Proceedings of the IDPT

SSRSt99

From the User's Needs to Adaptive Documents

Cornelia Seeberg and Achim Steinacker and Klaus Reichenberger and Abdulmotaleb El Saddik and Stephan Fischer and Ralf Steinmetz

In this paper we describe an adaptive hypermedia learning system called Multibook. First, we describe the needs of the users which we take into account, then we explain the knowledge base of the system containing both actual pieces of knowledge and meta-information. At last, we show how this knowledge base can be used to generate individual lessons for the diverse learners according to their user profile.

BibTeX entry

Important Copyright Notice:

In Proceedings of the INTEGRATED DESIGN & PROCESS TECHNOLOGY "IDPT'99", June 1999. will appear in the IDPT 2000 proceedings.

From the User's Needs to Adaptive Documents

Cornelia Seeberg[1,2], Abdulmotaleb El Saddik[1], Achim Steinacker[1], Klaus Reichenberger[1], Stephan Fischer[1] and Ralf Steinmetz[1,2]

1
Industrial Process and System Communications
Dept. of Electrical Eng. & Information Technology
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany

2
GMD IPSI
German National Research Center
for Information Technology
Dolivostr. 15 • D-64293 • Darmstadt •
Germany

{cornelia, stein, reichen, abed, fisch, rst }@kom.tu-darmstadt.de

ABSTRACT

In this paper we describe an adaptive hypermedia learning system called Multibook. First, we describe the needs of the users which we take into account, then we explain the knowledge base of the system containing both actual pieces of knowledge and meta-information. At last, we show how this knowledge base can be used to generate individual lessons for the diverse learners according to their user profile.

INTRODUCTION

Multibook is a Web-based adaptive hypermedia learning system with a focus on multimedia elements, especially interactive simulations.

The growing WWW makes it necessary to provide more individualized mechanisms for delivering information to the users. This means that the documents are either generated (see for example [11]) or linked on-the-fly. Adaptivity is the way to avoid of either writing several times a document to meet the needs of the diverse users or offering each user the same document. Especially for learning systems which are supposed to be used from different classes of learners without help from a teacher, it is important to support the users individually.

The Multibook system enhances the idea of guided tours. The system generates on-the-fly individual lessons for each user with a restricted possibility to obtain additional material. This way we overcome the gap between static guided tours ("one-size-fits-all") and a traditional hypertext system with its often cited problem of "lost-in-hyperspace".

Let us observe authors supposed to write some educational text concerning a certain topic. The first thing they do is to consider their audience. With this in mind, they will have a brainstorming session and collect all the headwords which are necessary to teach the topic. Giving them an order means to have an outline for the text, a table of contents. Having completed this, the next task is to find suitable material to fill the outline: text paragraphs, images, maps, graphics, tables, tests. This choice also depends on the needs and preferences of the audience. Teachers giving a class have the advantage to use other kind of material, such as experiments, videos, role plays.

They also can select some pieces as additional issues in case the audience is bored or overtaxed. Or they can change the concept having prepared an alternative set of material.

It is ambitious to expect that a machine can do such an intelligent work, since teaching has been a complex cultural technique, and to become a teacher means long years of learning her/himself, and dedication. But we can try to implement algorithms which are orientated on the behavior of a good author or teacher in order to achieve a learning system which extends the possibilities of a traditional book.

RELATED WORK

Recently, ideas from intelligent tutoring systems (ITS) and from hypermedia have been brought together opening the field of adaptive hypermedia. This synthesis responds to the specific strengths and weaknesses of both approaches. ITS [1], [6], [17] provide a high level of guidance in learning, they model and control the entire learning process in great detail. A domain model representing all facts of the field that are to be learnt usually forms the background for a model of the learner's knowledge and knowledge acquisition. Free exploration by the learner does not play an important role in ITS, the navigation decisions are very much up to the system. ITS tend to anticipate all learning paths, i.e. states of the system and their implications for the learner's mental model. Therefore, the complete material that is presented to the learner must either be included in prefabricated lessons, tailored for communicating a specific part of the domain model in a specific learning situation. Or it must be automatically generated from the domain model. In text based domains, where text is the most important medium for instructional content, this approach is problematic, even if a long-term solution is offered by sophisticated natural language generation technology [2]. As discussed in [5], most of the existing ITS systems are environments for coding and testing a specific programming language. Furthermore, users often need considerable time for learning how to use an ITS, because the development of an intuitive user interface for ITS systems seems to be a very difficult task.

