September 2003

Living Books

Peter Baumgartner
Universität Koblenz-Landau

Ulrich Furbach
Universität Koblenz-Landau

Margret Groß-Hardt
Universität Koblenz-Landau, margret@uni-koblenz.de

In: Uhr, Wolfgang, Esswein, Werner & Schoop, Eric (Hg.) 2003. *Wirtschaftsinformatik 2003: Medien - Märkte - Mobilität*, 2 Bde. Heidelberg: Physica-Verlag

ISBN: 3-7908-0111-9 (Band 1)
ISBN: 3-7908-0116-X (Band 2)

Living Books

Peter Baumgartner, Ulrich Furbach, Margret Groß-Hardt
Universität Koblenz-Landau

Abstract: In this paper we introduce a personalized way of presenting learning material, which is in particular suited for the use in e-Learning environments. We will explain the knowledge-based aspects of this technology, and we demonstrate with an example the additional feature of interactivity, by which the approach deserves the name "Living Book". The various possibilities of using Living Books in university teaching are discussed.

Keywords: E-Learning, Artificial Intelligence

1 Introduction

E-Learning offers a chance to explore new forms of learning and to provide learning infrastructure for a broad spectrum of applications.

To this end various different e-learning platforms are in use for the distribution of learning material, the organisation of courses and the communication between learners and teachers. In this paper we concentrate on one main aspect, namely on the learning material. We elaborate the Living Books concept, which is currently developed in various projects at the University Koblenz[1].

One main characteristic of Living Books is, that it is not only the presentation of the book which uses new technologies, like web browsers or a server-client architecture, moreover, it is also the way the material is stored and retrieved. The material is devided into a large number of small units or slices that are stored together with meta-data on a web server. These slices can be combined by various intelligent techniques based on a user query, taking into account a user model. The result of such a query to the book is a document, which is personalized and fulfills given characteristics of predefined learning scenarios.

[1] This work is supported by
- EU IST-Programme www.TRIAL-SOLUTION.de
- Bmbf Neue Medien in der Bildung www.in2math.de
- DFG Focus Programme www.wissenskommunikation.de

Another characteristic of this concept is the possibility of interaction: the user can run programs that demonstrate topics from the book, by invocation directly from within the book, using examples from the book or by giving own input data. The result of these interactions is contained within the delivered document, and is kept as a personalized version for each user on the server.

In the following two sections we will describe these features, and thereafter we will report on our experiences of using our system in university teaching.

2 Knowledge Based Books

The system uses a knowledge management system (KMS) to store and to retrieve the parts of the book.

The user gives a query or specification of the view to the book she wants to get delivered and the system then composes a personalized version of the book according to a user model, which is available to the KMS. This is depicted in Figure 1 and in the following we will briefly explain this process.

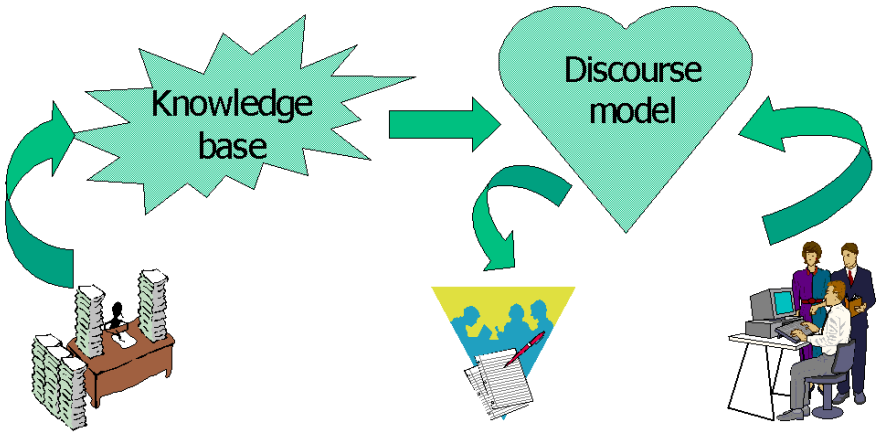

Figure 1: Interaction between user and KMS.

