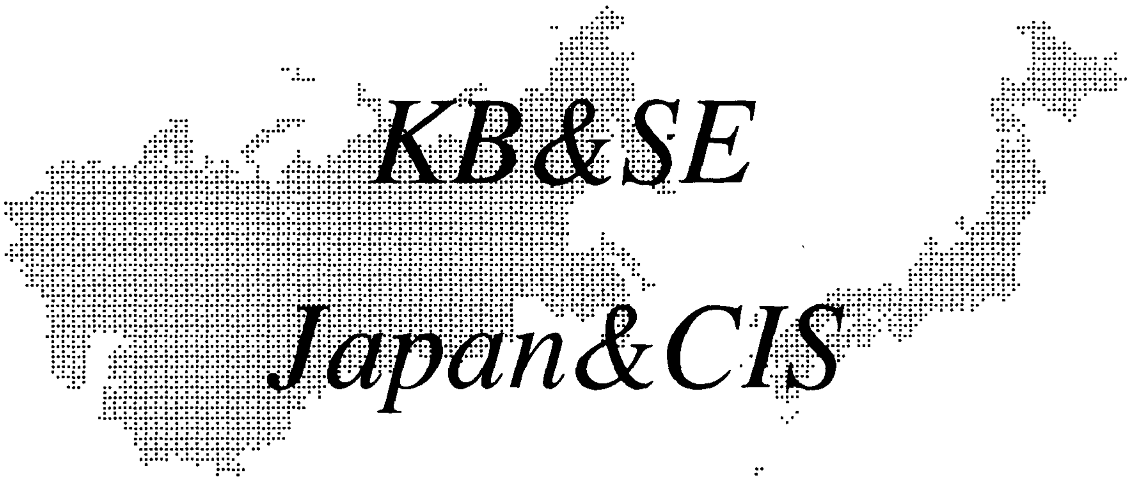
Proceedings of

JCKBSE'94

Japan-CIS Symposium on Knowledge Based Software Engineering' 94



Site
Pereslavl-Zalesskii, Russia

Date
May 10-13, 1994

Sponsored by
SIG on Knowledge Based Software Engineering,
Institute of Electronics, Information and Communication Engineers(EIC), Japan
Association for Artificial Intelligence of Russia

In cooperation with
SIG on Knowledge-Based Systems, Japanese Society of Artificial Intelligence
Research Center for Artificial Intelligence of Program Systems,
Institute of Russian Academy of Science
SIG on Software Engineering, Information Processing Society of Japan
CAI Society of Japan
IEEE Tokyo Section Computer Chapter
Research Center for the Problems of Quality of Education, Russia

Supported by
International Communications Foundation (I CF), Japan

ISIS-TUTOR:
AN ADAPTIVE HYPERTEXT LEARNING ENVIRONMENT

Peter Brusilovsky
International Centre for
Scientific and Technical Information
Kuusinen str. 21b, Moscow 125252, Russia
e-mail: plb@plb.icsti.su

Leonid Pesin
Department of Applied Mathematics
and Cybernetics
Moscow State University, Moscow, Russia

Keywords: intelligent tutoring system, adaptive task sequencing, hypertext, intelligent learning environment, print format language, concept map, student model

Abstract. The paper discusses a student-model-centred approach to building hypermedia-based intelligent learning environments and then presents ISIS-Tutor system, which provides an example of integrating a hypermedia technology into an intelligent learning environment. ISIS-Tutor is intended for learning the print formatting language of an information retrieval system.

I Introduction

Some years ago many developers of educational systems consider Intelligent Tutoring Systems (ITS) and learning environments as different and even contradictory ways of using computers in education. Recent success of well-known Intelligent Learning Environments (ILE) Smithtown [10] and Sherlock [8] showed that these ways are not contradictory but rather complimentary. ITSs are able to control learning adaptively on various levels, but generally do not provide tools to support free exploration. 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 discover some important features of the subject.