Hypertext and hypermedia systems exploit the nature of different media such as text, pictures, audio, video or simulations as a medium for making differentiated statements and communicating less structured knowledge. In addition, a hypermedia system offers more than predefined learning paths - by selecting different nodes in different order, the individual learners produce a multitude of paths through the material. The drawback of these systems is that the learning process cannot be controlled in a well defined way. This results in insufficient guidance. In particular when used for educational purposes, hypermedia systems are striving for a higher degree of control [9], [4]. We regard as decisive the step of adding conceptual information on top of the hypermedia chunks, being the basis for their intelligent selection and sequencing. Connecting the concepts with semantic rather than didactic relations that would already imply sequences or dependencies among the concepts [3] provides yet a higher degree of flexibility - this way, the concept space lends itself to realizing different learning strategies and goals but also to tasks like information retrieval.

The InterBook project is a well known adaptive hypermedia system, which also makes use of a concept space, called domain model [18], [7]. InterBook supports navigation through the lesson with adaptive annotation, showing the type and the educational state ("ready to be learnt", "recommended", "not ready to be learnt") of each offered link. InterBook is based on the domain modeling approach of ELM-ART, a WWW version of ELM-PE, that is currently one of the most advanced intelligent learning environments for programming. In the online description of InterBook the term "I3-textbooks" (integrated + interactive + intelligent textbook) is proposed for approaches which integrate the on-line representation of learning material with the interactivity of problem solving environments and intelligence of ITS. In this sense, the Multibook project can be seen as a specific type of an I3-textbook, although our application domain brings along some different problems which requires slightly different approaches and techniques, which we will describe in the remainder of this article.

THE USER PROFILE

The content of the Multibook system, currently being developed at the Darmstadt University of Technology, is based upon the printed book "Multimedia: Computing, Communications & Applications" by Ralf Steinmetz and Klara Nahrstedt [15] consisting of about 1000 pages, and a selection of Java applets. The aim of Multibook is to have individual views on this material according to the needs and preferences of the individual users, realized in different lessons for different users.

The needs Multibook can meet are categorized in four dimensions:

- **Learning aim:** Here the users can specify their role, whether they are students, programmers or managers in the position to decide to buy a media server or to hire a programmer to implement JPEG etc.
- **Background Knowledge:** In Multibook the lessons are available in six different levels of difficulty.
- **Teaching method:** For the time being Multibook offers two different teaching methods. A very structured lesson following a hierarchical teaching method, and a lesson with problem oriented motivation.
- **Content type:** Here the users can select the kind of media they want to obtain in their lesson and those they definitely don't want to be included. (Surely, there will be restrictions due to the fact that the main medium still is text.) We consider this point as important for three reasons:
  - Pedagogical: For some users it is easier to learn from pure text, for others pictures etc. are helpful for a better understanding.
  - Technical: It doesn't make sense to send videos to a user who is connected to the server via a modem.
  - Social: With the opportunity to state the media preferences the system can select suitable material for handicapped users. A deaf user, for example, doesn't get an audio file, but some

visual material. This is not a makeshift, but a real alternative.

These are the four dimensions of the user profile within Multibook. In the beginning, the profile is filled with the demands and preferences of the users. While the users are working with Multibook, the system keeps track of what information they have already seen/learnt, what additional material they have demanded to see, the results of the tests etc.

To be able to generate lessons according to these four dimensions of the user's needs, the knowledge base of the systems has to provide meta-information.