It is the *Slicing Information Technology (SIT)* [Dahn01] for the management of personalized documents. Its kernel has been developed within the ILF-system in the German focus program *Deduction*, which was carried out in the nineties. The Slicing Book technology handles documents or textbooks that are split into small semantic units, so called *slices* or *units*, which may be e.g. a paragraph, a

definition, or a problem in the original documents. Additional meta data play an important role to describe e.g. dependencies among slices, possible between slices from different documents. Also, keywords can be assigned to slices to indicate what the contents of a slice is about. The process of slicing is partially automated, but usually needs some further manual work.

The Living Book is embedded in a software called the *SIT-Reader*, which allows HTML-based access to slices. The slices together with the meta data are available trough a server-client architecture. The client has access through a interface, which offers all functionality through a common internet browser; see the screenshot in Figure 2.

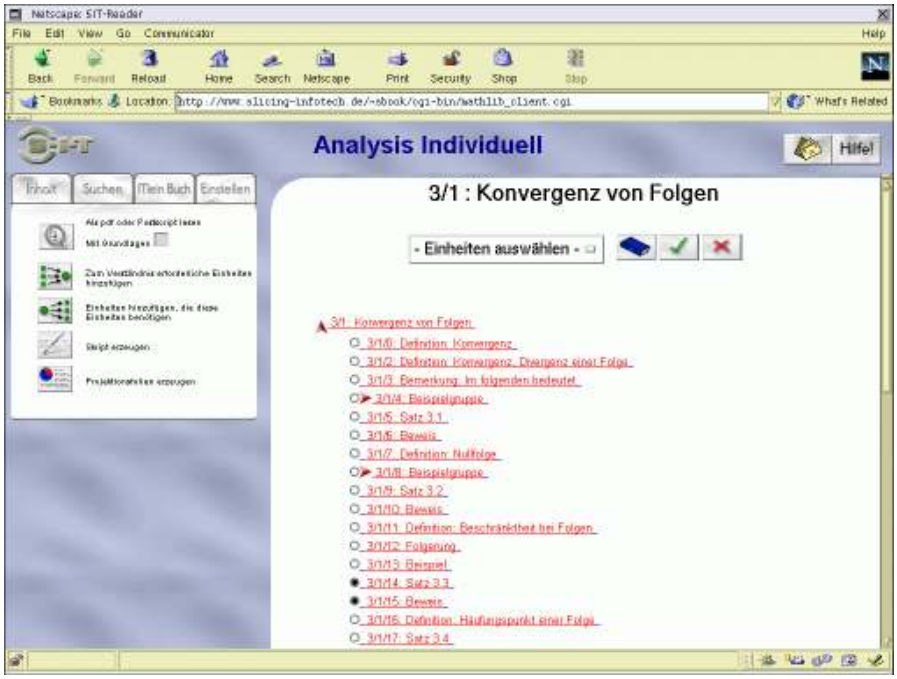

Figure 2: The Browser-Interface of the Reader

To use the system, a user can mark units, like e.g. analysis/3/1/15 and analysis/3/1/16 representing e.g. theorem 3.1.15 in the analysis book together with its proof. Then she can tell the system that she wants to read the marked units and gets a PDF document containing just the units she wants to view. If the user thinks that this information is not sufficient for her understanding, she can tell the system to include all units which are prerequisites of the units selected.

However, the Living Book goes beyond what is possible with the SIT-Reader alone. For instance, a user may select a certain chapter, say e.g. chapter 3

containing everything about integrals in the analysis book. But instead of requesting all units from this chapter the user wants the system to take into account that she knows e.g. unit 3.1 already, and she possibly wants just the material that is important to prepare for an exam. Based on the units with their meta data the deduction system can exploit this knowledge and combine the LaTeX based units to a new document (hopefully) fitting the needs of the user. In conclusion, we not only have the text of the books, we have an entire knowledge base about the material, which can be used by the reader in order to generate personalized documents from the given books.