The wine 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 complimentary. 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 [5]. The guidance can be provided by an intelligent tutoring component. In comparison, hypermedia approach can add 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 result by developing an educational computer system which integrates the capabilities of an intelligent tutor, a learning environment, and a hypermedia system. To investigate the ways of building an ILE for programming we have designed and tested the Intelligent Tutor, Environment and Manual for Introductory programming (ITEM/IP). ITEM/IP [2] integrates an ITS for programming, a programming environment, and adaptive on-line help facilities on the basis of the domain model and the overlay student model which serve as a kernel of the system. The domain model is a kind of semantic network. Nodes and links of the network represent elements of domain knowledge and relations between the elements. The student model represents the level o£ the student's competence for each of the knowledge elements.

More recently we have tried to apply our student-model-centred approach to some other domains. In particular, we have developed ILEs for print formatting language [9] and geography [4]. 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. Now we are trying to extend our approach by integrating the hypermedia technology into our existing ILEs. The problems here is not just to design a hypermedia component for one of the existing ILEs, but to find the ways to integrate this component into a particular ILE. It means that the hypermedia component has both to use the student model to adapt its performance to the given student, as well its to update the student model reflecting the results of the student's work with the component.

This paper describes briefly our general approach to building a hypermedia-based intelligent learning environment [3] and then presents ISIS-Tutor system, which is an intelligent hypertext learning environment intended for learning the print formatting language of well known information retrieval system CDS/ISIS/M. From our point of view, ISIS-Tutor provides an interesting example of integrating a hypermedia technology into an intelligent learning environment. We conclude with a summary of the approach and lessons learned from our ISIS-Tutor project.

2 Integrating a hypermedia component into ILE: an approach

Our main idea is to take the domain model network as a basis for the hypermedia network. Such ideas have been discussed previously [6,7] and we consider these ideas as very important. More exactly, we suggest to design the hypermedia network just as the 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 constitutes main patlis 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 have a frame as an internal representation. We suggest also to generate these hypermedia pages from corresponded frames by a special program rather then store it directly in presentation forinat. It saves page design time and provides space for adaptation. This approach enables its to connect easily the hypermedia, network and the student model. First, any student's 'visits' to a, particular node of the network can be reflected by the corresponding component of the overlay student model (which is based on the same domain network). Second, the current state of the student model for the given node of the network and its links can be used by the hypermedia component to adapt the screen layout of the corresponding hyper-node, as well as the number and status of visible links to other nodes. More considerations about integration of hypermedia into an intelligent learning environment can be found in [3].

3 ISIS-Tutor: The domain and the architecture

ISIS-Tutor is an intelligent learning environment to support learning the print formatting language of well known information retrieval system CDS/ISIS/M for IBM PC (briefly - ISIS). This system is supplied by UNESCO and used widely in ICSTI and other information centres in the world. Print formatting language is a key to many CDS/ISIS operations and mastering the language is important for effective use of the system. In some sense, it is a special prograimnning language. To display or print a result of search or all database content an ISIS user have to write some sequence of print formatting commands, really more or less complex program on print formatting language. This format program is used by ISIS to do display of print the selected records of database in a desired format. ISIS consecutively applies the print formatting program to every record being printed thus printing an external presentation of the record. Print formatting commands, for example, can type a field of the current record or a part of field, can manipulate current output position or type a constant character string and so on. Print formats are also used in indexing and sorting. There are over 50 different commands in the print forinatting language, so a tutoring system for the language is really helpful.

ISIS-Tutor system is designed in ICSTI and Moscow State University. The first version of ISIS-Tutor was described in a short paper [9]. In this paper we present recent version of ISIS-Tutor which, from our point of view, provides an example of integrating a hypermedia tecluiology into an intelligent learning environment. ISIS-Tutor is written in embedded ISIS-Pascal and uses the power of ISIS for knowledge bases storage and access. ISIS-Tutor resembles the original architecture of ITEM/I? [2] in most details. It also contains an environment to experiment with the language and a tutor component which deals with three kinds of teaching operations: concept presentations, examples and problems. It uses similar domain and student models. However the domain model which is a network of 69 concepts and constructs is twice as more complex than one in ITEM/IP. The overall 'space' of teaching material is bigger as well. Finally, ISIS-Tutor contains the hypermedia component. The integration of guided tutoring mode (provided by the tutor) with free exploration mode (provided by the hypermedia and the environment) is the most interesting feature of ISIS-Tutor.

