Adaptive Systems for Web-based Education

Proceedings of the AH'2002 Workshop on
Adaptive Systems for Web-based Education
Málaga, Spain, May 2002
Selected papers

Peter Brusilovsky, Nicola Henze and Eva Millán (Eds.)

Universidad de Málaga
Departamento de Lenguajes y Ciencias
de la Computación

Volume editors

Peter Brusilovsky
School of Information Sciences
University of Pittsburgh
135 North Bellefield Ave., Pittsburgh, PA 15260, USA
e-mail: peterb@mail.sis.pitt.edu

Nicola Henze
Institut für Technische Informatik
University of Hannover
Appelstraße 4 . 30167 Hannover. Germany
email: henze@kbs.uni-hannover.de

Eva Millán
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga
Bulevar Louis Pasteur, 35. Málaga 29071. Spain
e-mail: eva@lcc.uma.es

ISBN: 84-699-8192-7
Depósito Legal: MA-401-2002
Impreso en España - Printed in Spain

Table of Contents

Preface

Adaptive Web-based educational systems provide an alternative to the traditional "one-size-fits-all" approach in the development of Web-based educational courseware. These systems build a model of the goals, preferences and knowledge of each individual student, and use this model throughout the interaction with the student in order to adapt to the needs of that student. The first pioneer adaptive Web-based educational systems were developed in 1995-1996 (Brusilovsky, Schwarz & Weber, 1996a; Brusilovsky, Schwarz & Weber, 1996b; De Bra, 1996; Nakabayashi et al., 1995; Okazaki, Watanabe & Kondo, 1996). Since that many interesting systems have been developed and reported. An interest to provide distance education over the Web has been a strong driving force behind these research efforts. A good help for the research community was provided by a sequence of workshops that get together researchers working on Adaptive Web-based educational systems, let them learn from each other, and advocate the ideas of this research direction via on-line workshop proceedings (Brusilovsky, Nakabayashi & Ritter, 1997; Peylo, 2000; Stern, Woolf & Murray, 1998). The workshop at AH'2002 is the fourth in these series of workshops. As earlier, the proceedings of this workshop assemble together an interesting collection of papers on various topics associated with adaptive Web-based educational systems. We hope that this volume will serve as another milestone for this research direction and will serve as a source of creative ideas for the researchers worldwide.

In total, 14 full papers and 7 short papers were submitted to the workshop. From this number, the program committee selected 8 contributions to be presented as full papers and 6 to be presented as short papers. 7 contributions were approved as position papers and are published on the workshop Web pages (http://www.lcc.uma.es/WASWE2002). Below we provide a brief summary of accepted full and short papers.

E. Triantafillou, A. Pomportsis and E. Georgiadou address a hard problem of accommodating to the cognitive style of learners in their paper entitled "AES-CS: Adaptive Educational System based on Cognitive Styles". They discuss field dependence as a measurable cognitive parameter and propose several ways to adapt to field-dependent and field-independent learners. In "Adaptive Web-based Learning System with a Free-Hyperlink Environment", H. Mitsuhara, Y. Ochi, and Y. Yano present an adaptive web-based learning system that contains a free hyperlink environment, which enables learners to create and share hyperlinks in the open web. These hyperlinks are adapted using collaborative filtering in order to avoid option overflow. J. Ohene-Djan introduces in "Ownership Transfer via Personalisation as a Value-adding Strategy for Web-based Education" a model for the personalization of hyperdocuments. The use of this model allows an interaction with electronic documents that is much closer to the manner that users interact with paper-based documents, without losing the advantages of hypermedia systems. M. Baldoni, C. Baroglio, V. Patti, and L. Torasso describe in "Using a rational agent in an adaptive web-based tutoring system" an adaptive tutoring system having a multi-agent architecture. In their approach to adaptation, they use an agent logic programming language (DyLOG) for building cognitive agents that, using reasoning techniques, help users find the solutions to their needs. "An Adaptive Web-based Tutorial of Agrarian Economy" by C. Carmona, E.Guzman, D. Bueno and R. Conejo describes the process of making an already existing static web system adaptive. M. Trella, R. Conejo, D. Bueno and E.Guzman propose in "An Autonomous component architecture to develop WWW-ITS" the construction of a framework for the development and integration of distributed, autonomous educational components that can communicate with each other. "Automatic Generation of a Navigation Structure for Adaptive Web-Based Instruction", by J. Masthoff, proposes the construction of a hierarchical navigation structure on the base of a concept network, outcomes and prerequisites of concept, descriptive information about the author and the content of the pages itself. C. Karagiannidis and D. Sampson address the common ground between adaptive educational systems and e-learning standards. Their paper “Re-using Adaptation Logics for Personalized Access to Educational e-Content” bridges the gap between adaptive systems and re-usable courseware by proposing a way to encapsulate adaptation logics in an adaptive Web system to make it re-usable in a different context.

The collaborative process of writing stories is described by J. Suhonen in "Using StoryML for adaptive Content Representation". Investigations about the design of "SHAAD: Adaptable, Adaptive and Dynamic Hypermedia System for content delivery" have been taken by D. Merida, R. Fabregat and J.-L. Marzo. To "Facilitate Navigation Planning in Self-directed Learning on the Web", A. Kashihara, S. Hasegawa and J. Toyoda use previews of a planned navigation sequence to support orientation and guidance in a web-based learning environment. J. Smid and P. Svacek work in "Interactive Tutoring Model Using Information Cycling" on the optimization of a user's path through a domain by determining the appropriate number of recapitulations and presentations. C. Romero, S. Ventura , C. de Castro, W. Hall , and M. Hong Ng show in "Using Genetic Algorithms for Data Mining in Web-based Educational Hypermedia Systems" how to apply genetic algorithms for data mining to infer from student’s interactions rules that will help to improve the system. Finally, D. Chesher, J. Kay and N. King report in "Simprac - Teaching system for management of chronic illness" how learner can practice the consultation of patients by investigating the individual consequences of their chosen treatment of patients.

References

Brusilovsky, P., Nakabayashi, K., and Ritter, S. (eds.) (1997) Proceedings of the Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education. Kobe, Japan: ISIR, available at http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/.

Brusilovsky, P., Schwarz, E., and Weber, G. (1996a) ELM-ART: An intelligent tutoring system on World Wide Web. In: C. Frasson, G. Gauthier and A. Lesgold (eds.) *Intelligent Tutoring Systems*. Lecture Notes in Computer Science, Vol. 1086, (Proceedings of Third International Conference on Intelligent Tutoring Systems, ITS-96, Montreal, June 12-14, 1996) Berlin: Springer Verlag, pp. 261-269.

Brusilovsky, P., Schwarz, E., and Weber, G. (1996b) A tool for developing adaptive electronic textbooks on WWW. In: H. Maurer (ed.) Proceedings of WebNet'96, World Conference of the Web Society, San Francisco, CA, October 15-19, 1996, AACE, pp. 64-69, also available at http://www.contrib.andrew.cmu.edu/~plb/WebNet96.html.

De Bra, P. M. E. (1996) Teaching Hypertext and Hypermedia through the Web. *Journal of Universal Computer Science* 2 (12), 797-804, http://www.iicm.edu/jucs_2_12/teaching_hypertext_and_hypermedia.

Nakabayashi, K., Koike, Y., Maruyama, M., Touhei, H., Ishiuchi, S., and Fukuhara, Y. (1995) An intelligent tutoring system on World-Wide Web: Towards an integrated learning environment on a distributed hypermedia. In: H. Maurer (ed.) Proceedings of ED-MEDIA'95 - World conference on educational multimedia and hypermedia, Graz, Austria, June 17-21, 1995, AACE, pp. 488-493.

Okazaki, Y., Watanabe, K., and Kondo, H. (1996) An Implementation of an intelligent tutoring system (ITS) on the World-Wide Web (WWW). *Educational Technology Research* 19 (1), 35-44.

Peylo, C. (ed.) (2000) Proceedings of the International Workshop on Adaptive and Intelligent Web-based Education Systems held in conjunction with 5th International Conference on Intelligent Tutoring Systems (ITS'2000). Ösnabrück: Institute for Semantic Information Processing, University of Ösnabrück, available at http://virtcampus.cl-ki.uni-osnabrueck.de/its-2000/.

Stern, M., Woolf, B. P., and Murray, T. (eds.) (1998) Proceedings of the workshop on Intelligent Tutoring Systems on the Web at 4th International Conference on Intelligent Tutoring Systems (ITS'98). San Antonio, TX: St. Mary University, available at http://www-aml.cs.umass.edu/~stern/webits/itsworkshop/.

Workshop chairs

Peter Brusilovsky	(University of Pittsburgh, USA)
Nicola Henze	(University of Hannover, Germany)
Eva Millán	(Universidad de Málaga, Spain)

Program Committee

Peter Brusilovsky	(University of Pittsburgh, USA)
Nicola Henze	(University of Hannover, Germany)
Eva Millán	(Universidad de Málaga, Spain)
Tanja Mitrovic	(University of Canterbury, New Zealand)
Marcus Specht	(Fraunhofer FIT, Germany)
Julita Vassileva	(University of Saskatchewan, Canada)

External Reviewers

Gordon McCalla	(University of Saskatchewan, Canada)
Tiffany Tang	(University of Saskatchewan, Canada)

AES-CS: Adaptive Educational System based on Cognitive Styles

Evangelos Triantafillou[1], Andreas Pomportsis[1], Elissavet Georgiadou[2]

[1] Computer Science Department
Aristotle University of Thessaloniki
P.O. Box 888, 540 06 Thessaloniki, Greece
{vtrianta, apompo} @csd.auth.gr

[2] Teacher in Secondary Education, Researcher
University of Macedonia, Greece
elisag@otenet.gr

Abstract: This paper describes the design and development of an Adaptive Educational System (AES) that includes accommodations for cognitive styles in order to improve student interactions and learning outcomes. Although, cognitive styles are one of the several important factors to be considered from designers and instructors of hypermedia-based courseware, little research has been done regarding the adaptation of hypermedia system to students' cognitive styles. Our research is an attempt to examine some of the critical variables, which may be important in the design of an adaptive hypermedia system based on student's cognitive style.

1 Introduction

The phenomenal growth of the Internet and the Web over the last years has led to an increasing interest in creating Web-based learning tools and learning environments. Hypermedia seems to be suitable for supporting the new constructivist way of active and self regulated learning. However, empirical studies have shown contradictory results about the efficiency and effectiveness of learning with hypermedia. Some studies indicate that hypermedia-based learning may contribute to enhance learning and promote cognitive flexibility when the learning environment is designed task appropriately [11]. At the same time, other studies have revealed problems for hypermedia-based learning with regards to cognitive overload and disorientation [8].

In order to overcome the problems identified, a hypermedia system should be designed in a way that can identify the user's interests, preferences and needs and give appropriate guidance throughout the learning process. Adaptive Hypermedia (AH) was introduced as one possible solution. Adaptivity is especially important for Web-based educational hypermedia, as these systems are expected to be used by several learners without assistance of a physical tutor who usually can provide adaptivity in an actual educational environment, i.e. classroom.

Adaptive Hypermedia systems can be developed to accommodate various learner needs; is the ideal way to accommodate a variety of individual differences, including learning style and cognitive style [1]. Numerous Adaptive Hypermedia systems have been implemented over the last fifteen years. INSPIRE [10] and CS383 [4] are good examples of Adaptive Educational Systems with regards to learning style. Although, cognitive styles are one of the several important factors to be considered from designers and instructors of hypermedia-based courseware, little research has been done regarding the adaptation of hypermedia system to students' cognitive styles [7] and this is the focus of our research.

Our research is an attempt to examine some of the critical variables, which may be important in the design of an adaptive hypermedia system based on student's cognitive style. As a case study a Higher Education module was developed, called AES-CS (Adaptive Educational System based on Cognitive Styles), to support the course "Multimedia Technology Systems" which is typically offered to fourth year undergraduate students in Computer Science Department at the Aristotle University of Thessaloniki, Greece. Before we proceed to describe the architecture and implementation of AES-CS system it is important to discuss on cognitive styles as considerable confusion appears in the literature regarding the terms cognitive style and learning style. Moreover, we will examine the design issues that were considered for the development of the system that are reported in the relevant literature and should be taking into account from instructional designers of adaptive hypermedia.

2 Cognitive Styles

There is a technical difference between the use of the terms cognitive style and learning style, although numerous authors use the terms interchangeably. Cognitive style deals with the 'form' of cognitive activity (i.e., thinking, perceiving, remembering), not its content. Learning style, on the other hand, is seen as a broader construct, which includes cognitive along with affective and physiological styles.

Cognitive style is usually described as a personality dimension, which influences attitudes, values, and social interaction. It refers to the preferred way an individual processes information. There are many different definitions of cognitive styles as different researchers emphasize on different aspects. However, Field dependence/independence (FD/FI) is probably the most well known division of cognitive styles [14].

FD/FI dimension refers to a tendency to approach the environment in an analytical, as opposed to global, way. Studies have identified a number of relationships between this cognitive style and learning, including the ability to learn from social environments, types of educational reinforcement needed to enhance learning, amount of structure preferred in an educational environment [12].

Field independent (FI) learners generally are analytical in their approach while Field Dependent (FD) learners are more global in their perceptions. Furthermore, FD learners have difficulty separating the part from the complex organization of the whole. In other words, FD individuals see things in the entire perceptual field (the forest than the trees). Additionally, FI individuals tend to be intrinsically motivated and enjoy individualized learning, while FD ones tend to be extrinsically motivated and enjoy cooperative learning. Specifically, FD individuals are more likely to require externally defined goals and reinforcements while the FI ones tend to develop self-defined goals and reinforcements [14].