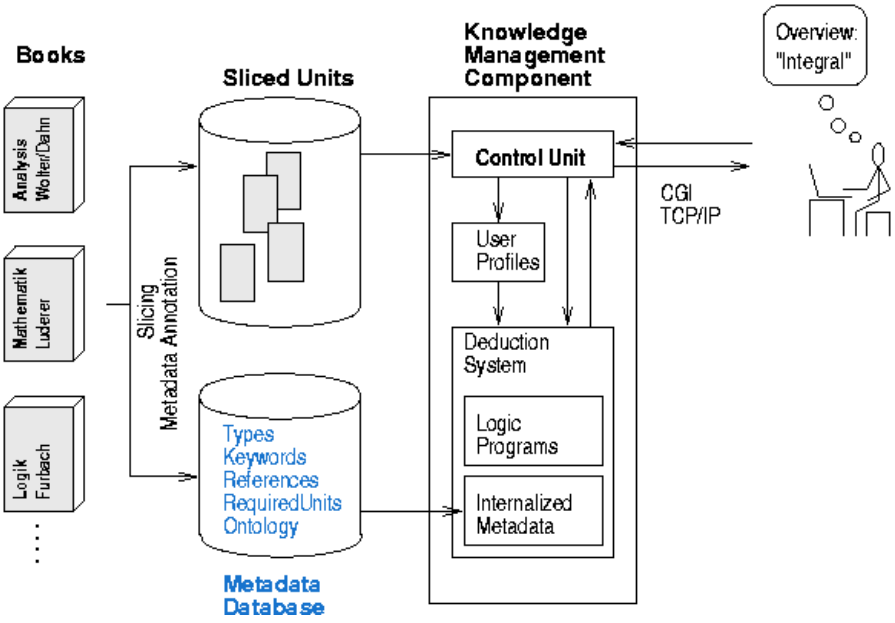

Figure 3: Knowledge Management System architecture

From a technical viewpoint, the most interesting component of Living Book is the Knowledge Management System (KMS), in particular the Knowledge Management Component with its deduction system KRHyper. Figure 3 depicts the architecture of the KMS. As mentioned there, KMS handles meta data of various types, which are *Types* of units ("Definition", "Theorem", etc), *Keywords* describing what the units are about ("Integral", etc), *References* between units (e.g. a "Theorem" unit about "Integral" refers to a "Definition" unit), and what units are *Required* by other units in order to make sense.

Having depicted the contents of the knowledge base, it should be clear that, compared with conventional books, there is a lot more information to be provided by the author. She is resonsible for the contents (of course), for the layout (as she is used, since the introduction of desk top publishing systems) and additionally she

has to provide the meta data, which then can be accessed by the KMS. Some of the meta-data can be extrated automatically out of the Latex-source (Word is supported, too), by analysing the typesetting structure of the units. However, most of the information have to be included by means of an authoring tool, which enables the author to define meta-data and and to do simple text-editing tasks.

Furthermore, a *User Profile* stores what is known and what is unknown to the user. It may heavily influence the computation of the assembly of the final document. The user profile is built from *explicit* declarations given by the user about units and/or topics that are known/unknown to him. This information is completed by deduction to figure out what other units must also be known/unknown according to the initial profile. More details about the KMS and the deduction system used within the KMS can be found in [Bau+03].

3 Living Books

One of the goals we are aiming at, is the support of explorative learning. This means, students have control about accessing the learning material from different angles and different levels of detail. They also have options how to solve certain problems. A problem can be solved manually or by means of some system; for instance, a truth table generator can help in deciding if a certain logical formula is satisfiable.

From within the documents, interactive elements are accessible that allow the user to make use of different mathematical or logical systems. Furthermore, formula generators provide a repository for training. The lecturer and the student might use these systems in different interactive settings and related to different teaching and learning goals. All systems are integrated seamlessly in the book; neither student nor lecturer do have to spend time in learning different interfaces of the underlying systems which might be quite time consuming and sometimes frustrating if the use of the system is complex. Instead, the Living Book offers an easy to use web based interface.

In order to achieve high quality layout of the electronic material, and also allow a user to print some parts of the material, LaTeX is used for typesetting with PDF as the target format, which in turn is used for displaying the material in a web based environment. LaTeX in combination with PDF on the one hand side guarantees excellent display of scientific content, as e.g. mathematical formulae, and on the other side allows to print (part of) the documents with high printing quality.

Currently, the Living Book “Logic for Computer Science” (author: Prof. U. Furbach, Universität Koblenz-Landau) is developed and has been used during lectures and tutorials, already. Due to the fact that this kind of electronic material

is quite new and experience still has to be collected, regular evaluations are taking place [SiVa02].