MUKTIBOOK'S KNOWLEDGE BASE

There are two spaces where the information and the meta-information are modeled: a domain model, which we call the ConceptSpace, and the MediaBrickSpace, where knowledge is represented in small modular pieces of text, images, audio, video, or animation. There are objects (concepts and media bricks), relations (semantic in the ConceptSpace and rhetorical in the MediaBrickSpace) and attributes of the media bricks. The concepts and the actual content are therefore separated. While a teaching method reflects the concepts and especially the order in which concepts are displayed, the actual content represented by the media bricks transforms the user preferences into a suitable document. In other words, the ConceptSpace represents the domain, the MediaBrickSpace is a set of possible explanations of this domain. As we have mentioned, formalization of the content to be learned lies at the core of the Multibook approach. However, we cannot aspire to substitute or completely control the process of learning from text or multimedia material rendered for human consumption and typically not available in a formal description.

Our approach is to compose individualized lessons using the network of facts and concepts from the subject domain (ConceptSpace) as an access structure to a body of text and multimedia material rather than to generate lessons directly by turning facts from the domain model into sentences, diagrams, questions and tests.

Being used as an index to the original learning material, an approach much closer to publishing practice, related to structuring and re-using assets etc., the facts in the ConceptSpace will not mirror each assertion that is made in the media bricks. It is of crucial importance for the approach to adequately choose the level of granularity, i.e. the detail and extent to which the subject field (and the learner's knowledge about it) is modeled. As a consequence of not formalizing in detail every statement of each media brick, the information we have about which bricks to combine into a lesson is a very general thematic information. If we want to take into account additional characteristics of the media bricks, e.g., their level of difficulty, or how the bricks fit into a certain argumentative strategy this information has to be given explicitly as attributes. We also established relations between the media bricks, these relations are not semantic, but rhetorical.

These considerations influenced the modeling of both the ConceptSpace and the MediaBrickSpace which will be described in the following:

ConceptSpace

As a rule of thumb, we choose the level of granularity such that objects of the model identify topics that could serve as chapter headings in a multimedia study book. The difference to traditional chapter headings is evidently that the information is held independently from the texts and other media bricks, with the effect that the topics are "centralized" instead of being scattered across the body of material. Formally, this is realized as an entity relationship model, where each object appears only once and accumulates information. Conversely, every topic that is to be related with some other piece of information has to be modeled as an object. While a few instance objects may be relevant (some concrete person or company, etc.) the main part of the network will be formed by abstract concepts.

Such a focus on topics/concepts suggests that our ConceptSpace will be a terminological ontology rather than an axiomatized ontology, i.e. an ontology where concepts and relations have associated axioms and definitions that are stated in logic or some computer oriented language that can be automatically translated to logic. [14]. We have, however, to draw a clear distinction between the generic and the partitive hierarchy relation (a distinction, the necessity of which is discussed in the area of domain modeling [8]). We are building upon this distinction when identifying, e.g., a component (AEpart) of a media server, a certain type of file server (generic) as a topic to be explained but want to start the explanation of the component, say a disk controller, with principles applying to controllers in general. Up to a certain point, deciding whether talking to the learner about controllers at a given stage of the file server lesson makes sense, assumes a conclusion across a path composed of different relations. To this end, we found it necessary to look at the AEpart relation as a statement relating the set of instances of one concept with the set of instances of the other and to additionally quantify this statement. In general, it must be possible to query the network starting from arbitrary points. This makes it necessary to also think of the terminological relation that we use as statements/assertions that have to be correct no matter which way you read them. They may not be based on implicit assumptions in which order the material is presented or any other context information.

We employ the following set of relations for the ConceptSpace:
Note: These relations and constructions are necessary to create the view of a learner. For other types, such as manager, more relations are needed. As we only intend to offer learning documents for students, we omit a more detailed list of relations here.

