[FiSt00] *Stephan Fischer, Ralf Steinmetz;* **Automatic Creation of Exercises in Adaptive Hypermedia Learning Systems;** Proceedings of 11th ACM Conference on Hypertext and Hypermedia (HT00). ACM, Juni 2000.

Automatic Creation of Exercises in Adaptive Hypermedia Learning Systems

Stephan Fischer
Industrial Process and System Communications
Dept. of Electrical Eng. & Information Technology
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany
Stephan.Fischer@kom.tu-darmstadt.de

Ralf Steinmetz
GMD IPSI
German National Research Center
for Information Technology
Dolivostr. 15 • D-64293 Darmstadt • Germany
Rst@kom.tu-darmstadt.de

ABSTRACT
In the last few years the automatic sequencing of course material became an important research issue, particularly with regard to the standardization of metadata for educational resources. Sequencing can help to generate hypermedia documents which match the learner's needs at its best. However, the generation of exercises is in most cases done manually. In this paper we propose an approach to generate exercises in an automatic way, exploiting the information which is already included in the knowledge base used in many adaptive hypermedia systems.

KEYWORDS: Adaptive Hypermedia Systems, Hypermedia Learning, Sequencing of Course Material, Knowledge Engineering.

INTRODUCTION

Course sequencing became an important research issue in the last years, especially with regard to standardization issues of learning metadata, such as Instructional Management Systems (IMS) [1] or the efforts of IEEE's Learning Technology Standards Committee (LTSC) [2]. It is the goal of a sequencing approach to generate a lesson for a targeted group, for example for students, which is tailored to the needs of that group. The contents of the respective lessons are built of hypermedia elements (texts, images, audio, video, animations) which have been stored in a modularized way.

These emerging standards will define a specification language and an environment for managing sessions in learning technology systems, e.g., computer-aided instruction, intelligent learning environments, intelligent tutoring systems.

The standards will (1) define the specification language, conceptual models, semantics, and syntax, (2) define the control transfer mechanisms and their encodings, e.g., how learning sessions are controlled and conducted, (3) define the data transfer mechanisms and their encodings, e.g., how student assessments and lesson plans are exchanged, (4) define an encoding method for storing and transferring session management "programs", i.e., interactive lesson plans.

Abstracting from the concrete goals of the standardization bodies an interesting observation can be stated: Much research is going on in the creation of lessons tailored to the specific user, particularly in adaptive hypermedia systems. On the other hand sophisticated test environments have been described which enable the learner to explore an unknown field with a great degree of freedom (see for example ELMART [3]). However, an integrated approach containing a lesson tailored to the user as well as interactive tests which can be explored in a self-defined way is still a challenging task. It should be mentioned that some programs do not intend to provide the old-fashioned style of learning documents where exercises follow an initial explanation of the topic to be explained. These approaches integrate exercises created manually with the necessary theory. The applications we are working with, however, use a hypermedia document to acquire necessary skills which can be tested afterwards.

In our approach we propose to offer exercises which have been created automatically. It can be observed in many educational books that exercises are a part of the book which offer the user the possibility to test his/her understanding of a complex topic. These exercises are in most cases not in a very interactive form. However, even the creation of these exercises is a very time-consuming task. It is hence our goal to derive very simple exercises from a knowledge base which can be found very often as part of an adaptive hypermedia system. Both the creation of exercises as well as the integration of hints to background material can be done automatically. It should be mentioned that it is not our goal to create an intelligent tutoring system as the exercises we derive do not adapt to the user by exploiting some kind of rule-based system. We hence see our approach as an initial step towards a powerful set of exercises which can be built both automatically and manually.

The remainder of the paper is structured as follows: In Section 2 we explain the architecture of Multibook, an adaptive hypermedia system where we integrated our exercise framework. In Section 3 we explain the algorithms necessary for creating exercises automatically. Section 4 summarizes related work. Section 5 concludes the paper and provides an outlook.

