PEG93

Proceedings of the Seventh International PEG Conference

A1 Tools and the Classroom:
Theory into Practice

Moray House Institute of Education
Heriot-Watt University
Edinburgh

July 2nd - July 4th 1993

ISBN 0 901580 57 0

INTELLIGENT TUTOR,
ENVIRONMENT AND MANUAL FOR PHYSICAL GEOGRAPHY

Peter Brusilovsky, 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

Introduction

Three classes of emerging computer technologies are now of great interest for specialists on **computer use** in education: **artificial intelligence, multimedia and hypermedia, and learning environments and microworlds. Traditionally developers of educational systems consider these approaches as independent or even contradictory ones. However the approaches are not contradictory but rather complimentary. Intelligent educational systems are able to control learning adaptively on various levels but generally do not provide tools to support free exploration and browsing through information. In comparison, hypermedia systems and microworlds provide a medium for student driven acquisition of declarative and procedural knowledge but they lack the control of an intelligent tutor. Without such control the student may never discover some important features of the microworld or the hyperspace, and may even get "lost in hyperspace".**

In some domains it is possible to achieve a good result by developing an educational computer system which integrates the capabilities of an intelligent tutor or coach, a learning environment or microworld, and a hypermedia system. This paper discusses the problem of integration and presents an approach for developing such integrated intelligent learning environments. To investigate the approach we have designed the Intelligent Tutor, Environment and Manual for Physical Geography (ITEM/PG), which is described in the paper.

ITEM/PG consists of an intelligent computer tutor, a hypermedia network and an environment for learning of relationships between components of the natural complex of an island. A **structural domain model and an overlay student model are used as a kernel for integration of different components of the system. ITEM/PG combines goal directed adaptive task sequencing provided by the intelligent tutor and student managed exploration and learning of the subject supported by the environment and the hypermedia network. ITEM/PG shows a way to integrate these computer technologies on the basis of represented knowledge about the domain and the student.**

Towards an integration of emerging computer technologies

Recent research has demonstrated that microworlds and 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 (Shute& Glaser, 1990; Hammond 1989). The guidance is often an important ingredient of effective learning from microworlds and hypermedia. The guidance can be provided by an intelligent tutoring system.

This consideration stimulates the research towards an integration of these emerging computer technologies. Good results have been achieved in integration of microworld environments with intelligent tutoring systems (ITS). Smithtown (Shute & Glaser, 1990) and Sherlock (Lajoie and Lesgold, 1990) provide good examples of intelligent learning environments. A number of researchers are working now on intelligent hypermedia (David, Thiery & Crehange, 1989; Kibby, 1991). Several educational systems which integrate knowledge-based tools and hypermedia interface have been reported hypermedia interface was reported (Briggs, Tompsett & Oates, 1991; Nicolson & Tomlinson, 1991; Olabe et al, 1991; Wang & Kushniruk, 1992) and general architectures for developing such integrated systems have been suggested (Beltran & Peninou, 1992; ElHani & Gouarderes, 1992).

For a number of years we have been investigating the problem of integration of an intelligent tutor, a microworld and an on-line help system in the domain of teaching programming. We analysed a number of existing systems which combine ITS and microworld features, and learned that most of these systems are more boxes of independent tools than really integrated environments. Our position was 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. Secondly, 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.

To investigate the ways of building an integrated intelligent programming environment we have designed and tested the Intelligent Tutor, Environment and Manual for Introductory programming (ITEM/IP). ITEM/IP (Brusilovsky, 1992b) 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 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 (items) of domain knowledge and relations between the elements. The domain model states the basis for the overlay student model. The student model variables corresponding to any particular node or link represent the level of the student's competence of the knowledge element or relation. The results of students' work with the tutoring component are reflected in the overlay student model and used by other system components to adapt themselves to the knowledge level of the particular student.

One of the goals of our "ITEM" project was to build a system where most of the modules can use the same student model to adapt their performance in various ways to the knowledge of the given user. We have specially investigated methods of adaptive curriculum sequencing which is one of the ITS functions (Brusilovsky, 1992a) and methods of developing an adaptive interface for intelligent learning environment (Brusilovsky, 1993).