The following is a list of representative interactive tools currently integrated in the Living Book. Since the book is dealing with logics these tools all are of course from the context formulae manipulation and theorem proving.

- *Formulae Database*: A set of "current" formulas is maintained on a user specific basis by the system. There are operations to add and delete formulas, and also to store and retrieve the current set under a chosen name. Another means to obtain formulae is the following.
- *Formulae Generator*: This tool generates propositional formulae that can be used in various exercises. Thereby, the Living Book memorizes which formulae already have been generated for a certain user. These formulae will not be presented a second time.
- *Truth Table Generator*: This system takes a propositional formula as input and generates a truth table. It is used in the first part of the logic course where the students learn about propositional formulae, satisifiabitlity and model generation.
- *Tableau prover*: This system is used to test satisfiability of propositional formulas. The salient feature of this system is a graphical, tree based visualization of the computed tableau. The visualization yields an illustrative and explanatory description of the steps executed in finding a proof (or a counterexample). A possible application scenario is to quiz students to formalize in propositional logic, say, a logical puzzle, and use the Tableaux prover to "debug" their solution.
- *CNF Transformation*: This system takes a propositional formula as input and generates its clause normal form (CNF). It can be used by the students to understand clauses and the semantic equivalence between propositional formulae and CNF representation. Also, the CNF representation is the input format accepted by most automated deduction systems, and the CNF obtained by this tool can be fed directly into the deduction systems in the Living Book.
- *Resolution systems*: The resolution calculus is the most prominent method for reasoning with CNF formulas. There are two respective systems included: a naïve one, and the state of the art theorem prover OTTER. The naïve one closely corresponds to the resolution calculus as introduced in the logic course. Since it is not optimized, it can be used for small examples only, and it thus demonstrates the need for optimized techniques (which are present in OTTER).
- *PROTEIN prover*: The PROTEIN model elimination prover takes CNF as input, and it returns a tableau similar as mentioned above; it is an alternative to

using OTTER, and it can be used to demonstrate the need for more than one system.

Basically, the Living Book and all the systems embedded are used not only by students but also by instructors to explain different topics. The mentioned interactions are also available as first-order predicate logic, which is another core topic in the logic course.

3.1 Interactivity - by Example

This section describes the use of the interactive components in the Living Book by means of an example. The domain is the logic-based diagnosis of electrical circuits. The theory behind logic-based diagnosis is sufficiently explained in the course. The purpose of the use of an interactive system here is to train students in modeling an application domain and let them use an automated theorem prover to solve some task from the application domain (i.e. finding diagnosis).

When treated within a (classroom) exercise, the modelling of the application would presumably be prepared in a stepwise way, thus gradually introducing more complexity and more involved questions to be solved with the interactive component. There is no need to describe these steps here in detail, and we are thus concentrating on the final step.

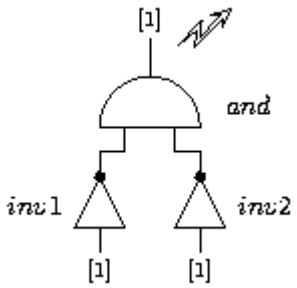

Figure 4: An electronic circuit to be diagnosed.

To mediate the principles of logic-based diagnosis, a small example is sufficient. The underlying circuit is depicted in Figure 4. Notice that the observed output ("[1]") contradicts the output expected from the given input vector ("[1] - [1]"). The students are expected to describe this circuit by formulas of propositional logic. Of particular importance is to take abnormal behavior into account.

A correct formalization is depicted in the upper half of the screenshot in Figure 5, and (parts of) the user input leading to it is depicted in the lower half.

Now, various interactive components could be used to solve the task at hand, which is to determine possibly faulty components of the circuit that are consistent with the observed behavior. The most natural interactive component to use here, however, is the *Tableau prover*. After invocation, a tableau for the given formulas is constructed. Tableaux are tree data structures, and it is easy to read off certain logical properties from them. In particular, from inspecting the branches the diagnosis task can be solved. In order to enable students to do so, tableau are typeset by some LaTeX tree drawing package. The result is delivered in PDF and can be inspected as usual. Figure 6 shows an overview of the tableau, and Figure 7 shows a zoomed-in view.