ADAPTIVE CREATION OF LESSONS IN MULTIBOOK

To be able to describe the automatic creation of exercises the underlying framework has to be explained. The content of the Multibook system (see http://www.multibook.de) currently being developed at the Technical University of Darmstadt, is the printed book "Multimedia: Computing, Communications & Applications" by Ralf Steinmetz and Klara Nahrstedt consisting of about 1200 pages, and a selection of Java applets [4]. The aim of Multibook is to have individual views on this material according to the needs and preferences of the individual users. These views are created on the fly using a sequencing approach [5].

A linear, printed book does not satisfy these requirements: It does not adapt flexibly to the level of difficulty, the learning aim, the learning strategy and the media preferences of the specific user. Apart from that, many aspects of multimedia technology can be explained in a better way using motion and interactivity.

A pure hypertext or hypermedia system does not satisfy these requirements either. A hypertext is static, in a sense that the text is either heavily linked and there is a suitable path through it for every user – then the user might be confused by the number of possibilities and not be able to find this path. There might also be only a few links, then the book would probably not satisfy the demands of each individual user.

An Intelligent Tutoring System (ITS) system does satisfy these requirements, but these systems only work in highly structured areas such as mathematics and especially programming languages. Many learning technology systems incorporate mechanisms for adapting lesson presentations according to the progress of the students. This adaptation scheme is the primary feature that produces "individualized" instruction – this is what is meant by "intelligent" in "Intelligent Tutoring Systems". It is for example the purpose of the standard developed by LTSC (see working groups P1484.6 and P1484.10 [1]) to provide a common mechanism for exchanging and developing this kind of information among the users, proctors, and developers of courseware (courseware is the educational content that is delivered to the students).
The level of control and the detailed knowledge of the state of the user can't be achieved in Multibook's subject domain. To formally model everything that can be said about multimedia in texts or images would not be a feasible approach. Also the extent and the extensibility of our material and the amount of text describing it are constraints which make it impossible to guide the user in such a controlled way.

The general functionality of Multibook, in other words the way the sequencing of lessons [6], is based on a knowledge base. This approach is similar to the standardization of IEEE's Learning Technology System Architecture (LTSA) as IEEE proposes the use of a knowledge library (knowledge base) which is responsible for the sequencing of a lesson, while the actual compilation of the lesson is performed by a delivery component (see Figure 1).

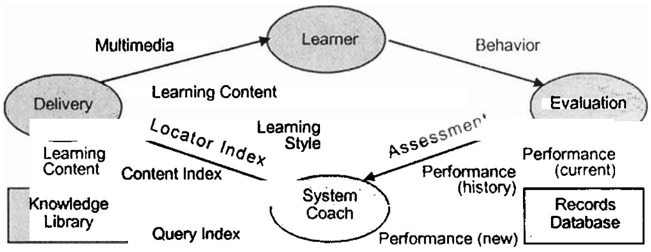

Figure 1: Architecture of IEEE's LTSA

It is essential to understand the setup of our knowledge base in order to understand the automatic creation of exercises. The knowledge base being used by Multibook works with two different spaces: The so called ConceptSpace and the MediaBrickSpace. Considering the way an author writes a document the following order can be specified: (1) an author acquires background knowledge, (2) an author creates an outline for a document, (3) an author fills the outline with content. These steps are modeled by different spaces in Multibook. The ConceptSpace contains an ontology in terms of keywords which is necessary to create the outline of a lesson. After the sequencing of the outline (equally to the creation of a table of contents) the "real" content (text, images, audio, video, animation) is filled into the outline using elements of the second space, the MediaBrickSpace. A general idea of Multibook is that it is necessary to employ different relations within the ConceptSpace and the MediaBrickSpace to model the different goals which both spaces have. In order to fulfill the requirements of both spaces we use relations with regard to knowledge management in the ConceptSpace and rhethorical relations in the MediaBrickSpace. We describe the necessary relations in the next few paragraphs.