3 Design Issues for Adaptive Hypermedia

In the ideal educational environment, a tutor with instructional experience on a learning domain can identify students' individual differences, with regards to cognitive styles and acquired knowledge, and thus can provide them with learning material individually selected and structured. Moreover, the interaction that takes place in a physical classroom allows tutors to experience and understand students' personal goals and preferences and thus to promote their skills. In order to simulate in a sense an ideal educational environment, an adaptive hypermedia system should provide learners the ability to use different instructional modes in order to accommodate their individual needs and to improve their performance. Therefore, it has to include in its design both issues of cognitive style and teaching strategy. Teaching strategy refers to the instructional material and the instructional strategy. Table 1 presents the instructional strategies adopted in AES-CS that support students according to their cognitive style. Next we will discuss briefly the most important of these strategies and the way they were implemented in the design of AES-CS.

Table 1. Instructional Strategies.

Field-Dependent learners	**Field-Independent learners**
Provide global approach	Provide analytical approach
Provide information from general to specific	Provide information from specific to general
Program control	Learner control
Provide advance organizer	Provide post organizer
Provide maximum instructions	Provide minimal instructions
Provide maximum feedback	Provide minimal feedback
Provide structured lessons	Allow learners to develop their own structure
Provide Graphics Path Indicator	
Provide social features	Provide individual environment

Program control versus learner control: The amount of learner control seems to be a central variable when integrating adaptive methods in educational settings. There are several arguments in the literature for and against learner control. On the one hand, learners' motivation is increased when they control the navigation of a hypermedia environment. On the other hand, research seems to indicate that the amount of learner control depends on the pre-skills and the knowledge state of a learner [13]. Furthermore, many studies have demonstrated student preference and improved performance using a linear structure. With regards to cognitive styles, there is evidence that FD individuals perform better using program control while FI ones prefer more learner control [16]. Since these findings are consistent with theoretical assumptions in FD/FI dimension, AES-CS provides both program and learner control option. In the case of learner control option, AES-CS provides a menu from which learner can choose to proceed the course in any order [Fig.1]. In the program control option there is no menu, but the system guides the user through the learning material via adaptive navigation support [see Fig.2 on next page].

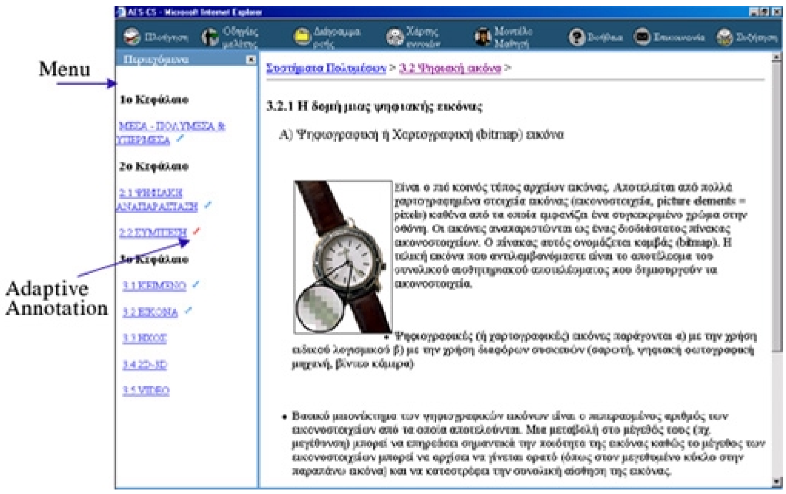


Fig.1. System screen with the initial adaptation for FI learners.

Instructions and Feedback: Studies have shown that FD are holistic and require external help while FI people are serialistic and possess internal cues to help them solve problems. FD learners are more likely to require externally defined goals and reinforcements while FI tend to develop self-defined goals and reinforcements [14]. Jonassen and Grabowski [5] in their study summarized the research on the implications of the individual differences based on FD/FI dimension. We consider these implications of style characteristics in order to design the instructional support and the instructional environment of AES-CS. As a result, the system provides clear, explicit directions and the maximum amount of guidance to FD learner, while it provides minimal guidance and direction to FI learner. Moreover, it provides extensive feedback to FD learner, while it provides minimal feedback to FI learner.

The use of contextual organizer: Another feature that is embedded in AES-CS is the use of contextual organizers according to FD/FI dimension. Field Dependent learners appeared to benefit most from illustrative advance organizers, while Field Independent learners preferred illustrative post organizers [9]. An advance organizer is a bridging strategy that provides a connection between one unit and another. It also acts as a schema for the learner to make sense out of the new concept. A post organizer serves as a synopsis and supports the reconstruction of knowledge. Usually, it is available after the presentation of new information.

Structure: Several problems of learning in a hypermedia environment arise from the structure of the environment itself. In an ideal web site, the structure is evident to the user and the information is organized coherently and meaningfully. Navigational tools are essential in order to assist learners to organize the structure of the web site as well as the connections of the various components. A coherent resource collection will allow the user to construct an accurate mental model of the topic.

Research has indicated that FD learners are less likely to impose a meaningful organization on a field that lacks structure and are less able to learn conceptual material when cues are not available [14]. Furthermore, Jonassen and Wang [6] argue that the FI learners generally prefer to impose their own structure on information rather than accommodate the structure that is implicit in the learning materials. In our approach, AES-CS provides two navigational tools in order to help learners organize the structure of the knowledge domain: concept map and graphic path indicator [Fig.2].

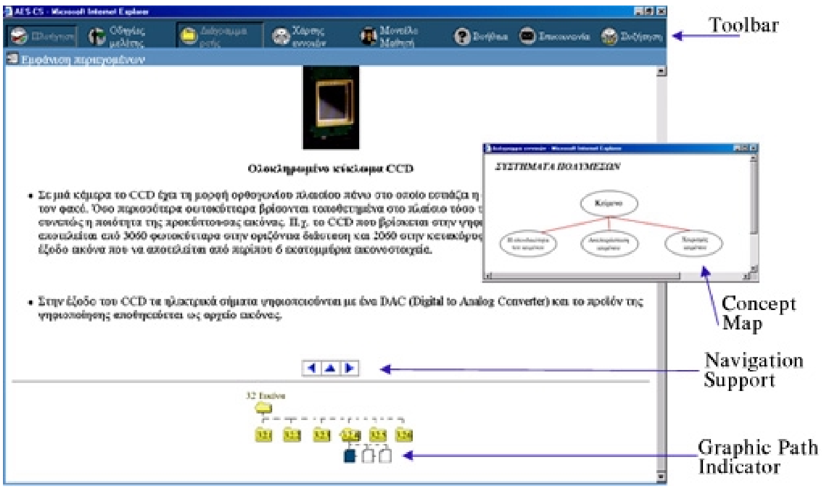


Fig.2. System screen with the initial adaptation for FD learners.

Concept map is a visual representation of a knowledge domain and consists of nodes representing concepts connected by directional links that define the relationships of the nodes. Concept maps may act as tools to aid study and assist to the comprehension of a domain. Furthermore, they are flexible tools, which can be used by students to develop their own maps to represent various domains of knowledge and can be used to see the forests and the tress thereby avoiding disorientation. In AES-CS concept map is used to help FD learners understand the big picture and place detail in perspective.

The graphic path indicator can orient users to the surrounding hyperspace and to the content organization, affecting both cognitive overhead and coherence. The graphic path indicator is dynamically created and presents the current, the previous and the next topic. The graphic path indicator appears at the bottom of each page and illustrates clearly the local neighborhood of a topic.

4 System Architecture and Implementation

The main characteristic of AES-CS is that it can be adapted to the cognitive style and to the level of knowledge acquired by the student. The system is organized in the form of three basic modules: the domain model, the student model, and the adaptation module [Fig.3]. These three components interact to adapt different aspects of the instructional process, i.e. adapting the content according to user's prior knowledge; adapting the presentation of contents through selection and combination of appropriate media; adapting the teaching strategies; modifying the selection of examples and links; and recommending appropriate hyperlinks.

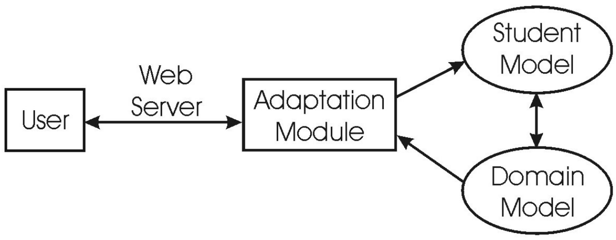


Fig.3 System architecture

4.1 Domain model

The domain model is a set of domain concepts. It serves as a basis for structuring the content of AES-CS. Each concept is structured into a set of topics. Topics represent basic pieces of knowledge for the given domain and their size depends on the domain. Topics are linked to each other thus forming a kind of semantic network. This network is actually the structure of the knowledge domain. In AES-CS each hypermedia page actually corresponds to one topic only.

4.2 Student model

The student model needs to be easy to construct and modify and should accurately reflect the characteristics of different students. Three different categories of information are built-in in the student model: personal profile (which includes static data e.g.

name and password), cognitive profile (which includes adaptable data like cognitive style preferences), and an overlay student knowledge profile (which illustrates student's knowledge on a subject). Table 2 shows the description of some attributes included in each category, and provides information on the possible values types of the attributes and how these can be acquired.

Table 2. The student model

Type	Item	Value	How acquired
Personal Profile	Name	Free text	user
	Password	Free text	user
Cognitive Profile	Cognitive style	FD or FI	user or system
	Program Control	Yes or No	user or system
	Learner Control	Yes or No	user or system
	Advance Organizer	Yes or No	user or system
	Post Organizer	Yes or No	user or system
	Graphics Path Indicator	Yes or No	user or system
Knowledge Profile	Concept 1	Unknown Know Learned Well- Learned	user or system
	Concept 2	- //-	-//-

In our research the Group Embedded Figures Test (GEFT) [15] was used to identify the field-dependent and field-independent cognitive style. In this test, subjects perceived the information, which is a series of simple figures, independently from the larger complex figure, in which the simple figures are embedded.

4.3 Adaptation module

To support adaptivity, AES-CS uses the 'adaptive presentation technique' [2] that aims to adapt the information presented to the user according to his/her cognitive style and knowledge state. Conditional text and page variants representations are used to accomplish adaptive presentation. With the conditional text technique, a page is divided into chunks. Each chunk of information is associated with a condition indicating which type of user should be presented with it. With page variants technique, two variants of the pages associated with a concept are prepared. Each variant of the page presents information in a different style according to FD/FI dimension.

'Adaptive navigation support' is a specific adaptive hypermedia technology that aims to help users to find an appropriate path in a hypermedia-learning environment [2]. AES-CS does not include adaptive navigation support in the case of learner control option. However, in program control option, AES-CS provides adaptive navigation support by manipulating the selection and the presentation of links through adaptive annotation and direct guidance. Adaptive annotation of hyperlinks supplies the user with additional information about the content behind a hyperlink. The selection and the colour of hyperlinks are adapted to the individual student by taking into account information about the learner's knowledge state and the instructional strategy. Blue colour is used for 'recommended' and gray colour for 'not ready to be learned'. With the direct guidance, the system suggests to the student the next part of the learning material. This technique can be seen as a generalization of curriculum sequencing but within the hypermedia context it offers more options for direct guidance. Student's prior knowledge is used by the system in order to provide him/her the most suitable sequence of knowledge units to learn and to work with.

Furthermore, an annotation mechanism is used to show several levels of student's knowledge on each domain model concept. A colored checkmark is used to distinguish the state of student knowledge on any concept: a blue checkmark means 'know' (determined by the student through the student model), a red checkmark means 'learned' (the student has visited the pages which presents the concept) and a green checkmark means 'well-learned' (the student has successfully completed a test) [see Fig.1].

5 Evaluation

The current interface of AES-CS and its functionalities are the result of revisions based on the analysis of the data collected during the formative evaluation. Formative evaluation is the judgments of the strengths and weakness of instruction in its developing stages, for the purpose of revising the instruction. The classically recognized types of formative evaluation are: expert review, one-to-one evaluation, small group, and field test [17].

Until now, the expert review, the one-to-one, and the small group evaluation were completed. In the expert review, a semi-structured interview aimed at determining the reactions of experts and a debriefing session were used. Five experts acted as evaluators in this phase: a teaching/training expert, an instructional design expert, a subject-matter expert, a subject sophisticates and an educational technologist. The subjects for the one-to-one evaluation and the small group were fourth year undergraduate students studying the course "Multimedia Technology Systems" in Computer Science Department at the Aristotle University of Thessaloniki, Greece. Ten subjects participated in the one-to-one evaluation and a semi-structured interview and debriefing session were used. Finally, ten subjects took part in the small group evaluation. We used several methods for getting information from students during this phase. First, a pre-test and a post-test were used to help us to measure the learning gain from the instruction. Second, an attitude questionnaire aiming to determine subjects' experience in using the system. Third, comment logs were used in order subjects to note specific strengths or weaknesses of the system during the instruction. Finally, debriefing sessions were used to measure the subjective satisfaction of subjects on the instructional and interface design of AES-CS.

A preliminary analysis of the data collected showed that the subjects were satisfied with the adaptation based on cognitive style. In addition, they felt that the system was clear and easy to understand and after working with it they had a better understanding of the area studied. However, they made suggestions for the improvement of the system. Some of the more significant recommendations that were implemented during the revision phase are as follows:

On student model
The subjects agreed that the student model was easy to use and modify, but most of them pointed out as a design weakness the scrolling of the html page that includes all the available options (level of knowledge, cognitive style, choice of characteristics) in one-single page.

On user interface
Most of the subjects suggested that they would prefer to have an option regarding the ability to adapt the background of the html pages. Finally several subjects recommended that the content menu should always appear at the left frame irrespective the program or learner control option.

On tools
Subjects agreed on the usefulness of the instructional guidance at the bottom of the screen but they suggested that it should be included in the minimum amount of space possible with regards to the overall appearance of the screen. Moreover, all the subjects suggested that the concept map and the graphic path indicator should be active so to be used as an extra navigation tool.

Other observations
All the subjects considered the self-assessment unit very useful but they suggested the need of additional information regarding the number of total questions in the system and the right and wrong answers. Moreover, few subjects recommended that they want to see the path they follow in order to reach a particular page.