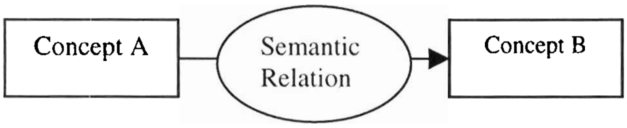

Fig. 1: Basis type of semantic relations

Superconcept: A is the superconcept of B means that A is the broader term of B. superconcept is a transitive, non symmetric, non-reflexive, hierarchical relation. Example: compression/decompression procedure is the superconcept of image compression. Inverse Relation: subconcept
AEpart: A has a AEpart (For All, there Exists part) B means that every kind of A (i.e. all its subconcepts) has B as a part. AEpart is a transitive, non symmetric, non-reflexive hierarchical relation. Example: compression/decompression procedure has decompression as a part. Inverse Relation: invAEpart
Note: There are other part-relations, such as AApart (for each A are all kinds of B parts) but in the Multibook system only the AEpart relation is employed. This relation could also be read as "compression/decompression procedure uses decompression".
EEpartOf: A is EEpartOf B means that a subconcept of B is part of at least one A. EEpartOf is a transitive, non symmetric, non-reflexive hierarchical relation. Example: compression is AEpartOf compression/decompression procedure. Inverse Relation: invEEpartOf
InverseProcedure: A is inverseProcedure of B means that A is the inverse procedure of B. inverseProcedure is a non transitive, symmetric, not generally reflexive relation (i.e. in most cases it is not reflexive, but there are procedures which are the inverse of themselves).
Example: Compression is the inverseProcedure of decompression. Inverse Relation: inverseProcedure
Follows: A follows B means that directly after concept A comes concept B, e.g. A, B are successive steps in a procedure. Follows is a non-transitive, non-symmetric, non-reflexive relation. Example: image processing follows image preparation when talking about JPEG. Inverse Relation: precedes
ProblemSolution: A is a problemSolution for B means that A solves a problem which arises due to B. Example: Compression is a problemSolution for the problem storage space

The next two relations are formally of a a different type.

Partition: A has a partition with regard to a hierarchical relation with a certain partition aspect means that the related concepts (partitionElements) build a partition of A, i.e. the partitionElements are distinct and all partitionElements together are A. The functionality is illustrated in Figure 2. Example: vector graphics and pixel images is a partition of image with respect to the aspect of the representation format.

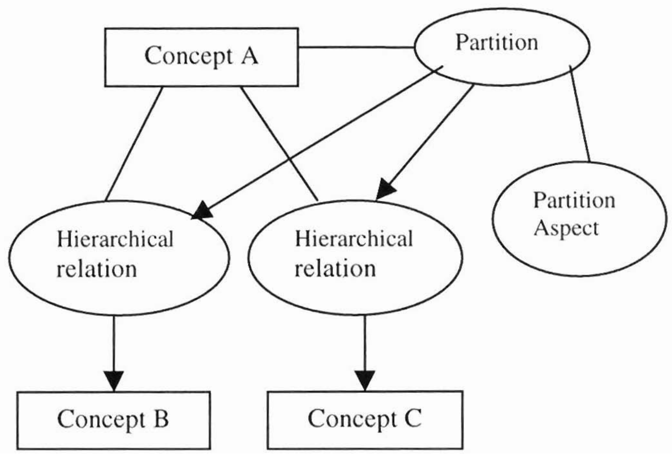

Fig. 2: "Partition" relation

Difference (Rahmstorf Relations): [13] A is difference of B means that there is a superconcept C of B and B is specified from C by the aspect of A. difference is a non transitive, non symmetric, non-reflexive relation. An example for difference can be found in Figure 3. Example: text is the difference of text compression with respect to the superconcept compression. Inverse relation: differenceOf

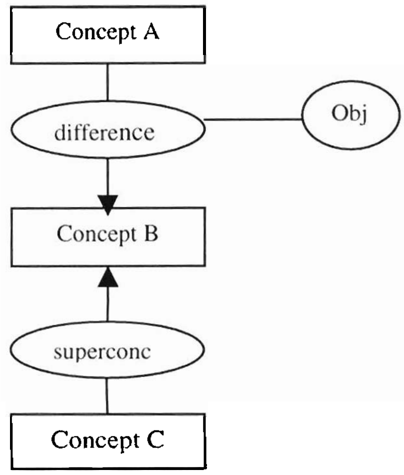

Fig. 3: "Difference" relation

There are different possibilities to choose a set of relations. The granularity of the model relates to the fact that we are interested in roughly structuring the domain rather than completely representing all facts. In the vocabulary of Knowledge Engineering, this makes our ConceptSpace a terminological rather than an axiomatic ontology [14]. We are aware of the benefits that the extended use of verb concepts, and the expression of a greater variety of facts they allow, could have for our domain. However, the current model is, for simplicity reasons, mainly built around noun concepts [8]. Semantic relations strongly guide the composition of lessons but do not predetermine a unique set of topics and the order of these, to map a single order of specific topics onto a teaching method.