A great advantage of the separation of concepts from content is that content in the system can be changed without affecting the overall structure stored in the ConceptSpace. It is also very comfortable to extend parts of a document, for example by deepening explanations or by examples, if additional media bricks are inserted. The process of inserting additional media bricks does not change the ConceptSpace.

Name of relation	Explanation
Superconcept	A node is a superconcept of another node.
AEPart	For all instances of a node there exists a subnode
EEpartOf	There exists a subnode for an instance of a super-node
(Inverse)Procedure	A node contains a (an

	inverse) procedure with regard to another node.
Follows/Precedes	A node follows/precedes another node (ordering in a document)
Formalize/IsFormalizedBy	A node formalizes / is formalized by another node
ProblemSolution	A node points to a problem-node which is connected to a solution-node.
Partition	Subnodes partition a domain, for example images are partitioned as b/ w, gray, and color.
Cost	The cost of another node
Uses	A node uses another node
Application	A node is an application of another node.
Instance	A node is an instance of another node

Table 1: Relations of the ConceptSpace of Multibook

In the ConceptSpace we use the set of semantic relations indicated in Table 1 (examples are provided in Section 3). Some of these relations are well known from approaches to knowledge management. Semantic relations which can be stored as metadata [8][17] can help to structure a complex domain. It should be mentioned that the set of relations we apply is in no way complete. Multibook uses technical information about multimedia. The semantic relations above are sufficient to model our specific domain. The use of the model for another domain would however imply a re-design of these relations. An example would be a medical learning system where a relation "causes" would have to be introduced describing diseases a node would cause.

The sequencing of an educational document according to a specific learning strategy is then a specific navigation on the ConceptSpace. The implementation of a learning strategy has to be done by a pedagogical expert [6] together with an expert in computer science in order to match the navigation on the ConceptSpace with a learning strategy.

Once the table of contents has been set up it has to be filled with content. Multibook models the content in the so called MediaBrickSpace. The difference to the ConceptSpace is the fact that we use a second set of relations in the MediaBrickSpace, rhetorical-didactic relations known from the area of natural language processing [7]. These relations (see Table 2) model dependencies between media bricks which have been stored in a modularized form. Exploiting the type of the relation, the preferences of the learner can be fulfilled. Using the type of media of a media brick, media preferences can be taken into consideration, for example, a user on a slow modem link does only get a text version of an educational document. This type of relations can also be used to switch between different levels of difficulties. As it is a difficult task to measure the kind of difficulty of a separated content module, Multibook uses the fact that the relative difficulty of aggregated bricks can be measured. It is quite obvious that a text A together with two transparent examples is easier to understand when compared to a version containing only the text A. Also a text A will become more difficult when presented together with a second text B which is connected to A with the rhetorical relation deepens. Another important advantage of the Multibook approach is the fact that the coherence of a lesson can be restored when using the type of a rhetorical relation. It should not be ignored that coherence is a great advantage of a sequential educational document. On the other hand, when created from content modules, a text will be very difficult to be read. However, the type of a rhetorical relation is well suited to restore the coherence. An example would be a text A together with an example B. As A and B are connected by the relation example, it is quite easy to include a little text-sample like "In the following an example will be given which illustrates the problem."

Name of relation	Explanation
example	A node is an example for another node.
illustrates	A node illustrates another node.
restricts	A node restricts the applicability of another node
amplifies	A node amplifies the use of another node.
continues	The explanation started in one node is continued in another node.
deepens	A node deepens another node.
opposite	A node is the opposite of another node.
alternative	A node is an alternative of another node.
analogy	A node is an analogy to another node.

Table 2: Relations of the MediaBrickSpace of Multibook

Multibook contains many other important components, for example a user profile which tracks the actions of a learner. These interesting topics are beyond the scope of this paper.

