of ITEM/PG

ITEM/PG [7] is an ILE for 13-14 year old students taking a course in the physical geography of oceans and continents. The pedagogical goal of the work with ITEM/PG is to learn the relationships between the different natural components of an island, located somewhere in the northern part of the Atlantic Ocean. The island is characterized by several components, such as position, origin, climate etc. Each component has several possible values. For example, the options for the origin of the island are continental, volcanic, roof or atoll. The values of the components are related to each other. For example, the climate depends on the position of the island. A more detailed description of the domain and a pedagogical basis for the overall project can be found in [6].

The main components of ITEM/PG are the domain model, the student model, the tutor, the learning environment and the hypermedia network. The student can work in exploratory mode, browsing the hyperspace or experimenting with the environment. At any given moment the student can apply to the tutor for the next "optimal" teaching operation, and can then follow the tutor's suggestion or give up.

According to our general approach, the hypermedia network is just the visualized domain network. There are three main kinds of hyper-nodes: component, value and rule. Each kind of hyper-node has a special screen representation which is not stored in screen format, but generated from the corresponding frame by a special program. For example, the program for the "value" kind of node generates a window which contains the name of the value, the description, the

icon and the hyper-links to the component node and related rule nodes.

The student model is used by the hypermedia component to adapt the screen representation of hyper-nodes to the current knowledge level of the given student. Applying the student model, the hypermedia component distinguishes four states for hyper-nodes: not ready to be learned, ready to be learned, known and well known. Thus, at any moment, the hyperspace is divided implicitly into several "zones". In particular, the ready to be learned nodes form a "zone of proximal development" [22] or a "knowledge front" [12].

Our idea is that different zones have different meanings for the student, and marking these zones visually should help the student in hyperspace navigation just as bookmarks help the student with textbooks. To mark the zones, the hypermedia component marks the hyper-links of each node in four different ways. For example, the links to the nodes which are not ready to be learned are dimmed, so as not to distract the student. The links to ready to be learned nodes are colored green, inviting the student to visit them.

Another important feature of ITEM/PG is the bilateral linking of the student model and the hypermedia component. The links are used in one direction when the hypermedia component accesses the student model to adapt itself. The links are used in the other direction when student visits to a particular node of the hypermedia component cause a change in the value of how well learned that node is in the overlay student model. Thus ITEM/PG provides a good example of how the hypermedia component can be integrated into the structure of an ILE.

4 Discussion

We have presented three systems which demonstrate some steps towards an integration of adaptive hypermedia and ILEs. Using our experience, we can now formulate the main features of our approach to developing an adaptive hypermedia component for ILE.

- The hypermedia network is designed as the visualized and externalized domain network.
- An external representation of a knowledge element (a page), is generated or assembled from its internal representation (a frame).
- The content of the hypermedia page is adapted to the student knowledge reflected in the student model.
- The links from any node to related nodes are marked visually, reflecting the current "educational state" of the related nodes for the given student.
- Student interaction with the hypermedia component is reflected in the student model and can be used by other components of ILE.

All these techniques are important for creating a truly integrated hypermedia for ILE. Each of the listed items provides a good area for research. Here we provide some references to existing works in these areas. The idea about creating the hypermedia network from domain network is quite popular and has been discussed in [14, 15]. Generation of adaptive concept explanation is another popular field of research [17, 21]. An application of these ideas to hypermedia was reported recently in [11]. Adaptive "history-based" navigation tools were discussed in [10].

Some problems, however, are not well studied and need additional investigation. For example, we use different methods to deal with links from the given node to not ready to be learned nodes: these links are hidden in ITEM/IP and ISIS-Tutor and dimmed in ITEM/PG. We can't say now which way is better. By hiding these links completely we can reasonably limit the cognitive load of the student, which is quite important for novices. However, hiding any links looks unnatural for hypermedia. It can form an incorrect mental model of the domain knowledge. Such adaptation is also more intrusive than just dimming these links. To investigate this problem, we plan to compare hiding vs. dimming in an experiment.

Another important problem is the relevance of adaptation. A system can use very elaborate strategies to provide the student with the "best" teaching operation or "optimal" help detail. The problem is whether the student agrees with the choice. The student could prefer another next operation or more (less) detailed help. To deal with this problem, we feel that the adaptation should not be intrusive, and that the student should be provided with control over adaptation. In our future work we plan to provide the student with the possibility to customize the level of adaptation. We plan to make the student model visible to the student and let the student change some part of it. We have not heard of any existing ITS or ILE with a student-accessible model. However, some ITS's [1] use a student-entered part of the student model for the purpose of adaptation. We consider techniques for creating visible and accessible student models to be the general line for the future development of ITS's [9, 19].