Now we are working on extending our "ITEM" -approach in two directions. First, we are trying to apply the approach to other domains. Second, we are trying to integrate the hypermedia technology into our "ITEM" environments (in both senses mentioned above). As a primary testbed we have selected the domain of Physical Geography. This domain is not completely new for us: previously we have designed and reported a computer environment for Physical Geography (Brusilovsky & Gorskaya-Belova, 1992). The next section presents the ITEM/PG system for the same Geography domain which integrates the capabilities of intelligent tutor, learning environment, and hypermedia system.

To integrate the hypermedia network into the ITEM/PG environment we take the domain model network as a basis for the hypermedia network. Such ideas have been discussed previously (Kibby, 1991; Hendley, Whittington & Jurascheck, 1991) and we consider these ideas as important. This approach enables us 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 details on using the student model in ITEM/PG are given in the closing section.

The domain and domain knowledge

ITEM/PG environment is for 13-14 years 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 different components of a natural complex. The students work with an island, located somewhere in the north part of -the Atlantic ocean.

The island is characterized by the following components of its natural complex: position, origin, structure of the ocean bed, tectonic structure, climate, type of crust, soil, flora and fauna. Each component has several possible values. For example, the options for the origin of the island are continental, volcanic, ruff or atoll. The values of the components for the given island are related to each other. For example, the climate depends on the position of the island, the origin of the island depends on the type of the Earth's crust and so on. The goal of the student is to learn the components of --the natural complex, their possible values, the relations between the components, and finally to acquire the very idea of the relationships between different components. More detailed description of the domain and pedagogical basis for the overall project can be found in (Brusilovsky & Gorskaya-Belova, 1992).

As we can see, the domain to be studied is quite simple. It includes knowledge about different components of the natural complex of the island, knowledge about its values, including such attributes as name, description and standard pictogram for each possible value, and knowledge about relations. The latter consists of rules, giving the different natural complex relations. These rules can be represented in the standard IF-THEN form (figure 1). There are about 60 rules, obtained from several experts in the field of physical geography.

IF the structure of the ocean -bed is a continental rise,
THEN the type of the Earth's crust is continental,
AND the tectonic structure of the island is a platform.

IF the type of the Earth's crust is continental,
THEN the origin of the island is continental.

IF the tectonic structure of the island is a platform,
THEN the relief of the island is a plain.

Figure l. Some rules from ITEM/PG knowledge base

The above kinds of knowledge form the domain model network. The basis for the network is the three-level tree, representing the hierarchy: island - > component - > value. This tree includes two main types of nodes - component node and value node - and two kinds of links. The third kind of the domain network node is the rule node. The rule node is related with all the nodes of the "main tree" which are mentioned in the rule. For example, the rule "IF the tectonic structure of the island is a platform, THEN the relief of the island is a plain" is related with the component nodes "tectonic structure" and "relief" and the value nodes "platform" and "plain". (We distinguish four kinds of links here: condition-component, resulting-component, condition-value, and resulting-value.) The amount of 60 existing rules gives a very complicated network with three kinds of nodes and six kinds of links, where the component nodes and value nodes are variously transitively interrelated through the rule nodes. The domain network is represented in ITEM/PG system in the form of a frame network, where each node is represented by a frame.