AUTOMATIC DERIVATION OF EXERCISES IN MULTIBOOK

As we understand our paper as a first step towards the automatic derivation and sequencing of exercises and as we are also well aware of the fact that the automatic generation of complex exercises is questionnable from the pedagogical viewpoint, we try to address the automatic creation of simple exercises. Many textbooks include a summary of a chapter followed by some simple exercises. It is the goal of this paper to propose a way by which exercises of that type can be created automatically.

The classic exercise in a technical book is to have the student extend the material already discussed – to prove a lemma or minor theorem, to generalize a synthetic approach, to apply an algorithm in an unexpected context. It should be mentioned that our approach – and maybe such approaches in general – will not be able to create such exercises with a generally sound pedagogical strategy behind the algorithms. We hence understand our work as an assistance for the teacher when enriching the lessons by simple questions, not as a replacement of the educator.

A selection of the types of simple exercises which can be found in technical computer science books is listed below. It should be mentioned that these types are the ones which are most interesting when teaching a technical domain, such as multimedia systems.

- *Part-of-questions.*
  Example: What are the parts of an adaptive hypermedia system?
- *Application-of-questions.*
  Example: What are the application areas for Intelligent Tutoring Systems?
- *Questions about details*
  Example: How many rows and columns constitute a QCIF-image?
- *Calculations*
  Example: Calculation of a packet length in Ethernet.

The location in the system where a derivation of exercises could take place can only be the ConceptSpace. As the MediaBrickSpace merely contains rhetorical-didactical questions, there is no information about dependencies of content available in the latter space.

The goal to derive exercises automatically from an ontology requires the analysis of the different types of questions.

Questions about details – as well as the classic complex questions mentioned above – cannot be derived from the ConceptSpace as details are never part of an outline. Also calculations cannot be derived automatically, as the necessary terminology of the ConceptSpace does not contain the necessary mechanisms. It is also doubtful if the support of the learner could be created automatically as complex questions can lead to a great variety of different (correct and wrong) solutions. The (manual) creation of exercises and the automatic support of a learner is indeed one of the areas of Intelligent Tutoring Systems.

Part-of questions as well as application-of questions can be created automatically in a multiple-choice-style which will be shown in the remainder of this section of the paper. To be able to explain the necessary algorithms it is necessary to explain a part of our ontology as an example (see Figure 3).

The part of the ontology illustrates the model we use to create lessons about the image compression algorithms in JPEG. It should be repeated that the model is only a part of the whole ontology we use. Some relations have been cut out for the sake of clarity. The model contains the concept JPEG in the center. The different parts of JPEG, the image preparation, the discrete cosine transform, the quantization and the entropy encoding are connected by uses-Relations. Also applications of JPEG are visible, for example the concepts "WWW" or "MPEG". These concepts are connected using the relation application. It should be mentioned that in this context we are not interested in MPEG or in the WWWitself, but in MPEG and the WWW as a possible application of JPEG.

To be able to create part-of-questions, we use the following algorithm:

1. Create the text of a question: "Which are the parts of <Name of the concept>. Please select the correct answers."
2. Build a list of the correct answers by following the "uses" relations, for example of the concept JPEG: Image preparation, Discrete Cosine Transformation, quantization and entropy encoding.
3. Build a list of wrong answers (explained below).
4. Merge both lists in a random order.
5. Check, if correct answers are provided by the learner.
6. If wrong answers are clicked in the multiple choice form then offer the option to branch to explanations of wrong concepts as well as to a repetition of the lesson being learned.