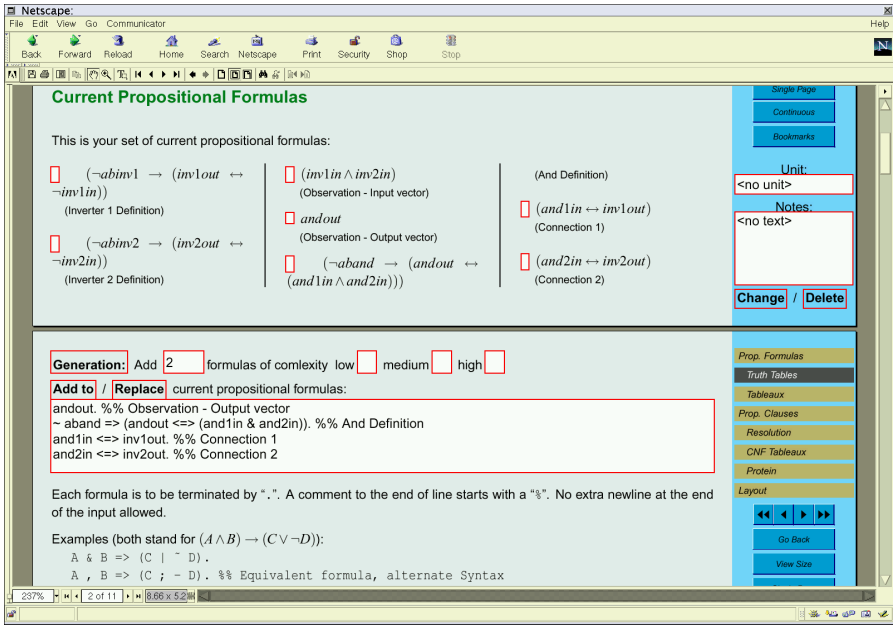

Figure 5: The formula editor, showing a formalization of the circuit in Figure 4.

The kind of interaction as just outlined is currently explored within the logic course held in the summer term 2002. First experiences are very encouraging, and a more formal evaluation will be carried out at the end of the course.

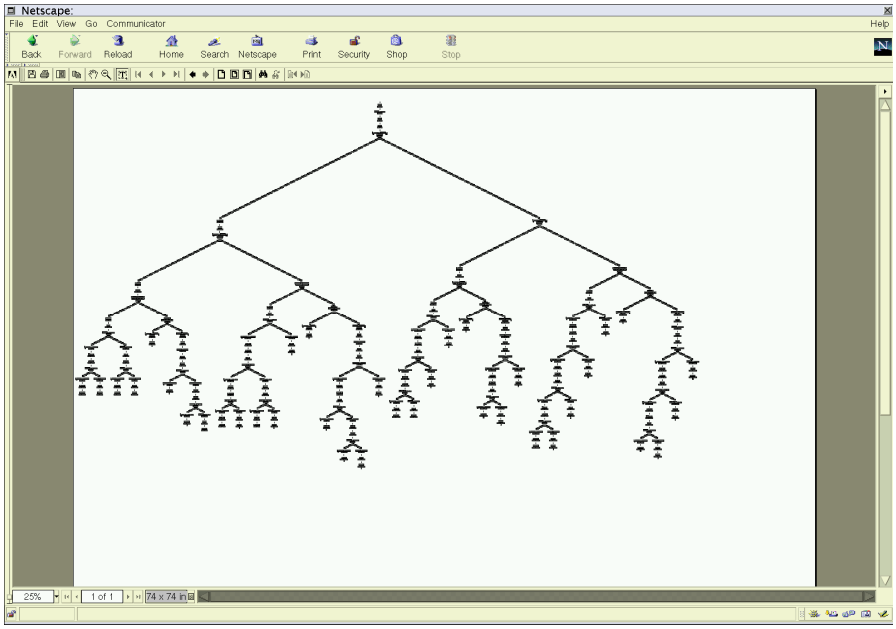

Figure 6: The tableau resulting from the formulas in Figure 5 (overview).