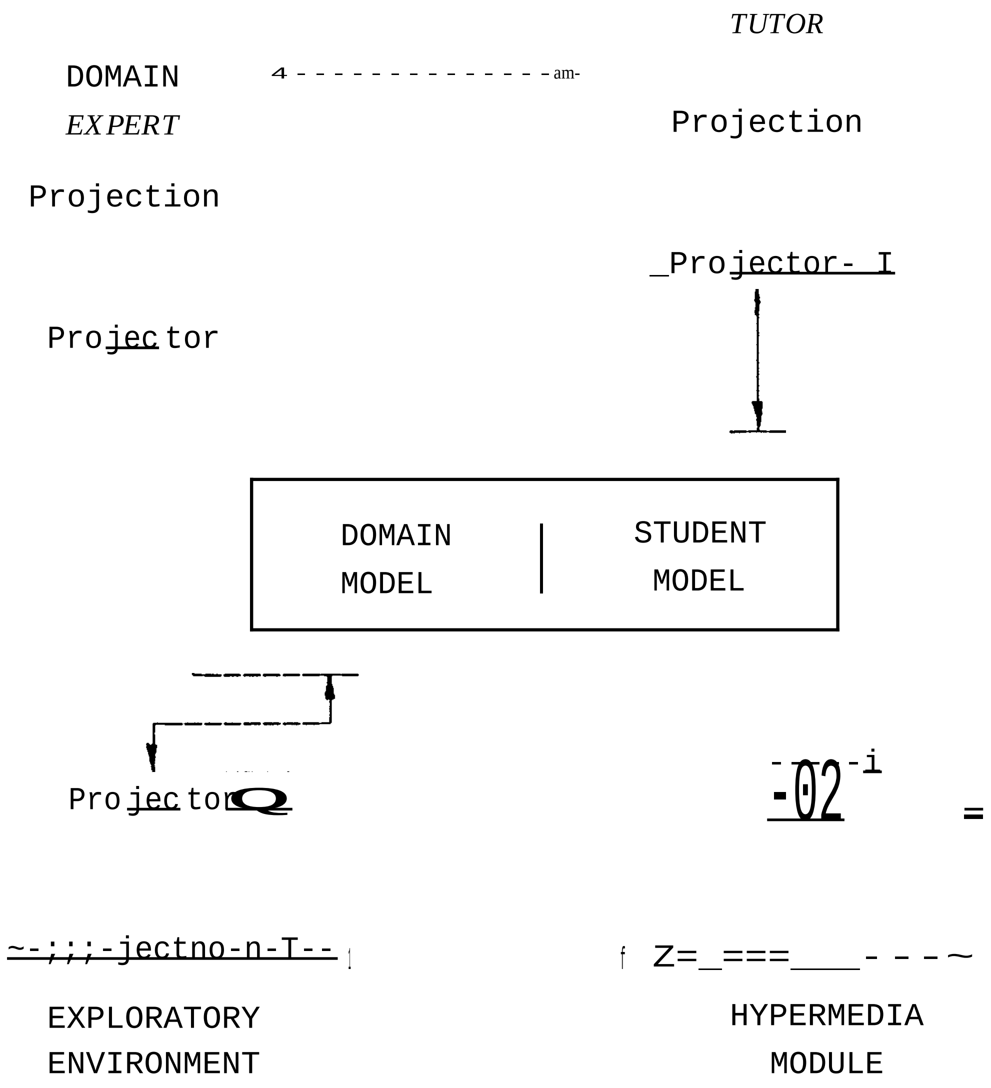
An intelligent environment for geography

The domain model is the central component of the system. Other main components of the ITEM/PG environment are the student model, the tutor, the learning environment and the hypermedia network (figure 2).

The overlay student model is based on the domain network. For each node of the domain network the student model represents the results of the student's work with this node (including work with the environment, the hypermedia subsystem and the tutor) and an assessment of the student's knowledge about the node. The amount of detail of this representation depends from the kind of node.

The hypermedia network is also based on the domain network. More exactly, the hypermedia network is just the visualized domain network. Each node of the domain network is represented by a node of the hyperspace, and the links between domain network nodes give all possible paths between hyperspace nodes. There are three main kinds of hyper-nodes: component, value and rule. Each kind of hyper-node has a special screen representation. It's important, that screen representations for the hyper-nodes are not stored in the screen format, but generated from the corresponded frame by a special program. For example, the program for the "value" kind of node generates the window which contains the name of the value, the description, the pictogram and the hyper-links to the component node and related rule nodes.

Figure 2. The Architecture of ITEM/PG



The hypermedia network provides the student with the tool for exploratory learning of the conceptual domain knowledge. The student can browse this network along the links learning about the components, values and rules. The hypermedia network does not provide the tool for learning of procedural knowledge, i.e. knowledge how to apply the rules to find the unknown values of the components of the natural complex. Such a tool is provided by the environment.

The environment component of the ITEM/PG is a modified environment described in (Brusilovsky & Gorskaya-Belova, 1992). Working with the environment the students find themselves on an island, the position of which can be chosen by the student on the map displayed on the screen. Knowing the position of the island, the student must determine the values of the other components of the island. To assist the student the environment provides a friendly interface and an intelligent checker.

To select the value of each of the island components the student uses pictographic and text menus, where each value of the natural complex components is represented by its name and pictogram. Pictograms are used also to form so-called report window, which is the compact representation of the problem-solving situation. The report window contains a small map showing the island's position and one pictogram for each component. If the value of a component has already been selected by the student, then the corresponding pictogram is used. For the remaining components question marks are displayed.