The tutor component supports adaptive task sequencing [1], which means that knowledge demonstrated by the student formerly are analyzed and system selects an optimal teaching operation to present. The student knowledge is represented in the student model reflecting the level the student has mastered each domain concept (skill). Each pedagogic task is connected with a set of concepts (we call it spectrum) which are required for its solving. Using the Student Model and task spectra the tutor can select an optimal task for the given student in each stage of learning.

The hypertext component supports student-driven acquisition of conceptual knowledge. It is important that hypertext component of the system is not a separate one, but an integrated part of the system. It means that the component use the student model to adapt its performance to the given student, as well as that it updates the student model reflecting the results of the student's work with the component.

The learning environment allows user to play and experiment with print formatting commands. It provides step-by-step execution and extended visualization. Student work in this component is also reflected by in the student model.

The student model is the heart of the system. It provides the system with the required adaptivity. Modules of ISIS-Tutor use the student model to adapt their work and update it reflecting the student progress. With the student model all modules can take into account the results of the student work with other modules. This feature turns ISIS-Tutor into a really integrated system. The following section describes ISIS-Tutor components in more details.

4 ISIS-Tutor components

The central part of ISIS-Tutor architecture are interrelated domain model and student model. In ISIS-Tutor the material being taught (knowledge about language) is represented as the domain model. The material is divided into a set of elementary concepts and structured as a directed graph (concept map) representing prerequisite relationships between concepts. The overlay student model (SM) reflects the extent the student has mastered language concepts by providing an integer counter for each of the concept. All learning material is also related with the domain model network. There are three kinds of teaching operations in ISIS-Tutor: concept presentation, problem to solve and example to analyze. The teaching operations (or learning tasks) are stored in the database of learning material. Each learning task is really a kind of frame. One of its slots contains the list of domain concepts related with the task, for example, the list of concepts used in the example. This list called spectrum provides the link with the domain model.

Each step of learning process in ISIS-Tutor is an application of one of two main teaching operations: selection and solving of a problem or selection and presentation of a new concept. The choice can be done by the user or by the system. In the first case user selects next teaching operation himself from suggested set of relevant concepts to study or problem to solve. The mode can be useful for an experienced student and for teacher, who wants to set a learning task manually for the student. In the second case system by using the Student Model selects the optimal teaching operation. This mode gives the adaptive sequence of tasks and concepts. Finally Jie student always can request a repeated presentation of a concept or a problem. This possibility is useful for a weak student, who has not mastered the material very well.

In any case selected teaching operation is a concept presentation or a problem solution. The example presentation is now not an independent teaching operation. It follows a concept presentation and can be requested from concept presentation window, where concept description, list of concepts connected with the current one and list of examples are offered to user. If the student selects one of these examples the system enters the environment mode with the selected example loaded.

The learning process is managed by the intelligent tutor. Using SM the tutor can select an optimal teaching operation - new concept presentation or problem, depending of the current state of user knowledge. The tutor usually tries to offer a problem for mastering concept which are known but not mastered enough by solving problems. If there are no relevant problems, then a new concept for study is selected. Considering SM on each step tutor can select optimal problem to consolidate just, studied concept or optimal concept to continue language study. The tutor can also compile the lists of all new relevant (ready to be used) teaching operations and all previously considered operations to repeat. After a teaching operation is completed the tutor updates SM - increase level (weight) of the learned concept or all concepts from the spectrum of successfully solved test or problem. Initially SM is empty, then while user works with the system the model is updated in parallel with growth of user's knowledge.

This scheme was used in the first version of ISIS-Tutor. It works well but lacks convenience of student-driven exploration of conceptual knowledge. For example, when learning or repeating a description of a concept, the student was provided with the references to all previously learned related concepts. However, to refresh the knowledge about it the student needs to re-enter the repetition mode from the top and access the related information by a three-level menu. New version of ISIS-Tutor adds some important improvements. The main visible difference with the old version is that all the teaching material in ISIS-Tutor forms real hypermedia network thus providing the student with more suitable way to browse it in student-controlled mode. When presenting a concept or construct, ISIS-Tutor provide an adaptive list of related concepts and examples. Now ISIS-Tutor supports the 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 to the student, who can read an information about the concept or experiment with an example and then use its links to navigate to other related concepts of examples.