References

1. A. Barr, M. Beard, R.C. Atkinson: The computer as tutorial laboratory: the Stanford BIP project. International Journal on the Man-Machine Studies 8, 567-596 (1975)
2. P.L. Brusilovsky: Turingal - the language for teaching the principles of programming. In: EUROLOGO 91. Proc. Third European Logo Conference, Parma. Italy, 27-30 August 1991, A.S.I., Parma, pp. 423-432
3. P.L. Brusilovsky: A framework for intelligent knowledge sequencing and task sequencing. In: C. Frasson, G. Gauthier and G.I. McCalla (eds.): Intelligent Tutoring Systems. Proceedings of the Second International Conference, ITS'92. Berlin: Springer-Verlag 1992, pp. 499-506
4. P.L. Brusilovsky: Intelligent Tutor, Environment and Manual for Introductory Programming. Educational and Training Technology International 29, 26-34 (1992)
5. P.L. Brusilovsky: Student as user: Towards an adaptive interface for an intelligent learning environment. Paper accepted to AIED'93 World Conference.
6. P.L. Brusilovsky, T.B. Gorskaya-Belova: The Environment for Physical Geography Teaching. Computers and Education 18, 85-88 (1992)
7. P. Brusilovsky, M. Zyryanov: Intelligent learning environment for geography: towards an integration of emerging educational technologies. In: N.Hammond and A.Trapp (eds.): CAL into the Mainstream. CAL 93 Conference Handbook 1993, pp.47-48
8. V.G. Brusilovsky, P.L. Brusilovsky: The student model in a teaching system for mathematical differentiation. Design and application of PC software in the education process. In: Proc. of the 5-th All-Union Workshop. Moscow: 1989, pp. 23-24 (In Russian)
9. A.T. Corbett, J.R. Anderson: Student modeling and mastery learning in a computer-based programming tutor. In: C. Frasson, G. Gauthier, G.I. McCalla (eds.): Intelligent Tutoring Systems. Proceedings of the Second International Conference, ITS'92. Berlin: Springer-Verlag 1992
10. B. de La Passardiere, A. Dufresne: Adaptive navigational tools for educational hypermedia. In: I. Tomek (ed.): Computer Assisted Learning. Proceedings of the 4th International Conference, ICCAL'92. Berlin: Springer-Verlag 1992, pp. 555-567
11. F. De Rosis, N. De Carolis, S. Pizzutilo: User tailored hypermedia explanations. In: INTERCHI'93 Adjunct Proceedings, Amsterdam, 24-29 April, 1993, pp. 169-170
12. I.P. Goldstein: The Genetic graph: a representation for the evolution of procedural knowledge. International Journal on the Man-Machine Studies 11, 51-77 (1979)
13. N. Hammond: Hypermedia and learning: Who guides whom?. In: H. Maurer (ed.): Computer Assisted Learning. Proceedings of the 2-nd International Conference, ICCAL'89. Berlin: Springer-Verlag 1989, pp. 167-181
14. R.J. Hendley, C.D. Whittington, N. Juraschek: Hypermedia generation from domain representations. Computers and Education 20, 127-132 (1993)
15. M.R. Kibby: Intelligent hypermedia for learning. In: R.M. Bottino, P. Forcheri, M.T. Molfino (eds.): Knowledge Based Environments for Teaching and Learning. Proceedings of the Sixth International PEG Conference. Genova: 1991, pp. 3-12
16. S.P. Lajoie, A. Lesgold: Apprenticeship training in the workplace: computer-coached practice environment as a new form of apprenticeship. Machine Mediated Learning 3, 7-28 (1990)
17. C.L. Paris: Tailoring object description to a user's level of expertise. Computational Linguistics 14, 64-78 (1988)

18. L.A. Pesin, P.L. Brusilovsky: Adaptive task sequencing in ISIS-Tutor. In: Abstracts of East-West Conference on Emerging Computer Technologies in Education, Moscow, 6-9 April 1992, p. 176
19. J. Self: Bypassing the intractable problem of student modelling. In: C. Frasson (Ed.): ITS'88. Proc. of the Intelligent Tutoring Systems conference. Univ. of Montreal. Montreal 1988
20. V.J. Shute, R. Glaser: A large-scale evaluation of an intelligent discovery world: Smithtown. Interactive Learning Environments 1, 51-77 (1990)
21. D.H. Sleeman: UMFE: a user modeling front end system. International Journal on the Man-Machine Studies 23, 71-88 (1985)
22. L.S. Vygotsky: Mind in society. Cambrige: Harvard University Press 1978
23. B.Y. White, J.R. Frederiksen: Causal model progressions as a foundation for intelligent learning environments. Artificial Intelligence 42, 99-157 (1990)