To check the correctness of the choices the student can call the intelligent checker. The checker uses the domain rules and the inference engine to verify the student choice. To do this, the checker solves the same problem comparing its own answers with the student's step by step. The checker evaluates the student's choices as correct or wrong, checks whether the logical inference was right and determines what lack of knowledge caused the student's current mistake. Thus the checker assesses the student's ability to apply the domain procedural knowledge, as well as the ability to think logically. The checker can generate error reports and explanations in natural language. Figure 3 gives an example of such a report which concerns some rules presented in Figure 1. The report can also be presented visually on the report window by crossing out the icons of incorrect selections.

```
The type of the Earth's crust is oceanic - an error! If the
structure of the ocean bed in the island location is a
continental rise (platform), then the type of the Earth's
crust is continental. The relief of the island is low mountains
- an error! If the tectonic structure of the island is a
platform, then the relief of the island is a plain.
```

Figure 3. The example of error report generated by the checker

The environment provides the tool for the exploratory learning of the procedural domain knowledge. Selecting an island, trying to determine its natural complex and then evaluating the choices, the student makes an experiment in the domain studied. This style of work enables the student to discover and master the domain model rules. (Note that the hypermedia component enables the student only to read ABOUT the rules, while to MASTER a rule the student need to apply it several times).

Both the hypermedia component and the environment on its own are exploratory learning tools which lack intelligent guidance. Without such guidance the student can have problems with learning the complex interrelated domain knowledge and spend too much time in ineffective exploration. For example, browsing the hyperspace, the student can avoid by chance some important nodes. Selecting the island here and there in the map, the student by chance can avoid some positions which are essential to master a subset of domain rules.

The intelligent tutor solves the problem of ineffective exploration by providing the required guidance. At any moment of work with the system the student can ask the tutor for the next optimal teaching operation. The tutor analyses the student knowledge represented in the student model and decides what to teach next and how to teach it. The tutor uses the following teaching operations: providing a relevant piece of the conceptual knowledge by addressing the student to the selected node of the hyperspace; checking the acquisition of a piece of conceptual knowledge by asking a question; demonstrating how to apply the newly learned rules by selecting the relevant island and solving it for the student step by step; posing a problem which requires the application of the optimal mixture of mastered and not mastered rules. The latter two kinds of teaching operations are supported by the tutor's ability to select the relevant island for any noncontradictory goal set of rules, i.e. to select the island which requires the application of the these goal rules to be solved. The tutor selects or generates the next operation and presents it to the students with the help of the hypermedia component (for conceptual knowledge), or with the help of the environment (for the posed problems and demonstrated problem solutions). Note that the tutor does not have its own user interface.

Working with the system, the student use either the environment, or the hypermedia subsystem. The student can work in exploratory mode, browsing the hypernetwork or experimenting with the environment. At any given moment the student can apply to the tutor for the next "optimal" teaching operation, and then follow the tutor's suggestion or give up. Thus ITEM/PG can support a continuum of learning styles: from pure exploratory learning without any application to the tutor to pure guided learning, when the student asks the tutor's advice at each step of work. Students themselves determine the proportion between guided and exploratory learning, as well as between hypermedia exploration and environment exploration. The student can easily change the mode of the work moving from guided to exploratory mode, from the environment to the hypermedia subsystem and vice versa. Such a style of work in ITEM/PG is possible because all the main components support and use the same picture of the student's knowledge, represented in the student model. The next section presents some details about the role of the student model in ITEM/PG and how it is used by the components to adapt their performance to the given student.

Student model centred architecture for an intelligent learning environment

Classic four-component architecture for ITS (Wenger, 1987) implies that the student model is kept updated by the student module (or diagnostic module) and used by the tutoring module. In our previous ITEM/IP project we have tried to make as many components of the systems as possible adaptive on the basis of the student model. The general goal for the ITEM/PG project was to design and investigate the architecture for a system, where all the system components may update the student model to reflect the results of the student's work and may use it to adapt themselves to the particular student.

The student model in ITEM/PG environment stands together with the domain model in the centre of the environment. The main student model collects and unifies information about the given student from different sources. Any student interaction with any of the system components is reported to the student model. These reports are time stamped and stored in the context of the domain model (i.e. pieces of information about the student are directly related to domain model nodes). Examples: "at time Tl the student visits the hypernode for the component Cl", "at time T2 the student was presented with the task solving example which concerns components C1 ... Cn, values V1 ... Vn, and demonstrate rules Rl ... RV "working free with the environment at time T3 the student select value V for the component C thus applying rule R correctly". The main student model unifies all the information about the student which can be used for the purpose of adaptation.