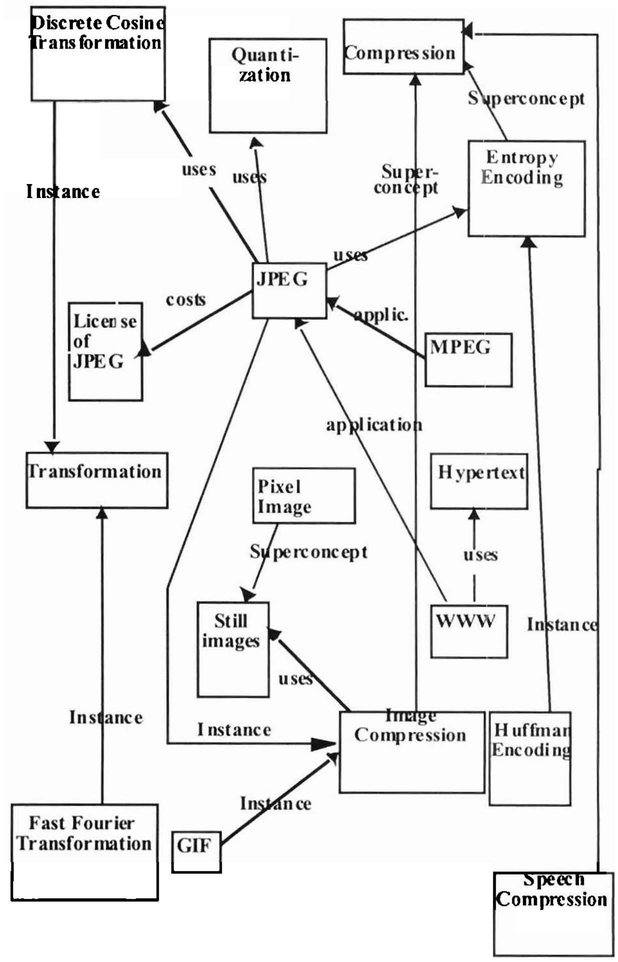

Figure 3: Part of Multibook's ontology

To be able to match the background knowlegde of the learner we suppose that a selection process with regard to a user profile has already taken place. Multibook contains a user profile which can not be described in the scope of this paper.

Obviously an important step of the algorithm is the selection of appropriate wrong answers as these answers cannot be picked in a random-style out of the ontology. However, the following assumptions can help to pick a set of wrong answers:

- Most ontologies contain one or more taxonomies which are tree-like hierarchical structures. An example would be the taxonomy Compression => Image Compression => JPEG.
- Wrong answers should originate from a similar context, for the JPEG-example a wrong answer like "compression of motion pictures" would be an appropriate choice, a wrong answer like "ATM-cell" would be an inappropriate answer.

We apply the following algorithm to select wrong answers in a semantically meaningful way:

1. Reduce the ontology to a taxonomy using the relation "superconcept". Pass "instance"-relations if necessary.
2. Leave the current branch of the hierarchical tree of the taxonomy up to the next superconcept. Pass "instance"-relations if necessary. Check for other subconcepts.
3. Jump to new subconcept and select new wrong answers for the question to be created, evaluating the "uses"-relations.
4. Repeat procedure with next hierarchy level if necessary

An example for such a selection is illustrated in Figure 4. The algorithm first selects the correct answers by evaluating the "uses" relations connected to the concept JPEG. In the next step it passes the "instance" relation to the concept "image compression" and from there to the concept "compression" using the relation "superconcept". Using the path from "video compression" to "MPEG" new components can be found which are wrong answers with a semantic relationship to the concepts connected to "JPEG".

It is very important to identify wrong answers which are in a close semantic relationship to the correct answers. Several observations can be found which restrict the search:

- An ontology can contain several taxonomies. An example for multimedia systems would be that there are two distinct taxonomies for compression and for networking. Although there are semantic relations between compression and networking (compression makes networking more efficient by a save of bandwidth) as part of the overall ontology, there are no such relations between the two taxonomies of networking and compression. An important reason is that a taxonomy is merely based on the relations "superconcept" and "instance" in our model. It can never happen that networking is a superconcept of compression in general or vice versa. This observation guarantees in very coarse bounds that it cannot occur that totally different concepts become part of the same question.
- It can happen that no branches of the taxonomy can be found where concepts are located which are semantically similar to the correct answers. On the other hand it is quite useless to create a question where all the answers are correct. Experiments we conducted with students showed that it is better to avoid an automatic generation of exercises in that case.