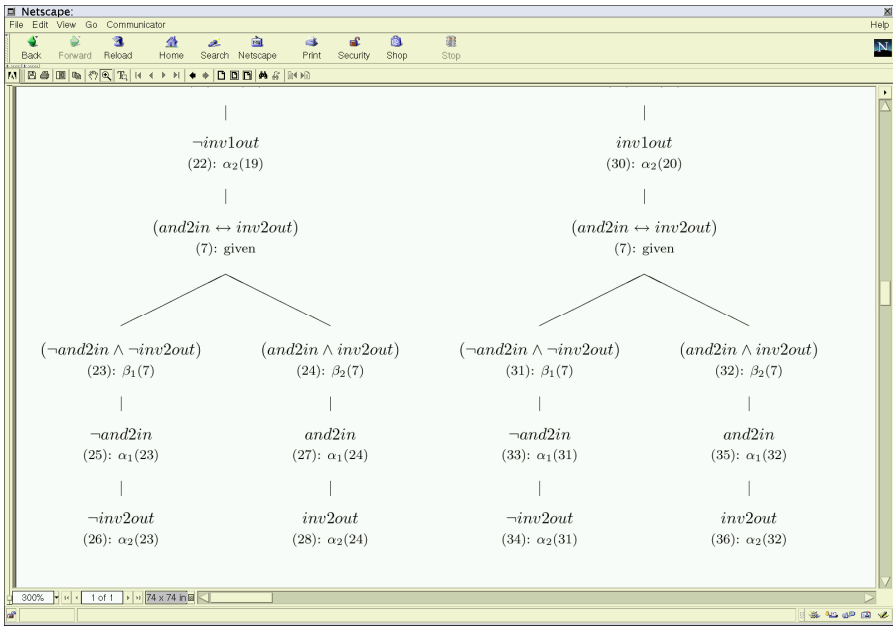

Figure 7: A zoomed-in view of the tableau in Figure 6.

3.2 System Architecture

The Living Book is a distributed system with a web based client interface and two servers realizing the main parts of the system: The content of the Living Book is provided by the SIT-Server. The SIT-Server maintains the units of one or multiple related books and keeps track of the user profiles and their personalized views onto the learning material. The SIT-Server communicates with the Interaction Server which is responsible for access to different mathematical and logic based systems embedded into the learning material.

The Interaction Server is responsible for integration of interactive components related to the Living Book. This server maintains the database of interactive system components from one and even possible from multiple books. The Interaction Server wraps the different mathematical and logic systems and provides one common interface to different connected systems. Furthermore it maintains the user profiles related to the interaction with the Living Book.

The documents delivered to the user are in the PDF format. They are individually compiled from sources written in the LaTeX typesetting language. This choice was motivated by the need for excellent typographic quality, and the possibility to accomodate interactive elements at the same time. Figure 8 illustrates the architecture.

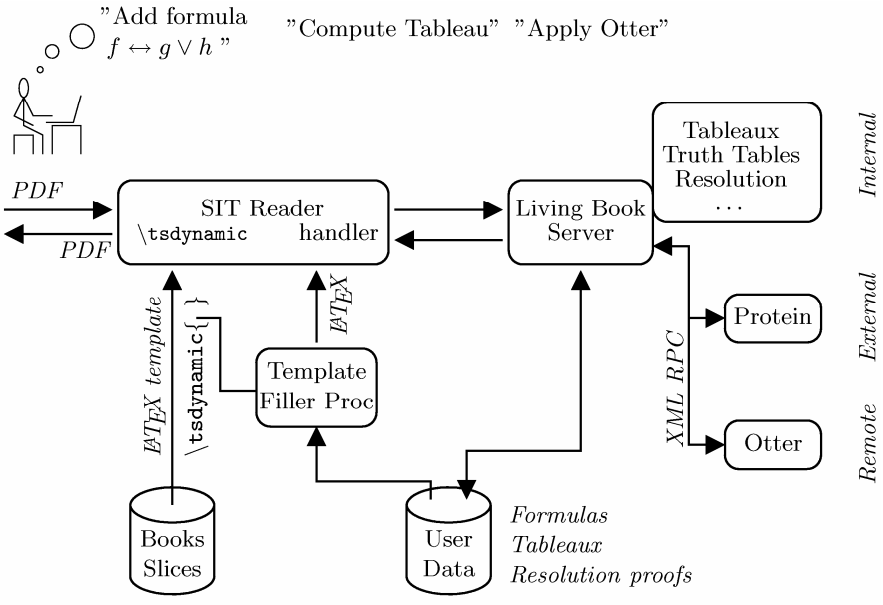

Figure 8: Living Book system architecture.