As mentioned earlier, most of the above suggestions were already implemented during the revision of AES-CS. However, they will be further processed in order to develop more the adaptive features of AES-CS and to fully complete the design of the system. After the final revision, summative evaluation will follow in order to assess the effectiveness of the system with reference to other educational material used for the instruction of the particular module.

6 Conclusion and Further Research

In this paper we have described the design and development of an Adaptive Educational System based on Cognitive Styles (AES-CS), a prototype that includes accommodations for cognitive styles in order to improve student interactions and learning outcomes. Currently, the basic architecture of the system has been implemented. The AES-CS (http://mlab.csd.auth.gr/adapt, 'guest' for login and password) has two main components: the client running in the Web browser and the server running in a PC workstation. The client is implemented as Web pages, residing in a number of linked frames. The development platform used is ASP technology using Dynamic HTML and JavaScript language that made possible to overcome HTML limitations.

Current adaptive hypermedia systems provide adaptation based on a stereotypical user model with limited levels of user differentiation. However, the complexity of learner's profile stresses that a different approach of the user model should be considered. Users are not simply analytical or global learners but instead are some combination of both characteristics. According to Carver, Hill and Pooch [3] the AH systems should not only model multiple dimensions of the user, but each dimension should have as much delineation as necessary to truly model the user.

In that terms, the system AES-CS is differentiated from related projects to the fact that it provides a fine degree of adaptation granularity. Learners have the ability to change the initial stage through the student model and/or appropriate interactive features [see Fig.1, 2]. The learners may modify the control options between learner and program control, may choose minimal or maximum feedback, may request instructions and so on. Our basic assumption is that adaptive systems need to be controllable by the user because they cannot be intelligent enough to appropriately adapt in all possible cases.

Further research is on progress concerning the evaluation of the initial adaptation. The initial adaptation of AES-CS to FD/FI learners was based on research results [6,9,16] and theoretical assumptions in FD/FI dimension [5,14]. An important research direction is to investigate if this initial adaptation is the most appropriate for the learners and to verify the design issues that were considered for the development of the system.

Additional research direction is the evaluation of the educational effectiveness of system's adaptation. We are going to investigate the hypothesis that the adaptivity based on student's cognitive style could be beneficial for the observed learning outcomes. An experiment by Jonassen and Wang showed that FI learners are better hypermedia processors, especially as the form of the hypermedia becomes more referential and less overtly structured [6]. In that term, part of our further research is to examine whether or not FD learners will reach the same level of performance as FI ones when studying in AES-CS environment.

References

1. Ayersman, D.J. & Minden, A.V. (1995). Individual differences, computers, and instruction. Computers in Human Behavior, 11(3-4), 371-390.

2. Brusilovsky, P.(1996). Methods and Techniques of Adaptive Hypermedia. User Modeling and User-adapted Interaction. 6, 87-129.

3. Carver, C., Hill, M. & Pooch, U. (1999). Third Generation Adaptive Hypermedia Systems. WebNet 99, Honolulu, Hawaii, 1999.

4. Carver, C., Howard R.& Lavelle, E. (1996). Enhancing Student Learning by Incorporating Learning Styles into Adaptive Hypermedia. EDMEDIA 96, Boston, MA. 118-123

5. Jonassen, D.H & Grabowski, B.L (1993). Handbook of individual differences, learning & Instruction. Hiilsdale, NJ. Lawrence Erlbaum Associates.

6. Jonassen, D. & Wang, S. (1993). Acquiring structural knowledge from semantically structured hypertext. Journal of Computer-based Instruction, 20(1), 1-8.

7. Liu, Y., Ginther, D. (1999). Cognitive Styles and Distance Education. On-line Journal of Distance Learning Administration, 2,3.

8. Marchionini, G. (1998). Hypermedia and Learning: Freedom and Chaos. Educational Technology 28,8-12.

9. Meng, K. and Patty, D.(1991). Field-dependence and contextual organizers. Journal of Educational Research, 84(3), 183-189.

10. Papanikolaou K., Grigoriadou M., Kornilakis H. (2001). Instructional and Interface Design in an Adaptive Educational Hypermedia System. Panhellenic Conference in Human-Computer Interaction (PC-HCI 2001). Patra, Greece, 2001.

11. Spiro R.J., Feltovich, R.J., Jacobson, M.J. & Coulson, R.L. (1991). Cognitive Flexibility, constructivism and hypertext: Random access instruction for advanced knowledge acquisition in ill structured domains. Educational Technology, 31, 24-33.

12. Summerville, J.(1999). Role of awareness of cognitive style in hypermedia. International Journal of Educational Technology, 1(1).

13. Williams, M.D. (1993). A comprehensive review of learner-control: The role of learner characteristics. Paper presented at the annual meeting of the Association for Educational Communications and Technology, New Orleans, LA.

14. Witkin, H.A, Moore, C.A., Goodenough, D.R., Cox, P.W.(1977). Field-dependent and field-independent cognitive styles and their educational implications. Review of Educational Research, 47(1), 1-64.

15. Witkin, H.A, Ottman, P.K., Raskin, E., Karp, S.A. (1971). A manual for the Embedded Figures Tests. Palo Alto, CA. Consulting Psychologists.

16. Yoon, G.S.(1993). The effects of instructional control, cognitive style and prior knowledge on learning of computer-assisted instruction. Journal of Educational Technology Systems, 22(4), 357-370.

17. Tessmer M. (1993). Panning and Conducting Formative Evaluations. Kogan Page Limited.

An Adaptive Web-based Learning System with a Free-Hyperlink Environment

Hiroyuki Mitsuhara[1], Youji Ochi[1], Kazuhide Kanenishi[2], and Yoneo Yano[1]

[1] Dept. of Information Science and Intelligent Systems, Tokushima University,
[2] Center for Advanced Information Technology, Tokushima University,
2-1 Minamijosanjima, Tokushima, 770-8506, Japan
E-mail: {mituhara, ochi, marukin, yano}@is.tokushima-u.ac.jp
TEL: +81-88-656-7495, FAX: +81-88-623-2761

Abstract. A typical learning method using the Internet is exploratory learning, where learners construct knowledge through exploring the web autonomously. A problem with web-based exploratory learning is a learning impasse caused by inadequate hyperlinks. MITS, an adaptive web-based learning system, has been developed in order to avoid this learning impasse. This system contains a free-hyperlink environment, which enables learners to create hyperlinks in the open web and share the hyperlinks. Furthermore, the hyperlinks have been adapted using collaborative filtering in order to reduce option overflow.

1 Introduction

Rapid growth of the Internet enables us to acquire knowledge from numerous web pages that exist all over the world. In this situation, web pages are expected to be informative learning resources. A typical learning method using the Internet is exploratory learning, where learners construct knowledge through exploring the web autonomously. Incoherence and complexity characteristic of the web can enhance the learners' ability to select correct knowledge and associate newly acquired knowledge with previously acquired knowledge.

In web-based exploratory learning, however, there is a learning impasse caused by inadequate hyperlinks. If a page is not complete with adequate hyperlinks to fulfill individual learners' interests or goals, the learners cannot visit the next page smoothly. It may also be true that the learners cannot understand a page without adequate hyperlinks to make up for their lack of knowledge. Search engines, which can supply adequate hyperlinks (pages), help escape from such a learning impasse but occasionally return meaningless hyperlinks resulting in loads in search activity. Thus, inadequate hyperlinks lead learners to this learning impasse, which stagnates the learners' exploration activity and consequently lowers learning effectiveness.

Adaptive web-based learning systems, which alter the hypertext contents or structure on the basis of individual learners' characteristics, can avoid this learning impasse. For example, DynaWeb increases exploration paths by adding hyperlinks on the basis of interests [1]. WebDL guides learners to new exploration goals by giving educational advice on the basis of learning progresses [2]. ELM-ART indicates the difficulty level of pages on the basis of knowledge levels [3]. Almost all such systems limit the adaptation to the closed web (i.e. web pages inside a certain server) and do not avoid the learning impasse in the open web (i.e. arbitrary web pages). This occurs because the teachers cannot describe information indexes and adaptation rules on the open web. AHA [5], KBS Hyperbook [7], and ITMS [8] deal with the open web but their adaptations are not completely independent of the closed web. Recently, this limitation has become a major focus in the domain of adaptive hypermedia [4].

MITS (Multi-hyperlink Tailoring System), an adaptive web-based learning system, has been developed in order to avoid the learning impasse in the open web, especially that caused by inadequate hyperlinks to learners' interests. This system contains a free-hyperlink environment, which enables learners to create hyperlinks in the open web and share the hyperlinks. Furthermore, the hyperlinks have been adapted using collaborative filtering in order to reduce option overflow.

The remainder of this paper is organized as follows. Section 2 clarifies the free-hyperlink environment. Section 3 describes MITS, especially adaptations using collaborative filtering. Section 4 reports the results of a small-scale experiment through trial use. Finally, conclusions are made and future works are summarized.

2 Free-hyperlink Environment

2.1 Fundamental Idea

The simplest way to avoid a learning impasse cased by inadequate hyperlinks is to increase the hyperlinks. If there is a focus on the information sharing environment, this idea may be actualized. There have been some web-based learning systems with such an environment. For example, SharlokII facilitates problem solving by enabling learners to share various kinds of information such as questions, answers, comments, and hyperlinks [11]. ReCoNote facilitates knowledge construction by enabling learners to share web notes structured with mutual links and comments [10].

It is conceivable that satisfying pages for a learner certainly exist somewhere in the vast web space and those have already been found by other learners. A proposal for a web-based exploratory learning environment, which enables learners to create hyperlinks (i.e. free-hyperlinks) that connect informative pages in the open web and share the hyperlinks, is therefore required. This environment is called a free-hyperlink environment. The free-hyperlinks differ from ordinary hyperlinks as follows.

- **Non-fixed location:** A free-hyperlink is not fixed on a certain page and is automatically embedded in every page with a corresponding anchor (text). This feature provides learners with many exploration paths.

  The free-hyperlink to be embedded is chosen by means of text matching that merely compares its anchor to a page.
- **Multi-direction:** A free-hyperlink can connect one anchor to some pages. This feature, which is similar to the concept of an extended link in XML, provides

learners with not only many exploration paths but also various kinds of knowledge for the same anchor.

- **Comment inclusion:** A free-hyperlink includes a comment on a page written by a learner. This comment, which may show the trait of the page (e.g. "This page is easy to understand thanks to useful diagrams") or a motive for creating the free-hyperlink (e.g. "This free-hyperlink is created to examine up-to-date technology."), can be used as an indication to select an adequate one from some free-hyperlinks.

The free-hyperlink environment will organize scattered pages and consequently form large-scale teaching material with many adequate hyperlinks. In other words, numerous learners substitute for the teachers who have to complete adequate hyperlinks. Avoiding the learning impasse in this manner may facilitate web-based exploratory learning.

Furthermore, the free-hyperlink environment may encourage learners to reflect upon their study, since a learner will be able to recognize the difference in knowledge for the same anchor by following other learners' free-hyperlinks in addition to his/her own free-hyperlink.

An empirical study shows that 58% of an individual's pages were revisits [13]. This result may indicate, from the viewpoint of web-based exploratory learning, that learners frequently review. The free-hyperlinks, which can be regarded as a bookmark function embedded in web pages, will be actively created to simplify review activity. Thus, the necessity to create the free-hyperlinks is ensured and the free-hyperlink environment appears to function satisfactorily. Figure 1 shows the concept of the free-hyperlink environment.

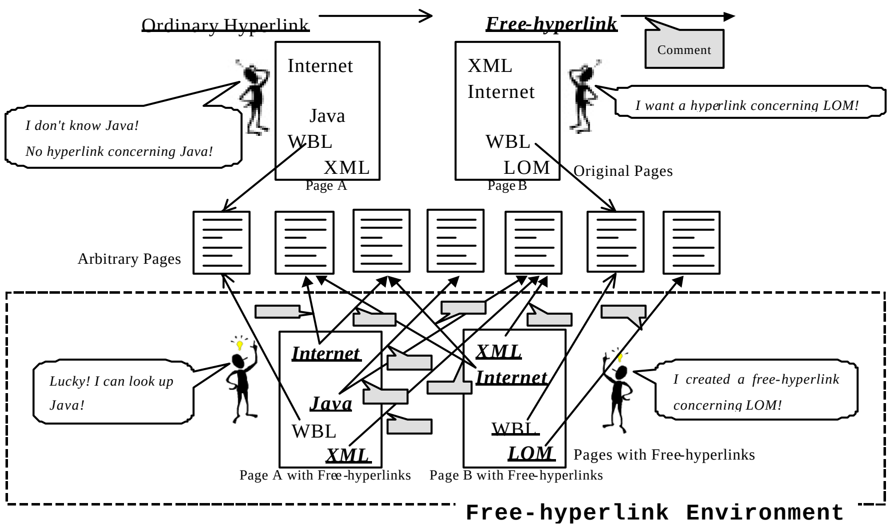


Fig. 1. The concept of free-hyperlink environment

2.2 How to Create Free-hyperlinks

When finding an interesting/unknown word(s) on a page, learners usually try to look up the word by using a search engine. Taking this process into account, a search engine has been integrated into the process of the free-hyperlink creation. Although this integration is probably time-consuming, the time will be minimized eventually since the free-hyperlinks can be shared. The free-hyperlink environment works together with a web retrieval support system WebCOBALT (Web Contents Observable System), which has a comment sharing environment for reducing loads in search activity [9]. Learners create the free-hyperlinks in the following process.