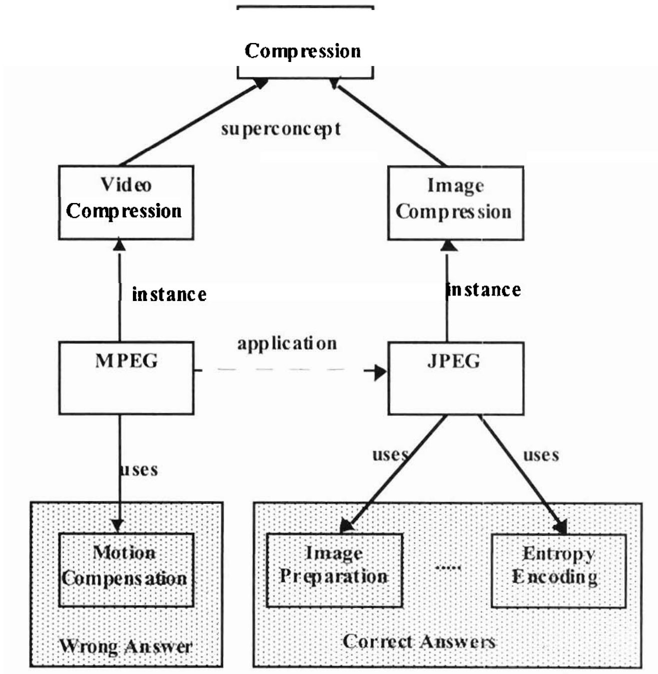

Figure 4: Identification of correct and wrong answers for a question (part-of exercise).

Having explained the way we generate "part-of"-exercises we continue our description of a second type of questions, the "is an application for"-questions.

To generate exercises which verify that a student has understood the general applicability of a concept we use a similar approach:

1. Create the text of a question: "What are the applications of <Name of the concept>. Please select the correct answers."
2. Build a list of the correct answers by following the "application" relations, for example of the concept "JPEG": MPEG is an application for JPEG.
3. Build a list of wrong answers (explained below).
4. Merge both lists in a random order.
5. Check, if correct answers are provided by the learner.
6. If wrong answers are clicked in the multiple choice form then offer the option to branch to explanations of wrong concepts as well as to a repetition of the lesson being learned.

With regard to Figure 3, the algorithm would select the correct answers MPEG and WWW where JPEG is applied. To pick the wrong answers a similar approach compared to the selection of wrong answers for "part-of" exercises can be used. However, an important observation lies in the fact that applications of concepts are in most cases instances. JPEG is for example an instance of the abstract concept "image compression". The selection of wrong answers is hence less critical compared to the selection of wrong answers for "part-of"-exercises. Although it would not make much sense to ask if, for example, Ethernet is an application for JPEG, the learner would not be surprised in a way he would be disturbed, if concepts from another taxonomy would be used in the "part-of"-questions. The algorithm to select wrong answers is as follows:

1. Reduce the ontology to a taxonomy using the relation "superconcept".
2. Leave the current branch of the hierarchical tree of the taxonomy up to the next superconcept. Pass "instance"-relations if necessary. Check for other subconcepts.
3. Jump to new subconcept and select new wrong answers for the question to be created, evaluating the "application"-relations.
4. Repeat procedure with next hierarchy level if necessary (if no appropriate concepts can be found in one level).

The algorithm is explained in a graphical way in Figure 5.