MediaBrickSpace

Media bricks can either be text or other multimedia elements such as images, graphics, video and audio streams and - with the main focus - animations implemented as Java applets. Also these multimedia elements satisfy the requirements of modularity, this means that it is necessary that the format of the media bricks enables the system to describe the content, grade of detail, and the underlying pedagogical concept of the media brick. Thus it is possible to integrate a media brick in a lesson, independent of the kind of media. A specific problem is the level of difficulty which is not a matter of a single media brick. Whether a media brick is difficult to understand depends on various aspects, such as background knowledge of a specific user or the media bricks the user has seen/ learnt before.

Particular emphasis of our work lies on the issue of coherence [16]. When a user of a hypertext does not have the possibility to choose the pages, i.e. he/she cannot establish the relation between the parts of the lesson by himself/herself, he/she is more likely to expect a coherent lesson, a lesson where the relations between the parts are obvious, although they are put together by someone else. For this purpose the media bricks are not only linked to the corresponding concept but also interconnected in the MediaBrickSpace by rhetorical relations based on the Rhetorical Structure Theory [10], [12].

Examples for such rhetorical relations are "deepen" or "explain". It is a major task of the system, to make these relations explicit. We currently use the following relations:

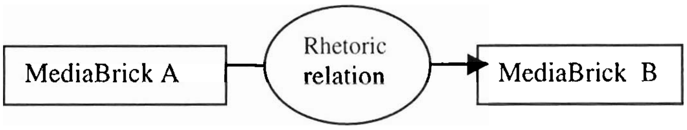

Fig. 4: Rhetorical relations between nodes in the MediaBrickSpace

example: MediaBrick B contains a text describing the runlength algorithm, MediaBrick A contains a concrete example with an uncompressed data stream and the runlength encoded stream.
illustrates: MediaBrick B contains a text with some numbers, MediaBrick A contains a diagram illustrating these numbers.
instance: MediaBrick B contains a graphic where all video compressing processes are listed. MediaBrick A describes MPEG.
restricts: MediaBrick B contains a description of a theory, MediaBrick A describes cases where the theory fails.
amplifies: MediaBrick B contains a video stream describing a CSCL system. MediaBrick A contains a text describing how CSCW systems work.
continues: MediaBrick B contains an uncompressed image. MediaBrick A shows the same image runlength encoded.
deepens: MediaBrick B contains an animation visualizing how the DCT works. MediaBrick A contains the formula of the DCT.
opposite: MediaBrick B states a theory of a specialist. MediaBrick A contains a statement of another specialist contradicting the first one.
alternative: MediaBrick B contains an image making clear what a pixel image is. MediaBrick A contains a text explaining a pixel image.

In general, the rules for building the lesson out of the relevant media bricks have to use the rhetorical relations and the characteristic of the media bricks, in order to match the users level of difficulty, media preferences and coherence expectations. Simultaneously, they have to work off the structure of the lesson compiled earlier and to fulfil the demands of the user's teaching method. Note that the teaching method requires rules working on both spaces. These goals are not always easily harmonized: The system will, for instance tend to select as the next media brick one that is connected to the current one but also one that is connected to the next topic in the planned structure.

COMPILING A LESSON

It is in the nature of learning that learners do not know in advance much about the domain they want to learn about, therefore they are not able to decide which aspects of the domain are relevant. It is hence the main goal of the Multibook system to offer help in form of tables of contents which are individually generated.
There are several reasons why it makes sense to provide explicit guiding for the user in the form of a table of contents:

- Users in a learning environment have to learn certain topics, if they want it or not.
- The users may not have the necessary overview of the area to choose the relevant concepts.
- The users are not able to compose a lesson according to a teaching method by themselves.
- The knowledge base is too broad to be managed effectively without help from the system. The system can select and combine the media bricks faster and more efficiently than the users.
- A table of contents offers the users some sort of orientation. They can check where they are and to which context the current topic belongs.