Second improvement: the first version puts only known related concepts to the list while the new one lists both known and ready-to-be-learned concepts (a concept is ready to be learned if all the preceding concepts are known to the student). To distinguish known and new concepts they are marked in two different colours. It enables the student to learn new knowledge while browsing the hyperspace and support student navigation. Third, ISIS-Tutor adds hypermedia-like access from a concept to related examples by listing in another adaptive menu all the examples which are related with the current concept and do not use unknown concepts. When selecting an example (usually a program fragment using the original concept) the student moves to the programming environment component with the selected example loaded for interpretation. The student can play with the example using the visualizing interpreter, change the data and the example itself.

Thus ISIS-Tutor provides hypermedia-like way of investigating the teaching material. Navigating along the links to related concepts and examples, the 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 instruction and repetition modes of ITEM/IP and provides a unified adaptive view to the structured teaching material. Note that new version of ISIS-Tutor keep using two menu-based entries to the teaching material (for new and repeating material), but now it is a secondary way of access, a kind of index.

The student-driven navigation is tracked by the hypermedia component and the results are reflected in SM by increasing level of concepts went through. Therefore study a concept from hypermedia has similar meaning for SM as after selection from a menu of by the tutor. On the other hand while student works in hypertext, the Hypermedia component not only updates SM but also uses it for adaptation of concept presentation. It is a real integration of hypertext browsing and guided learning.

The environment is a tool for exploratory learning of the language and acquisition of procedural knowledge. Regular CDS/ISIS/M facilities for exploratory learning are quite bad. The user can see only one of the three plain things on the screen: the print format, that database records, or the results of printing. Also ISIS user usually gets formatted text as result of application of whole format to all database records and can hardly understand what some individual operator did. It may be good enough for regular work but it is very bad for a novice user exploring the language. It's very hard to learn from experience in ISIS, because the semantics of each print conlnland can't be understood well.

ISIS-Tutor environment was specially designed to support exploratory learning. Working with the learning environment the student can see in the same time a record of the sample database, a print format string, and the result of formatting the given record by the given format. User can change record (select other database record or create a new one) or change format (select other example from list suggested in concept presentation window, edit or enter his own format). Another important feature of the environment is ability to see execution result of each operator in format separately. In ISIS-Tutor environment student can execute the format step-by-step or command-by-command thus learning what changes the executed command adds to the output.

5 Lessons learned

We decided to design ISIS-Tutor to test our general ideas about integrating ILE and hypermedia and to check some particular features of a hypermedia-based ILE. We consider it as iuterinediate versions and do not plan to use it in present form in the real classroom. This work was really creative for its and we have the feeling that our general approach is really good and we are on the right way. The integration looks quite natural and really adds new quality to our old systems. Here we summarize main features of our approach to integrating a hypermedia component into an intelligent learning environment

- The central part of the hypermedia network is designed as the visualized and externalized domain nlodel network.
- All teaching operations are directly represented in the hypermedia network and interlinked with all related concepts.
- An external representation of a concept or a teaching operation is generated or assembled from its internal frame-based representation.
- The content of the hypermedia page is adapted to the student knowledge reflected in the student model.
- The hyper-links from general index and 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.

The most important lesson we learned from the experience with ISIS-Tutor is that hypermedia really brings new quality. It means that simple mapping of a prehypremedia system structure to the hypermedia-based system structure is usually not the best solution. Some part of the system should be redesigned and some new features can be added which can never be useful without the hypermedia.

For example, when designing a hypermedia-based ISIS-Tutor we applied the same way of student-controlled access to learning material as in ITEM/IP: two hierarchical menus "learn' and "repeat" for new and old material respectively. It works good in ITEM/IP, but in ISIS-Tutor it appears that these two separate menus for "new" and "old" concepts and local unified (coloured) menus of related concepts provided by each hyper-node give three different views to the same domain knowledge, what really creates some problem with understanding of the domain knowledge structure. In the next version of ISIS-Tutor which is being developed now we will replace "old" and "new" menus by a classic hypermedia feature: an index of hyper-nodes (concepts). We slake this index adaptive: the names of concepts are coloured according to the state of the student knowledge on it. It is consistent with the way of presenting a list of related concepts in each hyper-node.