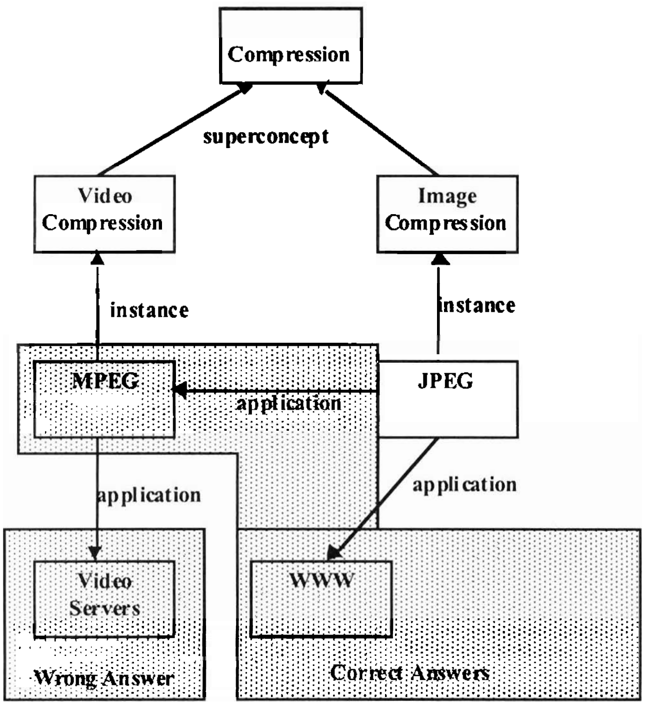

Figure 5: Identification of correct and wrong answers for a question (application-exercise).

EXPERIENCES

In a first evaluation our students worked with the exercise environment. The experimental setup contained both questions generated by hand and questions generated automatically. Part of the evaluation was the question if the student could identify the exercises created automatically. It turned out that in most cases the students were not able to distinguish between both types of exercises. A clearer identification of exercises generated automatically became possible when there was a great local displacement of the concepts chosen as correct and wrong answers in the taxonomy. In other words, each time more than one superconcept-relation was passed in order to find wrong answers the semantic similarity of correct and wrong answers became smaller and the students were able to identify the difference. However, most students stated that they might not have noticed the difference if we would not have told them in advance. However, it should be repeated that this assumption is only true if the tree of a specific taxonomy is not left and wrong answers don't result from a different taxonomy.

RELATED WORK

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 [14],[13]. 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 [12] 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 general setup in Multibook has acquired a great amount of experiences gained in adaptive hypermedia systems, such as Interbook [9] [13] or ELM-ART [3] [16]. More details about related work in adaptive hypermedia documents with regard to the realization and implementation in Multibook can be found in [5].

Multibook is based on an terminological ontology. Related work on ontologies has been described in [11] [16], the use of ontologies with regard to the rhetorical structure theory has been described in [7] and [10].

The realization and the implementation of Multibook has to be seen in a close context to the work of the Learning Technology Standards Committee (LTSC) which also proposes the use of a knowledge base and of a delivery component, matching in some sense the use of a ConceptSpace and a MediaBrickSpace [1].

CONCLUSIONS AND OUTLOOK

In the paper we described a way to create exercises automatically. The creation of exercises is based on a knowledge base (ontology) containing the curriculum of multimedia technology taught in Darmstadt. We use the fact that many ontologies also contain taxonomies allowing for the creation of different types of questions.

The implementation of a knowledge base uses two spaces within the Multibook approach: The ConceptSpace, and the MediaBrickSpace. In the paper we have shown that an evaluation of the properties of the ConceptSpace is sufficient to generate questions automatically.

It could be argued that our approach only works on our specific realization of the ontology. In fact, the contrary is true. The generation of exercises may vary from ontology to ontology. However, the algorithms we proposed rely on the fact that an ontology contains a set of disjunct taxonomies which is independent from our implementation of the ontology.