The components of ITEM/IP do not use the student model directly, but instead use local views on the student that we call projections. A projection represents the information about the student, which is essential for the component to adapt its work to the student. A component has as rich and wide a projection, as it can use for the purpose of adaptation. A projection is filled and updated from the main student model by a special set of rules called a projector. Each component has own projection and projector, which provides the interface between the projection and the main student model. One part of the projector rules is used to project the main student model into the local projection. These rules refer to the student model in their left sides and contain commands to update the projection in their right sides. Another part of the projector rules is used to provide back projection: to project the results of the student's work with the component into the form used by the main student model.

We think that the proposed student model centred architecture is a good basis for creating an integrated learning environment, as well as any other integrated adaptive system which consists of a set of different components. The use of projections and rules provides the open architecture with a good degree of flexibility. A new component can be easily integrated into the environment by designing a set of rules which connect the central model with the given component and its local view on the student. Since the performance of the component depends on the projection, we can tune the performance for the particular use by changing the projector rules (or even the projection itself) without any influence to the other components.

The proposed architecture differs from classic ITS architecture, where the student modelling (diagnostic) component projects the information about student directly into a final form represented in the overlay of a "buggy" student model and used by the tutor or coach. From our point of view the student model of a classic ITS is just one of the local projections: the one used by the tutoring component. Other components of the system (such as the microworld) can use quite different projections. The main student model in ITEM/PG stores partly processed

information about the student, because further processing can lose information important for one of the components. The ITEM/PG student model is more than a traditional "history", but less formalized than a classic overlay model. Further processing and projection to more traditional overlay form is made separately by the projectors according to the requirements of different component.

The simplest projection is used by the environment component of ITEM/PG. It distinguishes two states only (known and not known) for components and values and three states (well-known, known and not known) for rules. We applied the following general principles of interface adaptation (Brusilovsky, 1993) to ITEM/PG:

- **Never mention (hide) unknown domain knowledge elements.**
- **Provide more feedback and explanations for new elements.**
- **Be concise in dealing with well-learned elements.**

According to these principles, the environment working in adaptive mode hides the unknown components and values in the menus. Thus the student is expected to operate with known components and values which have been learned either from hypermedia or from the tutor. When checking the student choice the checker generates different diagnostic messages according to the state of the rule which caused the error. The message for a well-known rule is more concise. than general. Vice versa, for a not yet known rule the student is additionally suggested to visit the hyper-node with the description of the rule.

The projection for the **hypermedia component distinguishes four states for components, values, and rules: not ready to be learned, ready to be learned, known and well known. It is similar to the environment view for the rules, but the hypermedia component divides further not known elements into two classes: ready and not ready to be learned. A hypermedia node is ready to be learned if it does not have the unlearned prerequisite knowledge elements. For example a particular value is not ready to be learned until we learn the corresponding component itself. The rule is -not -ready--to-be learned until we learn all the values and components used by the rule. This information is used by the hypermedia component to to divide all references from the current node to the related nodes into four classes by visually marking them. These visual marks give the student guidelines in the hypernetwork browsing. For example, the links to the nodes which are not ready to be learned are dimmed or hidden so as not to distract the student. The ready to be learned nodes form the "zone of proximal development" (Vygotsky, 1978) and are colored green inviting the student to visit it.**

Thus ITEM/PG provides the adaptive hypermedia with high level adaptation based on the unified student model. We think that this approach is more powerful than the more 'historic' approach to adaptive hypermedia discussed by de La Passardiere & Dufresne (1992). However we need special classroom experiments to evaluate the effect of adaptive hypermedia. Another feature that requires testing concerns links to not ready to be learned nodes. By hiding these links completely we can reasonably limit the cognitive load of the student, but such adaptation is more intrusive than just dimming these links. We plan to compare hiding vs dimming in an experiment.

The most detailed projection is used by the tutoring component. It distinguishes three states for components, values, and pictograms related with each value(unknown, known but unlearned, learned). For the rule node, which unifies the conceptual and procedural pieces of knowledge the tutor distinguishes several levels: unknown, know-

about, know-how-to-apply, can-apply and mastered. The knowledge state for the given rule is used to decide what kinds of teaching operations are relevant to work with this rule and what kind is optimal. Thus, presentation is optimal for unknown rules, demonstration is optimal for know-about, and -so on. ITEM/PG use about the same structure of tutoring module and the same algorithms as ITEM/IP, which are described in detail in (Brusilovsky, 1992b). The only difference is that ITEM/IP always selects the optimal teaching operation from the prestored set, while ITEM/PG is able to generate problems and demonstrations by selecting an island which requires the application of the optimal mixture of mastered and not mastered rules.