(1) Search query designation
A learner chooses an interesting/unknown word(s) as the search query by dragging the word on a page. Thereupon, a pop-up box is displayed (Fig. 2(a)). The word is simultaneously set as an anchor. He/she can add words to the search query in order to narrow down the search focus (The anchor is intact). By clicking on the "Search" button, the search query is sent to an existing search engine via WebCOBALT.
(2) Search result receipt
He/she receives the search results modified by WebCOBALT. Generally, search engines return a page title, URL, summary text, and statistic as elements of search results. WebCOBALT adds comments written by learners to each result (Fig. 2(b)). The comments can be clues used to find a suitable page.
(3) Visit and free-hyperlink creation
By selecting a page, the page is presented together with a component for creating a free-hyperlink (Fig. 2(c)). The component is embedded in every page during exploration. The anchor is automatically set in the "Anchor" blank (He/she can modify the anchor in this blank). He/she writes a comment with three attributes, "informative", "neutral", and "useless". If the "informative" attribute is designated, the free-hyperlink between the anchor and the page is created immediately after clicking on the "Submit" button. Incidentally, this comment is also used in WebCOBALT.

2.3 How to Share Free-hyperlinks

The free-hyperlinks embedded are highlighted with a star-shaped icon in order for learners to distinguish the free-hyperlinks from original hyperlinks. By clicking on a free-hyperlink, a pop-up box containing the information about the free-hyperlink is immediately displayed. The information consists of a page title, URL, creator's e-mail address, creation date, and a comment. In the case where the same anchor has more than one free-hyperlink, these are displayed inside the same pop-up box. By clicking on a page title, the corresponding page is presented. A snapshot of free-hyperlinks, which are adapted using collaborative filtering, is shown in Fig. 4 (b). The next section describes the adaptation in MITS.

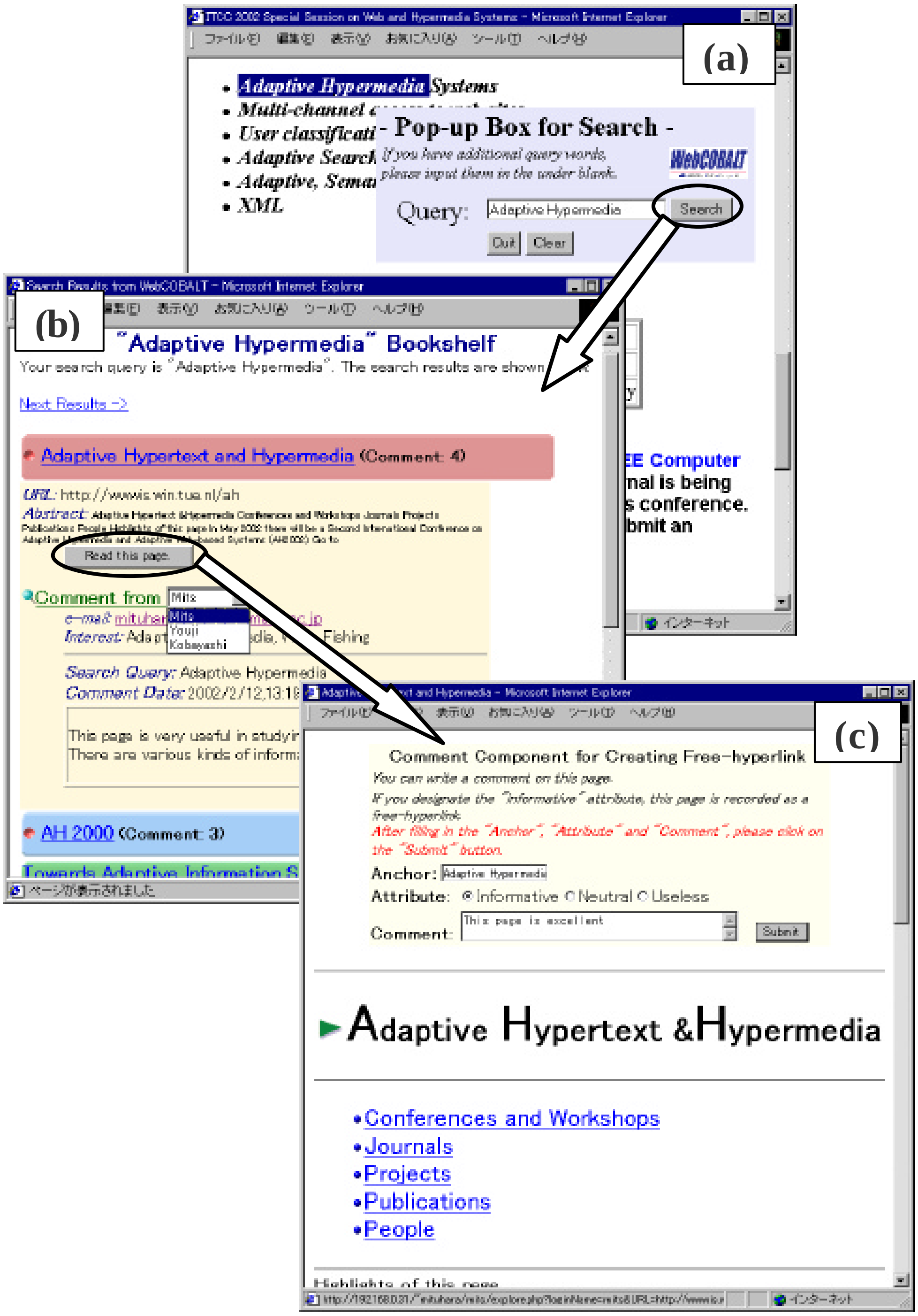


Fig. 2. The process of free-hyperlink creation using WebCOBALT

3 MITS

It is apparent that the increase of free-hyperlinks causes option overflow that interrupts the selection of adequate free-hyperlinks. In addition, meaningless free-hyperlinks are presented owing to a failure in text matching (e.g. to the anchor of "CD", the free-hyperlinks concerning "Compact Disc" and "Cash Dispenser" may be presented). MITS, which contains the free-hyperlink environment (Fig. 3), helps to solve these problems through adaptation by eliminating the free-hyperlinks on the basis of learners' interests.

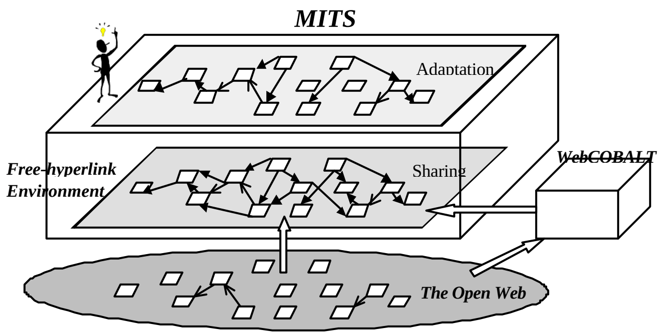


Fig. 3. The outline of MITS

3.1 Adaptation

The intuition behind the adaptation in MITS is as follows: learners will create free-hyperlinks for interesting pages rather than objective or unknown pages. Objective or unknown pages may become unnecessary at the point in time when learners complete reading the pages. However, learners' interests are preserved in the long term. This indicates that free-hyperlinks which connect interesting pages will be principally created. Therefore, free-hyperlinks represent learners' interests and the adaptation is performed on the basis of these interests

MITS uses collaborative filtering for adaptation. Collaborative filtering, which is normally used in recommender systems (e.g. GroupLens [12] and SurfLen [6]), is a technique for recommending items estimated to be informative by users with high similarity. On the other hand, it seems that collaborative filtering in MITS reduces option overflow and the failure in text matching by eliminating the free-hyperlinks created by learners with low similarity.

3.1.1 Similarity calculation

Calculating similarity between learners is based on the following assumption.

- *Learners who create the free-hyperlink with the same anchor have similar interests.*

MITS counts the number of free-hyperlinks with the same anchor from a free-hyperlink matrix and calculates the similarity by means of the Pearson's correlation coefficient (*r*). Next, it presents the free-hyperlinks of five learners with high similarity ($r>0.0$). Table 1 shows an example of the free-hyperlink matrix. The matrix consists of the "anchor " row and the "learner" column. Given two learners' free-hyperlink lists in the free-hyperlink matrix as $X=[x_1,x_2,\ldots x_n]$ and $A=[a_1,a_2,\ldots a_n]$, *r* is calculated from the following formula. Here *n* represents the entire number of anchors.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(a_i - \bar{a})}{\sqrt{\left(\sum_{i=1}^{n}(x_i - \bar{x})\right)^2 \left(\sum_{i=1}^{n}(a_i - \bar{a})\right)^2}}$$

However, collaborative filtering has a disadvantage: adaptation accuracy is low in the initial phase where estimation data are few (Naturally, no data indicates no adaptation). MITS is optimistic about this disadvantage, since the free-hyperlinks (i.e. estimation data) are actively created to simplify review activity (as described in 2.1).

Table 1. An example of free-hyperlink matrix

	X	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**
"Internet"	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1
"Dialup"	0	0	1	0	0	0	0	0	1	1	0	1	0	0	0	0
"TCP/IP"	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0
"ISDN"	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
"ADSL",	0	0	1	1	0	0	1	0	1	1	1	1	0	0	0	0
"Ethernet"	1	0	0	1	0	1	1	1	0	1	1	0	0	0	0	0
"firewall"	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1	1
"concrete"	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1
"building"	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

Note: "1" indicates that the free-hyperlink has been created.

3.1.2 Example of adaptation

In Table 1, there is learner X (target), who is interested in the computer network, and another fifteen learners, A ($r=-0.28$), B ($r=-0.37$), C ($r=0.47$), D ($r=-0.18$), E ($r=0.66$), F ($r=0.59$), G ($r=1.0$), H ($r=-1.0$), I ($r=0.05$), J ($r=0.18$), K ($r=-0.37$), L ($r=-0.18$), M ($r=-0.28$), N ($r=-0.37$), and O ($r=-0.47$). This example illustrates adaptation in the case where learner X visits a page shown in Fig. 4 (a), which has no original hyperlinks. The words of "Internet", "Dialup", "TCP/IP", "ISDN", "ADSL", "Ethernet", and "firewall" appear on this page, which correspond to anchors listed in Table 1. If there is no adaptation, free-hyperlinks concerning all these anchors would be presented.

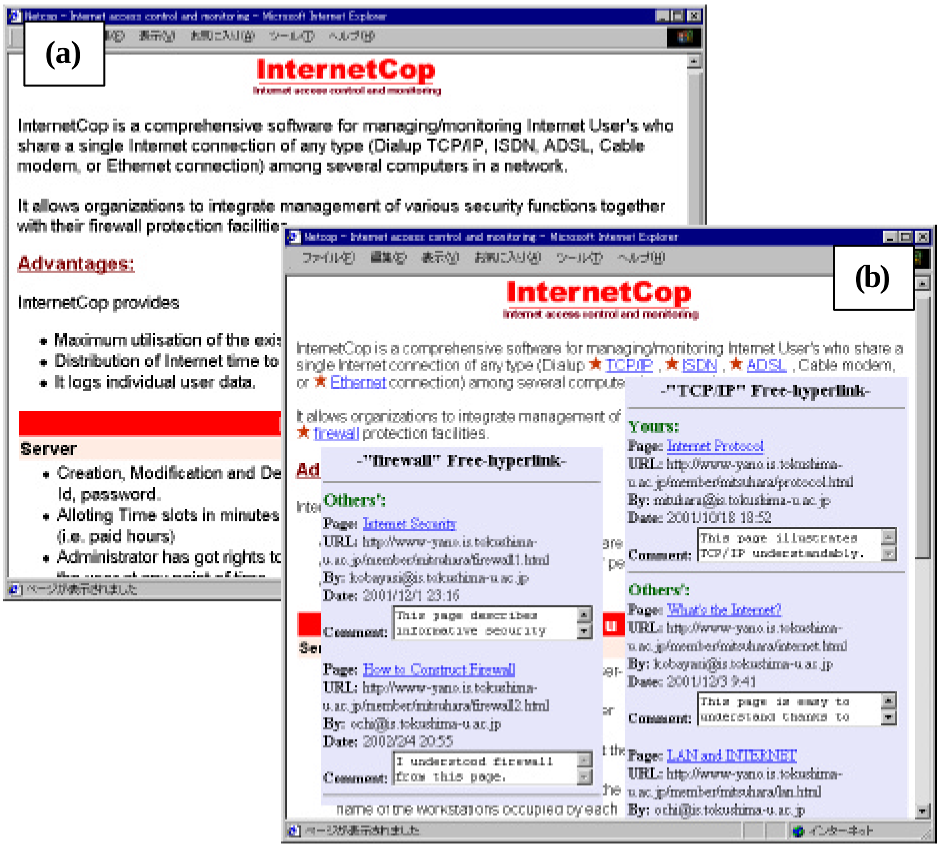


Fig.4. Adapted free-hyperlinks

MITS calculates the similarity and chooses five learners with high similarity, C, E, F, G, and J. In addition to the free-hyperlinks created by learner X ("TCP/IP" and "Ethernet"), the free-hyperlinks created by the similar learners ("TCP/IP ", "ISDN", "ADSL", "Ethernet", and "firewall") are embedded (Fig. 4(b)). In this way, option overflow is adaptively reduced. Furthermore, the failure in text matching is also reduced. Specifically, the free-hyperlinks created by learner M, N, and O are eliminated. They are interested in architecture or civil engineering and regard firewall as the meaning of fireproof wall and not Internet security. To eliminate their free-hyperlinks concerning firewall helps learner X to select adequate free-hyperlinks smoothly.

3.2 Implementation

MITS is a simple C/S system (Fig. 5). Basically, learners can use this system on a web browser without plug-in software. The following describes a server-side system composition.