The type of multiple choice exercises created automatically is neither interactive nor does it allow a kind of self-explorative learning. Also the approach might be restricted to technical domains such as computer science. It should again be stressed that it is not our goal to create a sophisticated test environment. The way which has been described in the paper merely matches the state of the art of many educational resources and can help to reduce the amount of work to produce exercises by hand.

For the future we will have to integrate the knowledge of the learner into our exercise environment. Questions which have to be answered include the intelligent provision of hints which help a student in the case of wrong answers. These hints can also be derived automatically by exploiting the structure of the ontology. An important aspect will be the use of a user profile which records the order in which a learner traverses the educational document. Currently we did not integrate the knowledge of the parts of a document a student has already seen. However, this can lead to the fact that questions are generated which base on a knowledge which has not yet been presented to the learner.
We understand our approach as a first encouraging step into the area of exercises created automatically. It has been the goal of the paper to describe the overall ideas. We are well aware that other types of questions could be generated automatically without the idea of using an Intelligent Tutoring System. It is our goal to think about other types of questions which can be derived automatically in the near future.

REFERENCES

1. Learning Technology Standards Committee (LTSC), http://ltsc.ieee.org.
2. Instructional Management Systems (IMS), http://www.imsproject.org
3. M. Specht and G. Weber. Episodic adaptation in learning environments. In Proceedings of the European Conference on Artificial Intelligence in Education, Lisbon, Portugal, September 1996.
4. R. Steinmetz and K. Nahrstedt, Multimedia: Computing Communications & Applications 2nd edition, to be published, Prentice Hall, 1999.
5. C. Seeberg, A. Steinacker, K. Reichenberger, S. Fischer, and R. Steinmetz. Individual Tables of Contents in Web-based Learning Systems. In Proceedings of the 10th ACM Conference on Hypertext and Hypermedia, February 1999.
6. A. Steinacker, C. Seeberg, K. Reichenberger, S. Fischer, and R. Steinmetz. Dynamically Generated Tables of Contents as Guided Tours inAdaptive Hypermedia Systems. In Proceedings of the EdMedia & EdTelecom, June 1999.
7. W.C. Mann and S.A. Thomson: Rhetorical Structure Theory: A theory of text organization; Technical Report RS-87-190, Information Science Institute, USC ISI, USA 1987
8. A. Steinacker, C. Seeberg, S. Fischer, and R. Steinmetz. Multibook: Metadata for the Web. In 2nd International Conference on New Learning Technologies, August 99.
9. InterBook Home Page http://www.contrib.andrew.cmu.edu/~plb/InterBook.html
10. N. Nicholas: Parameters for Rhetorical Structure Theory Ontology; http://www.arts.unimelb.edu.au/Dept/LALX/publications/rst_ontology.html
11. J. Sowa: Ontologies for Knowledge Sharing, Manuscript of the invited talk at TKE'96, 1996
12. I. H. Beaumont; Interactive Anatomy Tutoring Systems ANATOM-TUTOR; in Brusilovsky P., Kobsa A., Vassileva J.; Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers, 1998
13. P. Brusilovsky, Methods and Techniques of Adaptive Hypermedia; in Brusilovsky P., Kobsa A., Vassileva J.; Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers, 1998
14. J. Greer, G. McCalla, J. Cooke et al; The Intelligent Helpdesk: Supporting peer-help in a university course; In Proceedings of the Intelligent Tutoring Systems Conference, San Antonio, USA 1998
15. D.H. Fischer; From Thesauri towards Ontologies?. In: M. Hadi; J. Maniez, and S. A. Pollit; Structures and Relations. In: Knowledge Organization, Proceedings of the 5th ISK Conference 1998.
16. G. Weber and M. Specht; User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems. In: Proceedings of the 6th International Conference on User Modeling, Cagliri, Italy 1997.
17. IEEE Learning Technology Standards Committee (LTSC) IEEE P1484.12 Learning Objects Metadata Working Group http://ltsc.ieee.org/wg12