Like a teacher in a first step the system composes an outline of the lesson according to the needs of the user, who is known to it via the user profile. According to this outline a table of contents is created. In the following we describe in an exemplary way the rules that operate on the different categories of the profile and on the relations in the ConceptSpace to compile an adequate outline for a lesson. The rules which are responsible for choosing the global structure of the lesson use the semantic relations corresponding to the learning aim. Note that these rules are not a hardwired part of the system and can be changed easily as long as the ConceptSpace contains the according relations.

Examples for rules using the semantic relations: In the profile, the user is characterized as a student learning for exams. Such a user will probably expect from the system to teach him/her the definition, the broader term, the components and the application of the concept to be learnt. In this case the system will search for concepts which are related to the chosen topic by the relations "superconcept", "AEpart" and "invAEpart", the definition comes from the chosen concept itself. (The selection of the topics has to be coordinated with their order arising from the chosen teaching method, and if necessary completed with other concepts needed by a teaching method.)

On the other hand, managers are interested in different aspects of the same topic such as implemented systems and economical aspects. We suppose they also want to learn something about the broader term and applications. Hence their outline contains the concepts connected to the chosen topic by the relations such as "instance" and "costs" besides "superconcept" and "invAEpart".

The outlines are represented to the specific user in form of tables of contents dynamically generated for them.

Selecting the Suitable Set of Media Bricks

After having completed the outline for the lesson, the actual content is collected from the MediaBrickSpace. Here, other aspects of the user profile have to be considered such as media preferences and level of difficulty. The metadata of the media bricks provide information about the kind of material and the level of difficulty.

The level of difficulty is automatically calculated by the system. For this purpose we exploit the properties of the rhetorical relations. For example, when MediaBrick A explains MediaBrick B we assume, that A is easier than B. Therefore, A has a lower level of difficulty than B. In this way we compare the whole net of media bricks. Eventually we obtain for each media brick a value indicating the level of difficulty. It is very unlikely, that there will be always six different media bricks describing roughly the same content for six different levels. Therefore we use a hybrid method: we also combine media bricks to find the suitable level of difficulty. If there is, for example, no media brick at a basic level, the system can offer the user a more difficult media brick together with an illustration. On the other hand, if the media brick is too easy for an expert learner, but necessary, it can be combined with a brick connected to it by the relation "broadens".

To meet the media preferences of the users we added the meta-information what kind of media the specific media brick is, to the media brick. The rhetorical relation "alternative" connects media bricks with about the same content, but different media used. So the system can choose whenever available the desired presentation format. As mentioned above, there is always a text version, but we included as much multimedia elements as possible, such as videos and especially interactive simulations.

We have seen that concerning the dimension of the learning aim, we exploit the semantic relations of the ConceptSpace, concerning the dimensions of the level of difficulty and media preference, the system uses the meta-information attributing the media bricks. The teaching method is the most complex dimension in the user profile and the one for which it is most difficult to simulate a real teacher. It is not restricted to one of the two spaces but requires rules working on both the ConceptSpace and the

MediaBrickSpace. So far we have implemented rules for two quite straightforward teaching methods. It is our goal to pass the Multibook system to domain experts (teachers etc.) to realize their theories about teaching methods. Two implemented teaching methods, we implemented so far, are the hierarchical and the problem oriented approach. We now will describe some of the rules to generate a problem oriented document to illustrate how the system works.

To motivate the users with a problem which is addressed by the chosen topic they need to know more concepts than the ones above mentioned, namely the one which is connected to the topic or one of its broader terms by the relation "problemSolution" and the concepts which are on the way from these concept back to the chosen topic. Also the relations of the MediaBrickSpace are used: It is not sufficient just to mention a problem, to emphasize it, the system gives an example of it which can be found via the rhetorical relation "example".