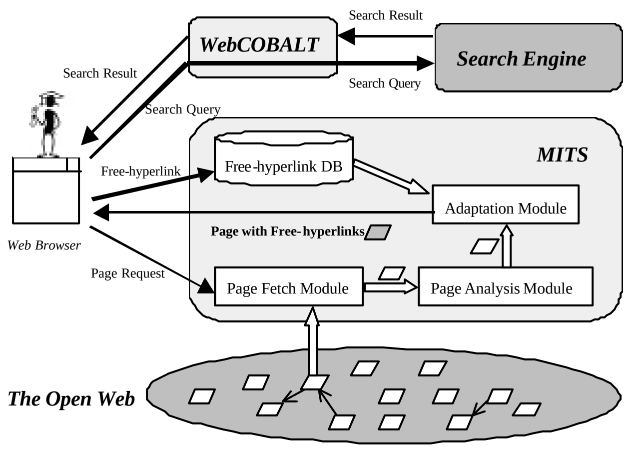


Fig. 5. System composition

Page fetch module: This module fetches a page requested by a learner.
Page analysis module: This module chooses free-hyperlinks to be embedded by means of keyword matching. In addition, it rewrites original hyperlinks on the page in order to always send the learner's request to MITS.
Adaptation module: First, this module embeds corresponding free-hyperlinks created by the learner. Next, it calculates the similarity and embeds corresponding free-hyperlinks created by similar learners. When the anchor of a free-hyperlink is involved in the anchors of original hyperlinks or other free-hyperlinks, this module adds a new anchor to embed the free-hyperlink beside the initial location.
Free-hyperlink database: The free-hyperlink matrix is stored.

4 Small-scale Experiment Through Trial Use

A small-scale experiment through trial use of MITS was conducted in order to evaluate its effectiveness. The specific evaluating points are the validity of the adaptation and the usability in free-hyperlink creation.

4.1 Method

The subjects of this experiment were graduate and undergraduate students who belonged to the department of computer science at Tokushima University. Twenty-six subjects were divided into the following two groups.

Group A (16 persons): Each subject created free-hyperlinks on a page describing 50 terms concerning computer science. This group concentrated on free-hyperlink creation.

Group B (10 persons): First, each subject created free-hyperlinks on a page describing 25 terms, which were chosen from the page given to group A. Secondly, he/she marked terms of interest from the 25 remainders. Thirdly, he/she read the page describing all the 50 terms, which were adapted on the basis of interests (the terms presumed to be interesting to him/her were presented as free-hyperlinks). This experiment requested him/her to refer to all free-hyperlinks (i.e. all pages that were connected with free-hyperlinks).

The usability in free-hyperlink creation was evaluated with questionnaires. The validity of the adaptation was evaluated with questionnaires. Then, the scores of recall and precision were calculated with the following formulas. The high score of recall indicates that the adaptation was performed without omission. The high score of precision indicates that the adaptation was performed without redundancy.

$$recall = \frac{n(T_m \cap T_f)}{n(T_m)}$$

$$precision = \frac{n(T_m \cap T_f)}{n(T_f)}$$

T_f: Terms presented by MITS as free-hyperlinks
T_m: Terms marked by a subject

4.2 Results and Considerations

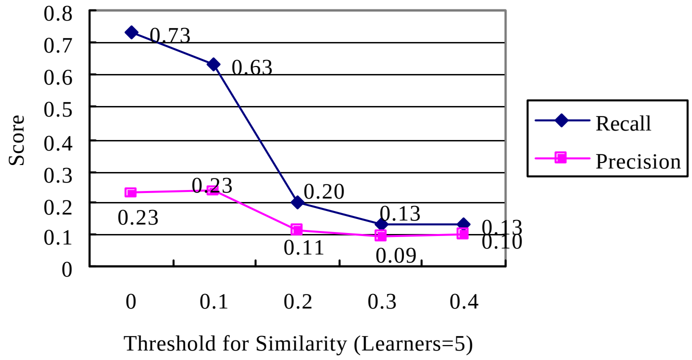
Through this experiment, group A created 158 free-hyperlinks to 44 terms and group B created 56 free-hyperlinks to 17 terms. Comments of 214 free-hyperlinks created were roughly classified into four traits (Table 2).

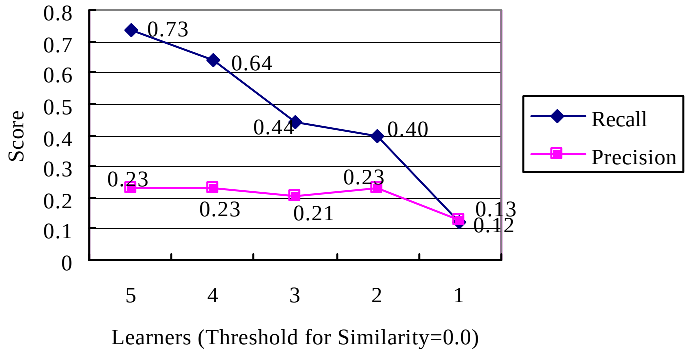
Table 2. The traits of comments

Trait	Example	Number
A mere summary	"This page describes JavaScript and prepares some sample programs"	93
A summary with personal opinions	"This page illustrates the concept of OOP understandably. I recommend this for novice learners!"	77
Personal opinions	"I think that DOM is very important in combining Java and XML"	27
Others	"This page is nicely designed"	17

(1) The validity of the adaptation

Figure 6 shows the mean scores of recall and precision. In the default values (*learners=5*, *threshold of similarity=0.0*) which choose learners with high similarity, the score of recall was 0.733 and the score of precision was 0.232. The score of recall is relatively high, whereas the score of precision is undoubtedly low. This indicates that many redundant free-hyperlinks were presented. Although these scores were





again calculated with the values lowered, the score of precision was not improved. This indicates that free-hyperlinks created by learners with the highest similarity may not necessarily fulfill a learner's interests. From these results, the default values were overall acceptable in this experiment. However, it is recognized that this experiment was too small-scale to prove the validity.

Table 3 shows the results of questionnaires which surveyed the validity. From the result for Q1, it was found that free-hyperlinks were principally created for interesting pages. This seems to correspond to the concept described in 3.1. Therefore, it may be true that free-hyperlinks represent learners' interests and the adaptation is performed on the basis of the interests. The result for Q2 indicates that free-hyperlinks presented actually tended to connect interesting pages. The mean value of the informativity of free-hyperlinks, which was subjectively estimated by group B, was 3.7 (Q3). In spite of the low precision, this value seems to be relatively high. This may indicate that originally redundant free-hyperlinks were changed to informative ones. It is possible that the subjects dug up new interests by following free-hyperlinks with comments. Judging from the above results, it can be concluded that the adaptation appears to function satisfactorily.

Opinions about free-hyperlinks were collected from group B. The relatively dominant opinions were as follows: "Efficiency of navigation was improved by following free-hyperlinks" and "Knowledge was broadened by following many free-hyperlinks". These opinions can be regarded as evidence that MITS avoids the learning impasse caused by inadequate hyperlinks. In addition, there were some interesting opinions as follows: "New learning goals were generated by others' comments", "There were new findings that familiar terms were used in unimaginable areas", and "To know learners with similar interests is useful in learning". These may indicate that MITS facilitate meta-cognitive activities and collaborative activities.

Although this experiment was small-scale (the terms were limited within the area of computer science, and the subjects and the free-hyperlinks were insufficient), the effectiveness of MITS was proven to a certain extent. In order to prove the effectiveness in more detail, there is a plan to open this system to the public and experiment on a large scale.

Table 3. The results of questionnaires to survey the validity

Question	Learners
Q1: What pages did you create free-hyperlinks for principally? (This question was given to both groups and was a multiple choice question)	21 (Interesting)
	4 (Unknown)
	2 (Not understood)
	3 (Objective)
	6 (Other)
Q2: What pages did free-hyperlinks presented connect principally? (This question was given to only group B and was a multiple choice question)	6 (Interesting)
	1 (Unknown)
	3 (Not understood)
	1 (Objective)
	0 (Other)
Q3: Were the free-hyperlinks presented informative for your learning? (This question was given to group B only. They were requested to choose from one to five degrees)	0 (5: Definitely yes)
	8 (4:Yes)
	1 (3: Intermediate)
	1 (2:No)
	0 (1: Definitely no)
	Mean: 3.7

(2) The usability in the free-hyperlink creation

Table 4 shows the results of questionnaires which surveyed the usability. The mean value of the usability in designating the search query (i.e. the usability of the pop-up box for search) was 3.65 (Q4). Affirmative opinions were as follows: "Not visiting a search engine was timesaving" and "Not typing words or doing copy&paste was timesaving". Negative opinions were as follows: "It was burdensome that the pop-up box was frequently displayed by unrelated mouse clicking" and "It was not useful due to elimination of some functions implemented in common search engines". The mean value of the usability in writing a comment (i.e. the usability of the comment component) was 3.38 (Q5). From negative opinions, it was found that this usability becomes lower especially in a page that consists of a few frames, since the comment components occupy small frames. The mean value of the willingness to write comments, which means the overall usability, was 3.15 (Q6). This low value probably results from the inconveniences shown above. These inconveniences ought to be quickly improved since the usability is strongly related to the increase of free-hyperlinks.

Table 4. The results of questionnaires to survey the usability

Question	Learners
Q4: Was the pop-up box easy to use when you wanted to designate the search query?	7 (5: Definitely yes)
	11 (4:Yes)
	2 (3: Intermediate)
	4 (2:No)
	2 (1: Definitely no)
	Mean: 3.65
Q5: Was the comment component easy to use when you wanted to write a comment?	3 (5: Definitely yes)
	10 (4:Yes)
	7 (3: Intermediate)
	6 (2:No)
	0 (1: Definitely no)
	Mean: 3.38
Q6: Will you willingly write comments during your exploration?	1 (5: Definitely yes)
	10 (4:Yes)
	10 (3: Intermediate)
	2 (2:No)
	3 (1: Definitely no)
	Mean: 3.15

Note: These questions were given to both groups who were requested to choose from one to five degrees.

5 Conclusions

This paper has introduced an adaptive web-based learning system with a free-hyperlink environment, which helps to avoid a learning impasse by enabling learners to create hyperlinks in the open web and share the hyperlinks adapted on the basis of individual learners' interests. As learners are inclined to construct knowledge through exploring the open web autonomously, such a system will be increasingly required.

The prototype system includes a few problems. One problem that must be solved is a failure to modify original web pages. For example, JavaScript embedded by MITS contends with JavaScript in the original web pages. Although the original JavaScript is currently overwritten to avoid this situation, this may reduce learning effects provided by original web pages. In particular, to invalidate JavaScript controlling hyperlinks seriously interrupts learners' exploration. Another problem is how to deal with unavailable free-hyperlinks in the present situation where numerous web pages are removed daily. Learners expect that informative pages will always be presented even if the pages have already been removed. To fulfill this expectation, it is very important to generate the next version of this system. In addition, system performance has to be considered for practical use. This research is trying to solve these problems and is preparing large-scale experiments.

Acknowledgement

This research was supported in part by the grant-in-aid for Scientific Research (B)(2) No.13480047 from Japan Society for the Promotion of Science.
This research was supported by a grant to RCAST at Doshisha University from the Ministry of Education, Science, Sports and Culture, Japan.

References

1. Bonder, R., Chignell, M., and Tam, J.: Website Authoring using Dynamic Hypertext, *Proc. of WebNet'97*, pp.59-64 (1997).
2. Boticario, J. G., Gaudioso, E., and Hernandez, F.: Adaptive Navigation Support and Adaptive Collaboration Support in WebDL, *Proc. of Adaptive Hypermedia 2000*, pp.51-61 (2000).
3. Brusilovsky, P., Schwarz, E., and Weber, G.: ELM-ART: An intelligent tutoring system on World Wide Web, *Third International Conference on Intelligent Tutoring Systems, ITS-96*, pp. 261-269 (1996).
4. Brusilovsky, P.: Adaptive Hypermedia, *User Models and User-Adapted Interaction 11*, pp.87-110 (2001).
5. De Bra, P. and Calvi, L.: AHA! An open Adaptive Hypermedia Architecture, *The New Review of Hypermedia and Multimedia, vol. 4*, pp.115-139 (1998).
6. Fu, X., Budzik, J., and Hammond, K.J.: Mining Navigation History for Recommendation, *Proc. of International Conference on Intelligent User Interface*, pp.106-112 (2000).
7. Henze, N. and Nejdl, W.: Extensible Adaptive Hypermedia Courseware: Integrating Different Courses and Web Material, *Proc. of Adaptive Hypermedia 2000*, pp.109-120 (2000).
8. Mitsuhara, H., Ochi, Y., and Yano, Y.: Open-ended Adaptive System for Facilitating Knowledge Construction in Web-based Exploratory Learning, *Proc. of Adaptive Hypermedia 2002* (2002).
9. Mitsuhara, H., Ochi, Y., and Yano, Y.: A Web Retrieval Support System with a Comment Sharing Environment: Toward an Adaptive Web-based IR System, *Position Paper of WASWBE'02* (2002).
10. Miyake, N. and Masukawa, H.: Relation-Making to Sense-Making: Supporting College Students' Constructive Understanding with an Enriched Collaborative Note-Sharing System, *Proc. of 4th International Conference of the Learning Science*, pp.41-47 (2000).
11. Ogata, H., Imai, K., Matsuura, K., and Yano, Y.: Knowledge Awareness Map for Open-ended and Collaborative Learning on the World Wide Web, *Proc. of ICCE'99*, pp.319-326 (1999).
12. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *Proc. of CSCW94*, pp.175-186 (1994).
13. Tauscher, L. and Greenberg, S.: How people revisit web pages: empirical findings and implications for the design of history system, *International Journal of Human-Computer Studies, Vol.47, No.1*, pp.97-137 (1997).

Ownership Transfer via Personalisation as a Value-adding Strategy for Web-based Education

James Ohene-Djan

Department of Mathematical & Computing Sciences
Goldsmiths College, University of London
New Cross, London SE14 6NW, UK
Tel: 0044 207 919 7171
Fax: 0044 207 919 7171
email:j.ohene-djan@gold.ac.uk

Abstract. This paper focuses on *ownership transfer* via personalisation as a value-adding strategy for web-based education. It contributes an approach to endowing hypermedia systems (e.g., for learning) with features that enable the personalisation of the interactions between users and the system which embodies the learning material. A model for the personalisation of hyperdocuments is introduced that aims to make available to users of, e.g. electronic learning materials, features that allow them to interact with these documents in a manner much closer to that of owners of paper-based ones, while still benefitting from the ability to traverse contextual links and to exploit the computational environment in which the material is embedded.