An example of a new feature which can hardly be applied in a pre-hypermedia ILE is a hypermedia links from a concept to related examples. Note that examples and problems in ITEM/IP and ISIS-Tutor are stored separately from concepts and each example or problem was related to several concepts. It's really a complicated network, but hypermedia technology is very natural for it. Direct access from concept to examples appears to be very useful and we are extending this feature in the new version. At first, we will provide a similar direct access from a concepts to all kinds of related teaching operations - examples, tests, problems, etc. Working with any concept the student will be able to request an active list of teaching operations related with the given concept for any kind of teaching operations. The list will also be adapted and coloured to protect the student from bumping into a teaching operations related with unknown concepts and to give the student support in the navigation. We will provide also a reverse links: from teaching operations to all related concepts. This in turn provides an alternative paths of browsing the concept network: for example, from concept to a related example and from this example to another concept.

Finally, we should mention that there are some design problems which needs additional investigations. 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 ISIS-Tutor and dimmed in ITEM/PG which is another example of our hypermedia-based intelligent learning environments [4]. We can't say now which way is better. By hiding these links completely we can reasonably limit the cognitive load of the student, what is quite important for novices. However hiding any links looks unnatural for hypermedia. It can form incorrect mental model of the domain knowledge structure. 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.

6 Prospects

Described system is a working program that provides development service in print format language learning. But there are some aspects which can be looked as directions for further investigation and development.

In process of creation of problem domain model it becomes clear that domain is relatively complex and needs some more kinds of relations in the domain networking addition to prerequisite relations. Further, special structure of problem domain needs new types of concepts - not only for elementary concepts.

Adaptation should not be intrusive, and the student should be provided with a control over adaptation.

We will be working further testing and improving our approach. New versions of ISIS-Tutor are developing now which should solve some problems mentioned above. Generally, the approach we are using is now more or less stable. When we are designing a new system according to our approach, serious part of work is just repeating and re-coding the same things as we did designing more old systems. Thus now we are going further and starting the design of an authoring system based on our approach.

References

Brusilovsky P.L. (1992x) A framework for intelligent knowledge sequencing and task sequencing. In Frasson C., Gauthier G. and McCalla G.I. (eds.) Intelligent Tutoring Systems. Proceedings of the Second International Conference, ITS'92. Springer-Verlag, Berlin. 499-506.

[2] Brusilovsky P.L. (1992b) Intelligent Tutor, Environment and Manual for Introductory Programming. Educational and Training Technology International, 29(l), 26-34.

Brusilovsky P., Pesin L., and Zyryanov M. (1993) Towards an adaptive hypermedia component for an intelligent learning environment. In Bass L.J., Gornostaev J. and Unger C. (eds.) Human-Computer Interaction. Lecture Notes in Computer Science #753, Springer-Verlag, Berlin, 348-358.

[4] Brusilovsky P. and Zyryanov M. (1993) Intelligent tutor, environment and manual for physical geography. In PEG'93. Proceedings of the Seventh International PEG Conference. Edinburgh. 63-73.

Hammond N. (1989) Hypermedia and learning: Who guides whom? In Maurer H. (ed.) Computer Assisted Learning. Proceedings of the 2-nd International Conference, ICCAL'89. Springer-Verlag, Berlin. 167-181.

[6] Hendley R.J., Whittington C.D. and Juraschek N. (1993) Hypermedia generation from domain representations. Computers and Education, 20(1), 127-132.

Kibby M.R. (1991) Intelligent hypermedia for learning. In Bottino R.M., Forcheri P. and Molfino M.T. (eds.) Knowledge Based Environments for Teaching and Learning. Proceedings of the Sixth International PEG Conference. Geneva. 3-12.

Lajoie S.P. and Lesgold A. (1990) Apprenticeship training in the workplace: computer-coached practice environment as a new form of apprenticeship. Machine Mediated Learning, 3(1), 7-28

Pesin L.A., Brusilovsky P.L. (1992) Adaptive task sequencing in ISIS-Tutor. In: Abstracts of East-West Conference on Emerging Computer Technologies in Education, Moscow, 6-9 April 1992, Moscow. 176.

[10] Shute V.J. and Glaser R. (1990) A large-scale evaluation of an intelligent discovery world: Smithtown. Interactive Learning Environments, 1(1), 51-77.