CONCLUSION & OUTLOOK

In this paper we have described Multibook which intends to have individual lessons according to the user profile, the foundation of the Multibook, i.e. the knowledge base, and we have described some rules how the system works on the knowledge base to fulfil the objectives.

Currently the user profile is static, one of the most pressing tasks for the future is to make it dynamically adaptive. This means that the interaction of the learners can change their user profile during the learning session. If, for example, a user has stated to be a beginner, but always achieving excellent test results, the system will suggest to change the level and can apply this to the next document.

We have done a first evaluation giving the students the choice of the teaching method. From this experience we learnt that people find it hard to select a teaching method. Therefore the system must offer help for finding the most suitable one, probably in form of a questionnaire.

ACKNOWLEDGEMENT

The authors would like to thank the Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) which partially funded the research project and the Volkswagen Stiftung. We owe also thank to our colleague Dietrich Fischer, and especially to Lilo Kolb and Moni Jayme for their constant support.

REFERENCES

[1] Andersen J.R., Boyle C.F., Corbett A., and Lewis M.; Cognitive modeling and intelligent tutoring; Artificial Intelligence, 42

[2] Andre E., Rist T., and Muller J.; Life-like presentation agents: A new perspective for computer-based technical documentation. In Proceedings of the AI-ED workshop on pedagogical agents, Kobe, Japan 1997

[3] Beaumont I. H.; Interactive Anatomy Tutoring Systems ANATOM-TUTOR; in Brusilovsky P., Kobsa A., Vassileva J.; Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers, 1998

[4] Brusilovsky P., Methods and Techniques of Adaptive Hypermedia; in Brusilovsky P., Kobsa A., Vassileva J.; Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers. 1998

[5] Deek F.P. and McHugh J.A., A review and analysis of tools for learning programming; in Proceedings of the EdMedia, Freiburg, Germany, 1998

[6] Devedzic V., and Debenham J.; An Intelligent Tutoring System for teaching formal languages; In Proceedings of the Intelligent Tutoring Systems Conference, San Antonio, USA 1998

[7] Eklund, J., Brusilovski, P.; Individualising Interaction in Web-based Instructional Systems in Higher Education; in Proceedings of the Apple University Consortium's Academic Conference, Melbourne, 1998

[8] Fischer D.H., From Thesauri towards Ontologies?, in Hadi M., Maniez J. and Pollit St. A.; Structures and Relations in: Knowledge Organization, Proceedings of the 5th ISK Conference, 1998

[9] Greer J., McCalla G., Cooke J. et al., The Intelligent Helpdesk: Supporting peer-help in a university course; In Proceedings of the Intelligent Tutoring Systems Conference, San Antonio, USA 1998

[10] Mann W.C., and Thomson S.A., Rhetorical Structure Theory: A theory of text organization; Technical Report RS-87-190, Information Science Institute, USC ISI, USA 1987

[11] Milosavljevic, M., Oberlander, J.; Dynamic Hypertext Catalogues: Helping Users to Help Themselves; in Proceedings of the 9th ACM Conference on Hypertext and Hypermedia, Pittsburgh, 1998

[12] Nicholas N.; Parameters for Rhetorical Structure Theory Ontology; http://www.arts.unimelb.edu.au/Dept/LALX/publications/rst_ontology.html

[13] Rahmstorf G., A New Thesaurus Structure for Semantic Information Retrieval. In Finding New Values and Uses of Information, Proceedings of the 47th FID General Assembly, Tokyo, 1994

[14] Sowa J., Ontologies for Knowledge Sharing, Manuscript of the invited talk at TKE'96, 1996

[15] Steinmetz R., and Nahrstedt K., Multimedia: Computing Communications & Applications, 2nd edition, to be published, Prentice Hall, 1999

[16] Thüring M., Hannemann J., Haake J.; Hypermedia and Cognition: in Designing for Comprehension; Designing Hypermedia Applications, Communications of the ACM, 1995

[17] Dynamic Course Generation on the WWW; In Proceedings of the AI-ED'97 Workshop on Intelligent Educational Systems on the World Wide Web, 1997

[18] InterBook Home Page
http://www.contrib./andrew.cmu.edu/~plb/InterBook.html