1 Introduction

The potential for harnessing hypermedia technology to address issues of learning has long been recognised [16, 8]. It has gathered new momentum with the emerging prominence of web-based education systems accessible via the Internet [9].

As noted in [16], the principal attraction of hypermedia is that it lends itself naturally to non-sequential educational approaches since it encourages the free association of the characteristics of human thought. Although researchers have delivered a variety of impressive systems [11, 21, 5] many can be criticised for the ad-hoc, technology-driven approach often taken towards their engineering and development [19]. The consequences of this ad-hoc approach have been the relatively low levels of interchangeability and of interoperability between systems. Such inefficiency has caused, as noted in [22], "the history of educational technologies [to be] littered with pockets of expertise, government pronouncements, and investment in equipment that has quickly ended up gathering dust in forgotten corners".

Personalisable, adaptive hypermedia systems, more often referred to as *adaptive hypermedia*, (AH) [10, 9] constitute an area of research that aims to enhance the functionality of hypermedia systems by making the user interaction process personalisable. The approach taken is to endow systems with personalisation features which may be initiated by the users or by the system itself. AH are assumed to be useful in areas, such as learning, where users have different information seeking goals, histories and preferences. AH aim to use knowledge provided by (or captured about) specific users to tailor the information and the links presented to each specific user. By applying the knowledge of users they amass, AH can support users in navigation by limiting the options for traversal to information units, suggesting relevant links to follow and providing additional information on links and information units.

In this paper we focus on *ownership transfer* via personalisation as a value-adding strategy for web-based educational systems. We report on a system-independent approach for the personalisation of learning materials deployed on the web whose complete formalisation can be found in [23]. The approach facilitates the personalisation of the interaction process between users and the learning materials, thereby enabling individual users to come closer to satisfying their specific, dynamic information needs.

The rest of the paper is structured as follows: To motivate the contributions of the paper, Section 2 contrasts what actions owners of paper-based materials can perform that users of hyperdocuments cannot, and vice-versa; Section 3 introduces the model contributed by the paper; Section 4 uses an example to show how the model proposed allows the desiderata identified to be met; Section 5 compares the results with those obtained by other researchers; Section 6 draws conclusions.

2 Motivation

This section discusses the motivation for the idea of ownership transfer, via personalisation, as a value-adding strategy by contrasting the kind of personalisation actions that owners of traditional learning materials (such as printed books) and users of HLBSs can carry out.

Assume that a user is a student at a university studying a degree in Computer Science. As part of this degree, the student is required to study a course in Internet Technologies and in particular the use of the Internet for commercial activities. The course book is *Electronic Commerce* by G Schneider and J Perry [24]. A small section of the book is depicted in Figure 3.

Consider the question as to how students interact with a textbook over extended periods of learning. Furthermore, consider interactions which would be difficult or impossible if the book were a hyperdocument. Some of the personalisation actions that fall into this category include:

- *Selective Reading*: To open and read some page(s) only (possibly consulting the table of contents or index) and perhaps only partially. In current hyperdocuments, options for traversal are defined by the links the designer of the hyperdocument has embedded in it. Users are not provided with facilities

to make links between pages and therefore define their own paths through the material. Also, when a user does want to traverse a link defined by the designer there is no metadata about the target thereby making it impossible not to commit to the traversal. Since traversing links can incur high computational costs, frustration sometimes ensues;
- *Annotation*: To add to the original printed text handwritten material that, for the student, enriches the meaning of the text. Examples include underlining or highlighting a passage, writing comments in the margin, cross-referring to some other page of the book or some other information resource (e.g., a website, another book, etc.). At present, to the authors knowledge, no hyperlink-based learning system deployed on the WWW provides features to allow users to annotate parts of a hyperdocument in context, i.e., at a particular point in the text;
- *Content Insertion and Deletion*: To add content (e.g. copied from another book, or a website) at specific points and to delete (or cross out, not necessarily physically) parts of the text. Features to enable the users of a hyperdocument to tailor its content have also not been implemented widely. In particular, users are not given the ability to remove parts of a hyperdocument which they deem inappropriate for study at a point in time;
- *Bookmarking*: To leave markers at particular pages (e.g., to indicate that they are more important in some respect than others). Note that marked pages are easily identifiable for selective reading; Although WWW browsers allow users of hyperlink-based learning systems to bookmark pages, such bookmarks are often limited to recording only the Internet address (location) of a page. Users are not given the opportunity of indicating the importance of some bookmarked page over another or why a particular page is bookmarked at all.

Each of the above actions are admissible on paper-based materials that the student owns but markedly difficult on hyperdocuments because the student has no means of claiming ownership of the material nor of personalising the latter (even if ownership could be transferred from the designer to the student, because the student may not have the designer knowledge required for personalisation to be carried out).

Now consider the question as to how students would interact with a textbook implemented as a hyperdocument. Furthermore, consider interactions which would be difficult or impossible if the book were *not* a hyperdocument. Some actions that fall into this category include:

- *Context-Based, Nonlinear Navigation*: To benefit from the availability of contextual links at points deemed appropriate by the designer of the hyperdocument and traverse the material nonlinearly;
- *Dynamic Information Management*: To make use of computational resources to react and respond to requests dynamically (e.g., to benefit from a web of information comprising many, possibly remote, independently generated and maintained interrelated hyperdocuments).

Each of the above actions are admissible on electronic materials that the designer prepares, but markedly difficult on paper-based documents even if the student owns them.

In summary, although users of hyperdocuments benefit from the concept of linking and from the fact that electronic media typically exist in a computing environment, there are many useful actions that the owner of paper-based documents can take that are not open to users of hyperdocuments. Transfer of ownership via personalisation actions seems, therefore, to be a central issue in deploying web-based systems in learning contexts.

3 The Model

This section introduces a model for the personalisation of hyperdocuments that aims to make available to users of hyperdocuments such as electronic learning materials, features that allow these users to interact with the hyperdocuments in a manner much closer to that of owners of paper-based ones, while still benefitting from the ability to traverse contextual links and to exploit the computational environment in which the material is embedded.

A general, open architecture for web-based systems, of the kind depicted in Figure 1 as a simplified data flow diagram, is assumed. Here we propose one view of the interior of the shaded oval in Figure 1. It is assumed that hyperlink functionality is provided as a client technology dependent on loose couplings to (at least) a user-interface server (UIS) and a database server (DBS). The classical example of a UIS is a WWW browser (e.g. Netscape Communicator). Among other functions, browsers broker requests and have rendering capabilities. Browsers can render formal texts (e.g., rendering expressions authored in HTML or XML). Examples of DBSs are database management systems (DBMSs) which support client server architectures (e.g., Oracle, or Informix).

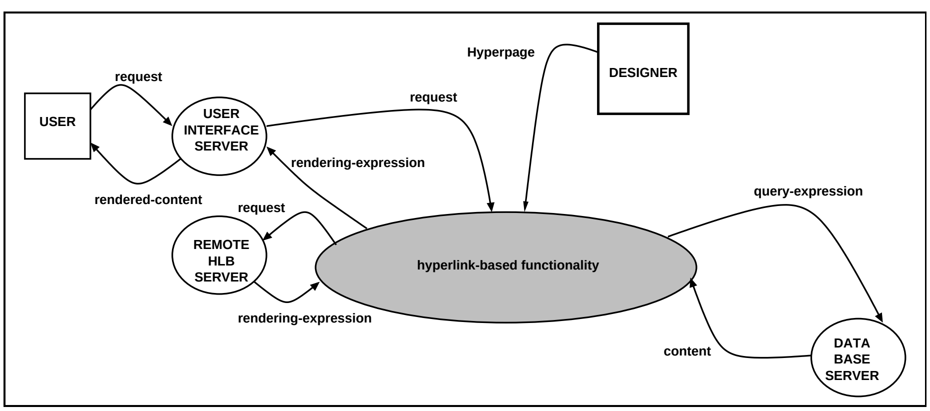


Fig. 1. An Architecture for web-based hypermedia systems

Broadly, the dynamics associated with Figure 1 are as follows. The UISs capture requests for desired hyperpages. The UISs channel requests for hyperpages into the hypermedia system proper. If a request is for a hyperpage which resides in a remote hypermedia system, then the core of hyperlink functionality interacts with it to obtain the requested hyperpage in the form of a rendering expression that the core can pass back for the UISs to render. If the request is for a local hyperpage (e.g. one which is known to the core) then the latter responds by returning a rendering expression to the UISs, possibly after querying one or more DBSs to fetch some or all of the content specified for the requested hyperpage.

Implicit in Figure 1 is the assumption that personalisation actions in HLBSs should not, and need not, be compounded with personalisation actions that might be provided by user-interface and database components in HLBS architectures. The shaded oval in Figure 1 is responsible for what users experience as hyperlink-based information retrieval. Notwithstanding the fact that users may want to personalise database and user-interface features, clearly it can be argued that whatever is in the scope for personalisation actions resides in the shaded oval.

To model adaptive, personalisable hyperlink-based interaction, a model is proposed in which the shaded oval in Figure 1 is partitioned into regions. Non-adaptive, non-personalisable hypermedia systems are modelled by a group of functions referred to as the *H-region*. Personalisable hypermedia systems require the addition to the H-region of the functions provided by the *P-region*. Adaptivity, seen as user-specific, system-initiated personalisation, comprises another region of functionality referred to as the A-region. (see [23] for complete architectural specificiation and formal semantics).

3.1 H-Region: Core Hyperlink-Based Functionality

The H-region models a core of hyperlink-based functionality. Conceptually, the H-regions behaves as a *composer of hyperdocuments from specifications*, i.e. what the designer of a hyperpage designs is not a hyperpage, but rather a specification of how to build the hyperpage upon request. Hyperpages are modelled as formal specifications and a formal language has been defined for this purpose [23].

The functionality of the H-region is depicted in Figure 2.

Within the H-region, users can only request for hyperpages to be rendered. The decisions that the designers of a hyperdocument have made with respect to content, navigation and rendering cannot therefore be overridden. Upon a request arriving from a UIS, a composition function parses the hyperpage specification into a series of actions that, when executed, convert the specification into renderable text that is sent to the UIS as the response to the original request.

The semantics of a hyperpage specification (of which an example is shown in Figure 4) have been formalised as a program which, when interpreted, typically fetches content from a DBS, composes the content into a renderable text (making use of template variables as a binding mechanism) and finally responds to the original request with renderable text. Figure 5 indicates how the hyperpage specified in Figure 4 might be rendered.

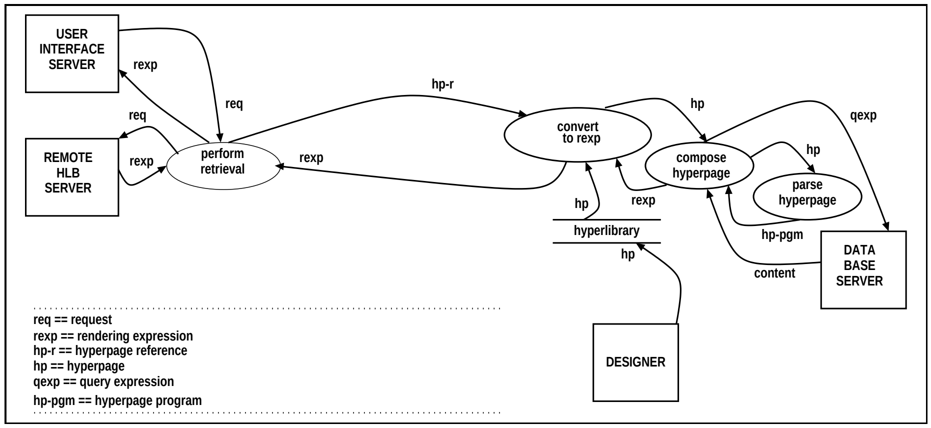


Fig. 2. The H-Region

3.2 P-Region: Personalisable Hyperlink-Based Interaction

The P-region comprises a group of functions that are non-disruptively added to the H-region in order to model personalisable hyperlink-based interaction. Personalisation is viewed as the process of handing over to the user the ability to *annotate specifications or rewrite them or both,* thereby allowing the user to override, in principle, each and every designer specification. Therefore, within the P-region users can not only request a hyperpage, but also annotate or rewrite it, thereby creating their own version of it. The decisions that the designers of that hyperpage have made with regard to content, navigation and rendering of the hyperpage can therefore be overridden by users and this kind of event characterises ownership transfer.

When personalisation functionality is layered over the core, a designer can annotate a hyperpage in preparation for differences in users' goals and histories. A user can personalise not only such annotations (of which an example is shown in Figure 6), but the hyperpage specifications as well. Personalisation requests (of which examples are shown in Figure 7) allow users to specify which hyperpages are to be personalised and how they should be transformed.

The kinds of personalisation actions modelled are based on annotating and rewriting the hyperpage specifications. Annotation pairs a hyperpage specification with notes of interest to the user and, by doing so, presumes that versioning takes place. Such notes take one of the following forms. Firstly, a note can assign user-specific values to user-generic attributes of interest (e.g., that the level of difficulty of a given page or component part is high, or that 'Internetworking' is a keyword of relevance to a given page). Secondly, a note can specify a rewriting action over the renderable text after it has been composed by the H-region, i.e. after content has been fetched and made ready for display (e.g. to map American into British spelling forms). This form of post-composition rewriting can also be conditional on the environment (e.g. replace images with captions if the display unit is text-only).

The existence of annotations on hyperpages allows for: personalisation of a specified hyperpage; the specification of *alternatives* to a specified hyperpage; the specification of *comparable* hyperpages to a specified one and the recording of information about a hyperpage (i.e. what are the current values of attributes set by previous annotations).

Annotations are not operations on hyperpage specifications, i.e., they do not alter the latter. Rather, they give rise to a user-specific pairing with a hyperpage specification. Rewriting also causes versioning and can be characterised simply as the editing of hyperpage specifications. If a user edits the hyperpage specification as conceived by its designers, ownership is thereby transferred. If, subsequently, the same user edits that hyperpage specification again, adjustment takes place.