The methods of adaptation used in ITEM/PG are relatively simple, as is the domain itself. Our goal was not to improve the known methods of adaptation, but to build a system where all the components can use the same student model to adapt their performance in various ways to the knowledge of the given student.

Conclusion

We describe the system ITEM/PG for the domain of Physical Geography which integrates the features of intelligent tutor, learning environment and hypermedia system. The system was developed to study the problem of integration of these emerging technologies, in particular to investigate the ways of integrating the hypermedia component into the environment and connecting it with the student model. The main contribution of this work in the context of our previous research in the domain of introductory programming (Brusilovsky, 1992a,1992b) is the development and application of a new student model centred architecture for intelligent learning environments and the development of an adaptive hypermedia system integrated into the overall environment. We think that the architecture suggested and the methods of development for such adaptive integrated hypermedia are general enough to be applied -outside-the domain of ITEM/PG. However, further investigations are required to test the usability of the architecture and the methods for the purpose of computer supported learning.

Acknowledgement

The authors would like to thank Mary Mark for her comments to an earlier version of this paper.

Ref erences

Beltran Th. and Peninou A. (1992) An object-oriented approach to produce educational hypermedia software. In Tomek I. (ed.) Computer Assisted Learning. Proceedings of the 4th International Conference, ICCAL'92. Springer-Verlag, Berlin. 111-123.

Briggs J.H., Tompsett C. and Oates N. (1991) Guiding learners through a medical hypertext database using knowledge based rules. 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. Genova. 571-577.

Brusilovsky P.L. (1992a) 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.

Brusilovsky P.L. (1992b) Intelligent Tutor, Environment and Manual for Introductory Programming. Educational and Training Technology International, 29(1), 26-34

Brusilovsky P.L. (1993) Student as user: Towards an adaptive interface for an intelligent learning environment. Paper accepted to AIED'93 World Conference.

Brusilovsky P.L. and Gorskaya-Belova T.B. (1992) An environment for physical geography teaching. Computers and Education, 18(1-3), 85-88

David A.A., Thiery O. and Crehange M. (1989) Intelligent hypermedia in education. In Maurer H. (ed.) Computer Assisted Learning. Proceedings of the 2-nd International Conference, ICCAL'89. Springer-Verlag, Berlin. 53-64.

de La Passardiere B. and Dufresne A. (1992) Adaptive navigational tools for educational hypermedia. In Tomek I. (ed.) Computer Assisted Learning. Proceedings of the 4th International Conference, ICCAL'92. Springer-Verlag, Berlin. 555-567.

ElHani O. and Gouarderes G. (1992) Standardized architecture for integrated open coursware. In Tomek I. (ed.) Computer Assisted Learning. Proceedings of the 4th International Conference, ICCAL'92. Springer-Verlag, Berlin. 198-211.

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.

Hendley R.J., Whittington C.D. and Jurascheck N. (1991) Generating hypermedia documents from domain knowledgerepresentation. In Bottino R.M. and Forcheri P. and Molfino M.T. (eds.) Knowledge Based Environments for Teaching and Learning. Proceedings of the Sixth International PEG Conference. Genova. 595-602.

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. Genova. 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);-9-28

Nicolson R.I. and Tomlinson P. USHIR: An intelligent knowledge-based hypermedia system. Hypermedia, 3(1), 1-33

Olabe M.A. et al (1991) Teaching credit management using expert system and hypermedia software. 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. Genova. 616-621.

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

Vygotsky L.S. (1978) Mind in society. Harvard University Press, Cambridge.

Wang H. and Kushniruk A. (1992) The UNIX tutor. In Frasson C., Gauthier G. and McCalla G.I. (eds.) Intelligent Tutoring Systems. Proceedings of the Second International Conference, ITS'92. Springer-Verlag, Berlin. 317-324.

Wenger E. (1987) Artificial intelligence and tutoring systems. Computational approaches to the communication of knowledge. Morgan Kaufmann, Los Altos.