4 What's about Didactics?

By the previous description of the system and the example given, it may have become clear, that this new kind of material can be used in various ways for individual learning. The functionality of Living Books also supports collaborative learning in groups.

In the teaching environment of the computer science courses in Koblenz we used Living Books during the previous semesters in different ways:

- The material can be used by the *individual student* to access the material presented by her teacher. In this respect, our work differs e.g. from Intelligent Tutoring Systems [Cor+97], providing individualized tutoring or instruction. The Living Book is a tool used within the didactical context of a lecture. Students can access the material anywhere: at home or through wireless LAN, which is covering the entire Campus of the University in Koblenz. Thus the material can be used by the student at any place for individual learning and preparation of exercises.
- The *teacher* can decide which parts of the material she wants to use in her course and by the various search mechanisms she can compose an individual version of the book. By this it is possible to leave out certain parts or to select a topic and then ask the book to automatically include those parts of the book, that are necessary for its understanding. This personalized version he can hand out the student as the course material. For these purposes we use the e-learning platform WebCT. For a careful description of this aspect see [Dah+02].
- Learning in *groups*, e.g. [Wess01], also is supported by the Living Book technology. Students can solve exercises in tutoring groups with the help of Living Book; afterwards they communicate about which slices of the material are helpful for finding solutions. By this it is as well possible to design new forms of teaching: the teacher is presenting material in a conventional class room situation, but from time to time he is alternating this with phases where the students have to cooperate in small groups towards a result, which then is discussed again within the entire class.

Those new forms of teaching are investigated intensively in the DFG Special Priority Program (Schwerpunktprogramm) "Netbased Knowledge Communication in Groups". The aim of this program is to develop an interdisciplinary research agenda (addressing the fields of cognitive science, social psychology, empirical educational psychology, computer science) that helps generating new research hypotheses and shape the future development in this area of research.

We aim at establishing "Net-Based Knowledge Communication" as a new and interdisciplinary research field. Within this field theoretical, empirical, and practical aspects of new communication media will be investigated and integrated.

We are working in this program towards an application of the synchronicity theory (see e.g. [Schw02]).

5 Related Work

Various interactive and personalized E-learning systems have been discussed in the literature.

In [Pri+00] an interactive electronic book (*i-book*) is presented. This i-book is devoted to teaching adaptive and neural systems for undergraduates in electrical engineering. The salient feature of this book is the tight integration of simulators demonstrating the various topics in adaptive systems and the incremental use of simulation during each chapter in order to develop successively on a certain subject. The i-book, though, does not cope with different learning scenarios or user profiles and offers the same documents for every student.

The paper [BrKr02] discusses perspectives for electronic books and emphasizes the need for personalized and user specific content. This article concentrates on personalized presentation of content, for instance by means of style sheet application to the content that is delivered to the user. Personalization applied to the content of the material as done in our approach is not considered.

Based on an explicit representation of the structure of the concepts in the domain of interest and a user model [Vass97] and [Mel+01] dynamically generates instructional courses. These approaches use planning techniques in order to determine the relevant materials on a per user basis. The user model in [Vass97] describes the student's knowledge, and contains history information about previous sessions as well as personal traits and preferences. Interactivity is not integrated in the works described by [Vass97]. In [Mel+01] an interactive and adaptive system is presented. Scenarios and user profiles are supported. Here, user profile distinguishes between *knowledge*, *comprehension* and *application* in order to reflect the different status of knowledge during learning.

These two approaches differ from Living Books in two main aspect: firstly, in Living Books, we have chosen a deduction based approach instead of planning techniques, and secondly, the user profile adapts according to what the users specify that they know. For instance, the user indicates those units that are already known. From this the system deduces everything that should be known, too, based on dependence relationships between knowledge units. In [Vass97;Mel+01], the user model is adapted based on information the system gathers from a user during a session, e.g. if a certain exercise has been successfully solved.

6 Conclusion

In this paper we have presented the Living Books developed at the University of Koblenz. The main goal of the Living Books is the support of explorative learning for undergraduates in mathematics and computer science.