A formal language for annotations has been defined, as has a formal language for the personalisation requests used to maintain annotations and hyperpage specifications [23]. Personalisation requests allow users to take the following actions to enhance their interaction process:

1. *Content Tailoring*: The insertion or deletion of hyperpages and, recursively, their page parts;
2. *Selective Content*: The hiding from users of parts of the content of hyperpages which they wish not to see;
3. *Content Rewriting*: The rewriting of specified phrases in hyperpages;
4. *Prerequisite Content Selection*: The fetching and displaying of prerequisite content for a particular hyperpage or page part;
5. *Linking*: The insertion of hyperlinks between hyperpages and their page parts;
6. *Link Annotation*: The annotation of links made between hyperpages and their page parts;
7. *Link Hiding and Blocking*: The hiding or blocking of links on a hyperpage.

3.3 A-Region: System-Initiated Personalisable Hyperlink-Based Interaction

The A-region details how system-initiated personalisation, namely adaptivity, may be modelled as a coherent and consistent extension of the H- and P-regions. Within the A-region adaptivity is viewed as the process of *allowing the system, by proxy, to take the initiative in the personalisation actions.* When adaptive functionality is layered over the H- and P-regions, both users and designers can define strategies as to when the system should take the initiative and actively tailor a user's interaction in the light of that user's information goals and history of use.

In our view, system-initiated personalisation is, in principle, as expressive as user-initiated personalisation and requires no technologies other than those involved in user modelling and in decision making from a user model.

An information goal characterises what a user wishes to achieve in terms of information gathering. Broadly, information goals specify how the information in the hypernetwork should be tailored for a user. A user's history characterises what is already known by the user about a hyperdocument and about the system itself. This knowledge is based on a record of previous interactions with hyperdocuments. Together, a user's information goals and history may be viewed as a user model.

In the context of this paper, a decision theory may be understood to be a general model of what is best to do and when. Central to the decision theory is a set of rules, each of which is an imperative command that is meant, in principle, to effect a state transition.

The approach taken to system-initiated personalisation centres on an adaptation function. This function implements an inference engine over a decision theory, specified as a set of active rules [13], that describes which actions are more likely to yield the most benefits given some accumulated knowledge of past interactions. The actions which the inference engine is in charge of suggesting are personalisation actions as defined in the P-region.

The adaptive process starts when an event is detected (e.g., a request for a hyperpage). As a result of the detection of an event, the adaptive function consults a user's information goals to determine which entries have been made.

This information is then used to construct and then suggest to the user a well formed personalisation request that is aimed at making user interaction more personal. If this suggested personalisation request is accepted by the user then it is submitted to, and acted upon, by the functions that comprise the P-Region.

In summary, the addition of the A-region allows the HLBS to take the initiative and launch personalisation requests on behalf of a user. For this to be possible, the adaptive function draws upon a model of a user's goals (both explicitly stated and inferred), a history of a user's requests and a decision theory. Such a decision theory will embody strategies towards achieving information goals. See [23] for details.

4 Personalisation at Work

This section uses an example to show how the desiderata identified in Section 2 are met by the functionality arising from the model described in Section 3. Consider Figure 3 where an excerpt from [24] is transcribed.

To specify a hyperpage to correspond to Figure 3, the approach taken is to divide a page into a sequence of segments called *chunks*. Each chunk comprises a specification of its content (i.e. either hardwired text or data returned as the result of queries sent to databases, figures, etc.) and a specification of how this content should be rendered for presentation (e.g. via HTML tags). A chunk may contain template variables. Conceptually, a template variable is associated with bits of text or else acts as a place holder for content which becomes available after evaluation of the query associated with it. The rendering part is specified in a language which the UIS can render, except that this renderable text may be

IMAP, the Internet Message Access Protocol, is a newer protocol that has advantages over POP and that may, one day, replace POP. IMAP, like POP, can download e-mail, delete mail from the server, or simply ask if there is new mail. IMAP, however, improves upon some of the shortcomings of POP. For instance, it defines how a client program asks a mail server to present available mail. IMAP can request that the server download only selected e-mail messages rather than all messages. Your e-mail client can view just the heading and the name of the sender of the e-mail message and then decide whether to download the message. IMAP, through your client e-mail program, allows you to create and manipulate mail folders or mailboxes on the server, delete messages, and search for certain parts of a note or an entire note – all without down-loading mail from the server to your PC.

Fig. 3. An Excerpt from [24]

interspersed with template variables. After content is retrieved and the template variables are assigned, textual replacement takes place substituting values (i.e. retrieved content) for references (i.e. occurrences of template variables). The overall result is renderable text (e.g. HTML).

In addition, a chunk may be associated with two sets of identifiers. The first set is the set of entry points to the chunk and the second is the set of its exit points. An entry point enables the chunk (as an anchor in the page) to be referenced in a request. An exit point enables a chunk to establish navigable links to the anchor denoted by the exit point. In WWW parlance, an entry point can be thought of as a *URL* and an exit point as a *hyperlink*.

Figures 4 shows one way in which Figure 3 might be written as a hyperpage specification. It is assumed that database queries for retrieving content are written in SQL and that HTML is used as a mark-up language in the construction of a renderable text. For more detail on how the evaluation of Figure 3 into renderable text takes place, see [23]. Figure 5 depicts the hyperpage after it has been rendered by a UIS (in this case, an HTML browser).

4.1 Hyperpage Annotations

A hyperpage annotation pairs a hyperpage specification with notes of interest. An annotation is, roughly, a collection of contextualised attribute-value pairs. An example of an annotation for the text specified in Figures 4 is given in Figure 6. The attributes used are *description*, which lets the student or designer provide a summary or abstract; *keyword*, which lets the student or designer tag content and thereby clarify the information contained; *level*, which lets the student or designer attach a measure (e.g. of difficulty) to the text; and *see-as-well*, which lets the student or designer specify hyperpages that describe comparable information. Again, [23] provides more details.

```
page{
 chunk{
  entry{
    [<a name="Email Message Protocol"></a>]  }
  content{
    A := 'IMAP',
    B := ', the',
    C := 'Internet Message Access Protocol',
    D := 'is a newer protocol that has advantages over',
    E := 'POP',
    F := 'and that may, one day, replace POP. IMAP,
         like POP, can download e-mail, delete mail
         from the server, or simply ask if there is
         new mail. IMAP, however, improves upon some
         of the shortcomings of POP. For instance,
         it defines how a',
    G := 'client program',
    H := 'asks a',
    I := 'mail server',
    J := 'to present available mail. IMAP can request
         that the server download only selected  e-mail
         messages rather than all messages. Your e-mail
         client can view just the heading and the name
         of the sender of the e-mail message and then
         decide whether to download the message.  }
  rendering{
    [<b><a href="imap.html">          A </a></b>] B
    [<b><a href="imap.html">          C </a></b>] D
    [<b><a href="pop.html">           E </a></b>] F
    [   <a href="clientprogram.html"> G </a>    ] H
    [   <a href="mailserver.html">    I </a>    ] J  } }
 chunk{
  entry{
    [<a name="IMAP"></a>]  }
  content{
    A := 'select definition
          from   glossary
          where  term = 'IMAP'  }
  rendering{
    A  } } }
```

Fig. 4. Fig. 3 as a Hyperpage Specification

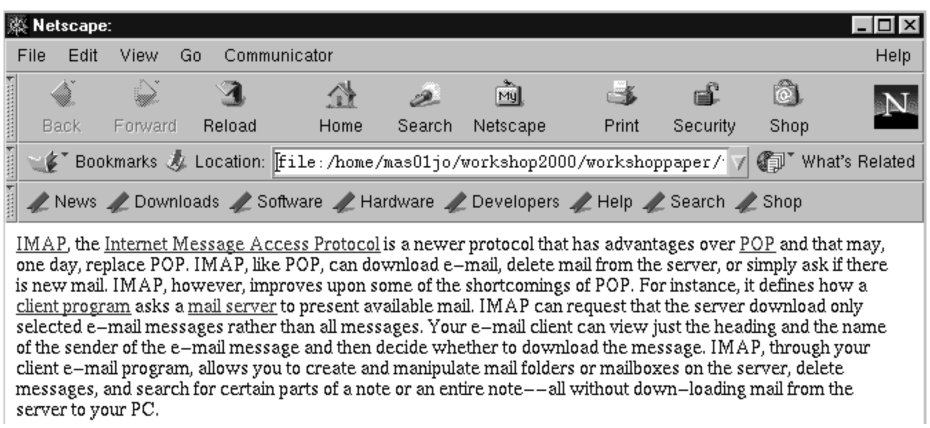


Fig. 5. Possible Rendering of Fig. 4

The annotation in Figure 6 will be associated with the hyperpage specification in Figure 4. Conceptually, this versions the page specifically for the annotator.

```
annotation{
 page:
  description := 'defines and contrasts IMAP with
                 other Internet protocols for
                 e-mail services';
  level       := 1;
  see-as-well := [http://www.internetmail.com/];
  keyword     := 'e-mail', 'electronic mail';
 [1, chunk]:
  description := 'Internet Message Access Protocol'
  keyword     := 'IMAP', 'POP',
                 'Internet Message Access Protocol';
  see-as-well := '[page-60.html]'
 [2, chunk]:
  description := 'Definition of IMAP
                  retrieved as dynamic content';'
 [2, chunk (R-spec)]:
  keyword     := 'Rendered using HTML'; }
```

Fig. 6. Annotating Fig. 4

4.2 Personalisation Requests

Personalisation requests allow a user (e.g. a student) to generate annotations, to update annotation (e.g. those provided at source by the designer), and to update (by versioning) the hyperpage specifications themselves. A few examples of personalisation requests (in a language, and with a semantics, formally defined in [23] that a student might issue (to the hyperdocument of which Figure 4 is a page) are depicted in Figure 7.

Example 1 in Figure 7 is a request to tailor the content found in the book. The request applies to all hyperpages (because the selection condition is vacuously `true`). Its effect is to insert the string, `"These pages belong to Student X"` into the rendering specification of the first chunk of each page. Example 2 is a request to add content. The request is applied to all hyperpages that contain the string "electronic mail" in its 5th chunk. The effect on each selected hyperpage is the insertion of a new 1st chunk. Example 3 in Figure 7 depicts a request to rewrite a specified string found in the hyperdocument. The request applies to all hyperpages and has the effect of inserting into the annotation of each hyperpage a request (as denoted by the arrow '`->`') that all occurrences of the string `"http://www.ecommercebook.com"` in the composed text be rewritten as `"http://www.ecommercebooknewversion.com"` in the the rendering text.

The example in this section, in spite of space limitations, illustrates how hyperpage specifications, that can be annotated and maintained via personalisation requests, enable users of hyperdocuments to claim ownership by actions that version pages by means of annotations and rewriting requests.

```
select-page-if            % Example 1
  true
hp-then-do {
  insert [1, chunk (R-spec)]
     "These pages belong to student X"  }
                            % ----------
select-page-if            % Example 2
  [5, chunk] contains "electronic mail"
hp-then-do {
   insert [1, chunk]
     chunk {
       content  { X := 'Many of the latest
                       electronic mail systems
                       now provide support for
                       sound and video files.'  }
       rendering { [<I>] X [</I>]} } }
                            % ---------
select-page-if            % Example 3
  true
ann-then-do {
  insert page :
       "http://www.ebook.com"
    -> "http://www.ecommercebook.com"; }
```

Fig. 7. Personalising Fig. 4

5 Comparison

This section compares and contrasts the proposed model with related work on modelling and implementing personalisation in hypermedia systems.

Little research has been conducted into formalising personalisation in web-based educational systems themselves. Most work has concentrated on providing personalisa tion features via data querying mechanisms (e.g., [1, 3, 14]). Although these contributions show how personalisation may be experienced, it can be argued that they merely exploit DBMS functionality rather than accounting for how hyperlink-based interaction itself can be tailored in a principled manner.

Since we have formalised (in [23]) a complete space of possibilities for personalisation actions, the adoption of our model enables a system to support all the personalisation actions described in [8] and many of those in [9]. Implemented systems such as Adaptive HyperMan [20], ELM-ART [11], Hypadapter [15] and 2L670 [5] can all be represented using our model. Moreover, while in our model every design decision can be overridden no proposal in the literature is as expressive as ours. In this way, we provide a more complete and coherent account of how personalisation enables and fully characterises ownership transfer.

In its aims, our model subsumes the work on the AH reference model AHAM [12]. However, the explicit aim of the AHAM is to represent techniques that have been implemented and hence serve as a primary reference for comparative studies. In contrast, the work reported here aims to induce a set of personalisation actions from a formal definition of a core of hyperlink functionality. Rather than simply formalising implemented examples of personalisation actions, we have induced a complete space of possibilities for personalisation actions. In this way, we provide a methodological approach to the area that goes beyond the effort embodied in the AHAM.

Work on XML [6] bears some resemblance with ours. Both approaches seek to divide a hyperdocument into units which either specify data or delineate regions where data may be obtained. The use in XML of arbitrary-value pairs associated with a document's structure, can be contrasted with our notion of note. The functionality of an XML-based web-based education system is not expected to be dissimilar from that of the H-region. However, we propose an abstract model of hypermedia-based interaction and not just of specification. Also, the specification elements in our model may, without loss of expressiveness or generality, be recast in SGML/XML.

6 Contributions and Conclusions

The challenge which the research [23] underlying this paper aims to meet is how to model, at a suitable level of abstraction, the space of possibilities for personalisation actions that could be made available to users of hyperlink-based learning systems.

To respond to this challenge, we have adopted the notion of ownership transfer as a means of providing a user with a learning experience that approximates that of printed materials while preserving and even building on the benefits associated with hypermedia technology. To enable ownership transfer, we have focussed on personalisation actions. To make it possible for personalisation actions to become part of web-based educational systems we have proposed a model based on hyperpages as specifications of dynamic content and rendering, preferences as hyperpage versioning by annotation, and personalisation requests as the events that cause that versioning to take place.