In the area of computer based learning, different learning environments are developed to be used for training and education purposes in business and academic institutions. Often, the goal is to replace lectures given by a teacher by an online-class in a virtual learning environment. In this work we have demonstrated the integration of personalized and interactive learning material into existing learning environments. With Living Books not only traditional forms of teaching might be supported; but, Living Books also effectively can be used in cooperative learning scenarios and thereby achieve a qualitative enhancement of current teaching methods.

The main achievements in Living Books are on the one hand side the dynamic assembly of personalized and learning scenario specific documents and on the hand side the support for an interactive use of the material. Personalized and scenario specific generation of documents allow to make efficient use of the material. This, in particular, is important, when course are offered in virtual learning environments where learners with different prerequisite are participating in a course. The integration of interactive tools furthermore provide a uniform interface to different kinds of problem solving systems. The learner does not have to deal with various input formats and with the details of how to use the different systems. Furthermore, by integrating interactive systems directly into the material, the results of applying these systems also are integrated in the personalized documents for the user. Thereby, the user always has a complete status of the work done in hands.

The techniques developed in the Living Books are quite general and can be reused in other context, too. The knowledge management system is a general purpose tool for personalized generation of documents [BaFu02] and may be used e.g. also in order to generate from a given complete system description those parts that are relevant for a certain user who deals with few functionality, only.

References

[BaFu02] Baumgartner, P.; Furbach, U.: Automatic Techniques for the Management of Personalized Documents. Annals of Mathematics and Artificial Intelligence – Special Issue on Mathematical Knowledge Management (Kluwer Academic Publishers, 2002). Accepted for Publication.

[Bau+03] Baumgartner, P.; Gross-Hardt, M.; Sinner, A.: Living Book – Deduction, Slicing and Interaction. Fachberichte Informatik 2-2003, Universität Koblenz-Landau, Universität Koblenz-Landau, Institut für Informatik, Rheinau 1, D-56075 Koblenz, 2003. submitted to Journal of Automated Reasoning.

[BrKr02] Bry, F.; Kraus, M.: Perspectives for electronic books in the world wide weg age. The Electronic Library Journal 20, 4 (2002), 275-287.

[Cor+97] Corbett, A.; Koedinger, K.; Anderson, J.: Intelligent tutoring systems. In: Handbook of human-computer interaction, P.P.V. Helander M.G., Landauer T.K. Eds., The Netherlands: Elsevier Science (1997), 849-874.

[Dah+02] Dahn, B. I.; Armbruster, M.; Furbach, U.; Schwabe, G.: Slicing books – the authors' perspective. In: Writing hypertext and learning (New York, Oxford, 2002), R. Bromme and E. Stahl, Eds., Pergamon Press.

[Dahn01] Dahn, I.: Slicing book technology – providing online support for textbooks. In: Proc. of the 20th World Conference on Open and Distance Learning (Düsseldorf/ Germany, 2001), H. Hoyer, Ed.

[Mel+01] Melis, E.; Andrea, E.; Büdenbender, J.; Frischauf, A.; Goguadze, G.; Libbrecht, P.; Pollet, M.; Ullrich, C.: Activemath: A generic and adaptive web-based learning environment. Journal of Artificial Intelligence and Education 12, 4 (2001), 385-407.

[Pri+00] Principe, J. C.; Euliano, N. R.; Lefebvre, W. C.: Innovating adaptive and neural systems instruction with interactive electronic books. Proc. IEEE 88, 1 (2000), 81-95.

[Schw02] Schwabe, G.: Mediensynchronizität beim Lernen. In Proc. GI-Jahrestagung (2002).

[SiVa02] Simon, A.; Valerius, M.: User Requirements – Lessons Learned from a Computer Science Course. Fachberichte Informatik 5-2002, Universität Koblenz-Landau, Institut für Informatik, Rheinau 1, D-56075 Koblenz, 2002.

[Vass97] Vassileva, J.: Dynamic courseware generation at the www. In: Proc. Of the 8th World Conference on AI and Education (AIED'97) (Kobe, Japan, 1997).

[Wess01] Wessner, M.: Ein kollaboratives Lernmodell für CSCL-Umgebungen. In: 5. Internationale Tagung Wirtschaftsinformatik 2001. H.U. Buhl, A. Huther, B. Reitwiesner (Hrsg.), Information Age Economy. Heidelberg: Physica-Verlag, 2001, pp. 367-380.