In Section 4 we have exemplified how our model is a step towards bringing together the complementary benefits, highlighted in Section 2, in paper materials owned by a student and hyperdocuments that the student can only use but not personalise. The model constitutes, therefore, a contribution to the goal of making web-based educational systems able to address users' information needs on an individualised basis, as discussed, e.g. in [4, 7]. Our model can be concretely used as a basis for the implementation of hypermedia systems (one instance of which is described in [23]) as well as for the study of implemented systems such as [2, 17, 18, 21, 5] and others.

References

1. Foto Afrati and Constantinos D. Koutras. A Hypertext Model Supporting Query Mechanisms. In *Proceedings of the ECHT'90 European Conference on Hypertext*, pages 52–66, 1990.
2. I Beaumont. User Modeling in the Interactive Anatomy Aystem ANATOM-TUTOR. *User Models and User Adapted Interaction*, 4(1):21–45, 1994.

3. Catriel Beeri and Yoram Kornatzky. A Logical Query Language for Hypertext Systems. In *Proceedings of the ECHT'90 European Conference on Hypertext*, pages 67–80, 1990.
4. D Benyon, D Stone, and M Woodroffe. Experience with Developing Multimedia Courseware for the World Wide Web: The Need for Better Tools and Clear Pedagogy. *International Journal of Human-Computer Studies*, 47(1):197–218, 1997.
5. Paul De Bra and Licia Calvi. 2L670: A flexible adaptive hypertext courseware system. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*, pages 283–284, 1998.
6. T Bray, J Paoli, and C M Sperberg-McQueen. Extensible Markup Language (XML) 1.0. Technical report, W3C, http://www.w3.org/TR/REC-xml, February 1998.
7. P Brusilovsky. Intelligent Tutoring Systems for the World-Wide Web. In R Holzapfel, editor, *Proceedings of Third International WWW Conference*, pages 42–45, 1995.
8. P Brusilovsky. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.
9. P Brusilovsky. Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87–110, 2001.
10. P Brusilovsky, A Kobsa, and J Vassileva, editors. *Adaptive Hypertext and Hypermedia.* Kluwer Academic Publishers, Dordrecht, February 1998. ISBN 0-7923-4843-5.
11. P Brusilovsky, E Schwarz, and G Weber. ELM-ART: An Intelligent Tutoring System on World Wide Web. *Lecture Notes in Computer Science*, 1086:261–269, 1996.
12. Paul de Bra, Geert-Jan Houben, and Hongjing Wu. AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In *Proceedings of the 10th ACM conference on Hypertext and Hypermedia*, Darmstadt, Germany, February 21-25 1999. ACM.
13. A.A.A. Fernandes, M.H. Williams, and N.W. Paton. A logic-based integration of active and deductive databases. *New Generation Computing*, 15(2):205–244, 1997.
14. Cord Hockemeyer, Theo Held, and Dietrich Albert. RATH — A Relational Adaptive Tutoring Hypertext WWW–Environment Based on Knowledge Space Theory. In Christer Alvegård, editor, *Proc. CALISCE 98*, pages 417–423, 1998.
15. Hubertus Hohl, Heinz-Dieter Boecker, and Rul Gunzenhaeuser. Hypadapter: An Adaptive Hypertext System for Exploratory Learning and Programming. *User Modelling and User-Adapted Interaction*, 6(2-3):131–156, July 1996.
16. G Jacobs. Hypermedia and Discovery-based Learning: A Historical Perspective. *British Journal of Educational Technology*, pages 119–129, 1992.
17. J Kay and B Kummerfeld. An Individualised Course for the C Programming Language. In *Proceedings of the Second International WWW Conference*, 1994.
18. A Kobsa, D Müller, and A Nill. KN-AHS: An Adaptive Hypertext Client of the User Modeling System BGP-MS. In *Proceedings of the 4th International Conference on User Modeling*, pages 99–106, 1994.
19. D Lowe and W Hall. *Hypermedia and the Web An Engineering Approach.* Wiley, 1999. ISBN 0-471-98312-8.
20. N Mathe and J Chen. User-centered Indexing for Adaptive Information Access. *User Modeling and User-Adapted Interaction*, 6(2-3):225–261, July 1996.
21. A Negro, V Scarano, and Rosaria Simari. User adaptivity on www through cheops. In P brusilovsky and P DeBra, editors, *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia*, Pittsburgh, USA, 20-24 June 1998.
22. C O'Hagan. Embedding ubiquitous use of educational technology: is it possible, do we want it and, if so, how do we achieve it? *Educational Technology & Society*, 2(4), 1999. ISSN 1436-4522.

Using a rational agent in an adaptive, web-based tutoring system

Matteo Baldoni, Cristina Baroglio, Viviana Patti, Laura Torasso

Dipartimento di Informatica — Università degli Studi di Torino
C.so Svizzera 185, 10149, Torino, Italy
Tel.: +39 011 6706711 — Fax.: +36 011 751603
E-mail: {baldoni,baroglio,patti}@di.unito.it

Abstract. In this paper we describe an adaptive tutoring system, having a multi-agent architecture. The kernel of the system is a rational agent, whose behavior is programmed in the logic programming language DyLOG. In the prototype that we implemented the reasoning capabilities of the agent are exploited both to dynamically build study plans and to verify the correctness of user-given study plans with respect to the competences that the user wants to acquire.

Keywords: Adaptive curriculum sequencing, curricula verification, reasoning about actions, multiagent systems.

1 Introduction and motivation

In current days, there is a growing interest in developing web sites and portals where information is presented (and sometimes selected) according to the reader's preferences and interests. E-commerce, on-line news channels, music broadcasting, recommendation systems are a few examples of possible applications. This form of flexibility is commonly referred to as *adaptation*. Many of the most advanced solutions to web site adaptation [1, 2, 11, 10, 17, 9], assume that adaptation should focus on the user's characteristics (age, education, favorite topics), therefore they vary presentations according to a reference prototype, the "user model", to which the user is associated. By doing so, such approaches catch important aspects connected to the personality and the general interests of the user. Sometimes user models are refined according to observations about the users behavior.

In our work we have studied another kind of adaptation, which could be of great help especially in the case of recommendation systems: adaptation based on the user's *intentions* and *needs*. Often the reasons for which users connect to a web site depend on specific needs, which cannot be inferred from the user's past behavior. The user model approach could, therefore, be enriched by adding to the system the ability to reason about intention, belief and action. In this article we describe the most recent achievements of a work, that we have been doing in the last years by presenting the ways in which our *virtual tutor* supports the definition of study plans.

Intention, as well as belief and action, has intensively been studied in *logics* and *logic programming* settings [18, 14]. In our approach to web site adaptation we use an agent logic programming language, called DyLOG [6, 5], for building cognitive agents, that exploit reasoning techniques for helping users and find ad hoc solutions to their needs. DyLOG is based on a *modal formal theory of actions*, so it can deal with reasoning about action effects in a dynamically changing environment. In this framework, adaptation is interpreted as a *reasoning problem*: thus, for instance, the agent can build a study plan that allows a student to acquire some desired competence, the same agent can also verify whether a study plan proposed by a student is correct (e.g. that course preconditions are respected and the plan will actually allow the student to acquire the desired competence), or diagnose why it is not correct. Independently from the kind of reasoning that is performed, web pages are generated from the system for communicating with the user: in a sense, most of the web site interaction could be considered a form of conversation.

Although our approach could recall some works on Natural Language cooperative dialogue systems, there are some differences. For instance, in 1997 Bretier and Sadek [8] proposed a logic of rational interaction for implementing the dialogue management components of a spoken dialogue system. This work is based, like DyLOG, on dynamic logic but while they exploit reasoning capabilities on actions and intentions to produce proper dialogue acts we use them to produce solutions to the user's problems. In this context, the novelty of our approach is that we exploit reasoning capabilities for building presentation plans guided by the user's goals rather than for dialogue act planning.

See the web site `http://www.di.unito.it/~alice` for technical information about the system.

2 Modeling a virtual tutor as a rational agent: adaptation through reasoning

2.1 The virtual tutor domain

The virtual tutor task is to help students by building (or verifying) study plans, where a study plan is a sequence of courses that the student will attend. Generally speaking, a study plan is supposed to allow a student to acquire a body of coherent and consistent knowledge; for instance, the student will become an expert about "web applications" or an expert about "bioinformatics". Such labels represent the *competence* that will be acquired. This competence, however, is likely to be structured into smaller pieces of knowledge, in different words competence is structured into a *hierarchy of competences*, see Figure 1.

We said that a study plan is a *sequence of courses*. Each course supplies a set of competences; on the other hand, each course can be attended with success only if the student's knowledge satisfies the prerequisites of the course, if any (either

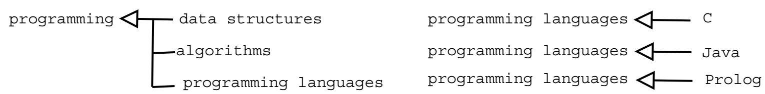


Fig. 1. As an example, this is a little excerpt from our competence hierarchy.

the student attended a set of preparatory courses or he acquired the necessary knowledge in alternative ways). For instance, let's suppose that a programming laboratory course where Java is tought, can be attended if a programming theory course has been attended (*precondition* to the action "attend the Programming Theory course"). When a student attends a course, he will gain competence (*unconditioned effect* of attending the course); some competences, however, may depend on *additional conditions* (*conditioned effect*). For instance, in order to attend our Java lab course it is not necessary to know the C programming language but if I attended a C programming language course I will also be able to understand the explanations about the relationship between C and Java, see Figure 2.

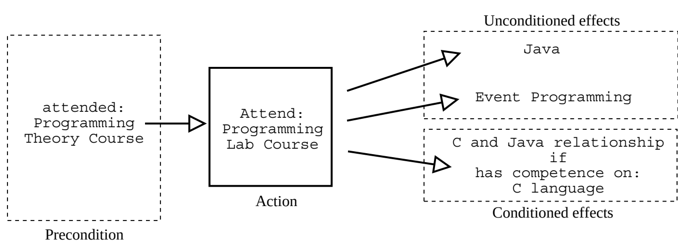


Fig. 2. Attending a course as executing an action.

The use of competence hierarchy-based representations allows a better description of prerequisites. Suppose that a course can be attended by students who know programming languages, independently from the specific course that they attended; it could be attended by students who studied the C programming language, by students who studied Prolog, and by students who studied Java (see Figure 1). Instead, in order to attend a Unix course it would be useful to know the C programming language.

Competence hierarchies can be used also to describe professional expertise independently from the specific courses that are offered and, as we will see, they can be exploited to build flexible study plans, that allow each student to achieve his/her intentions (goals), adapting to the taste and personal interests of each student.

2.2 Adaptation as reasoning about actions

In an *intention-based* adaptive system, *reasoning* is fundamental. In our approach adaptation is intended as an interaction between the user and an intelligent agent aimed at “presenting” and “discussing” study plan proposals. Such proposals are obtained after a reasoning process applied to the domain description (internal to the system) and the specific situations and interests obtained from the user. In the virtual tutoring system, once a student decided the professional expertise or the set of competences he wants to acquire, the system builds plans that lead to the acquisition of the desired knowledge. In general, many alternative study plans would allow the student to reach his goal; in this case, the system must tailor the plan according to the user’s preferences. For instance, in the above mentioned case, in which a student should know a programming language in order to attend a certain course, the student will be asked which programming language course he would prefer to attend, among the available ones. Supposing that our student is interested in Artificial Intelligence, he will be likely to choose the Prolog course.

Building study plans is indeed interesting but the most interesting reason for using a reasoner is that it can actually perform *different* kinds of reasoning, based on the same knowledge base. Planning is just one possibility. Remaining on the same application, let’s now consider this other situation: a student wrote his study plan, he thinks that by following it he will become a “bioinformatics expert”. We still have the “bioinformatics expert” description, the difference w.r.t. the previous case is that now we have a sequence and we want to understand if it will lead the student where he would like to go. A similar case is when a student does not refer to a predefined expertise but collects an own set of desired competences and writes a study plan that is supposed to let him achieve them. Observe that the same kind of reasoning could be exploited by professors, who are defining new study curricula and want to check their consistency.

In order to obtain such a flexible behavior, the core of the system that we developed is a set of *rational agents*. We call our rational agents *reasoners*.

2.3 Implementing the virtual tutor behavior with DyLOG

The language that we used for implementing our reasoners is DyLOG [6, 5]. DyLOG is based on a logical theory for reasoning about action and change in a modal logic programming setting. It allows one to specify a rational agent behavior by defining both a set of simple actions, that the agent can perform (some of which are sensing and suggesting actions for interacting with the user) and a set of Prolog-like procedures, which build complex behaviors upon simple actions. The DyLOG intepreter allows both to execute the procedures, which define the agent behavior, and to reason about their execution, extracting (possibly) conditional plans [3]. The plan extraction process of the interpreter is a straightforward implementation of the proof procedure contained in the theoretical specification of the language.

In our virtual tutor *actions* correspond to *attending courses*. For each "attend course" action, we define its *preconditions* and *effects*: preconditions say when the course can be attended (in terms of passed courses and acquired competences), effects describe the competences that will be acquired. Effects can be further conditioned to more specific requirements (see Figure 2).

As we have seen, *competences* are organized into a set of hierarchies. Hierarchies are built upon sets of *causal implications*. For instance, the hierarchy shown in Figure 1 is represented as:

$has\_competence(programming)$ **if**
  $has\_competence(data\_structures) \wedge$
  $has\_competence(algorithms) \wedge$
  $has\_competence(programming\_languages)$.
$has\_competence(programming\_languages)$ **if**
  $has\_competence(c\_language)$.
$has\_competence(programming\_languages)$ **if**
  $has\_competence(java\_language)$.
$has\_competence(programming\_languages)$ **if**
  $has\_competence(prolog\_language)$.

In the above rules we state that the "programming" competence is achieved if we have competence about data structures, algorithms and we know a programming language (C, Java, or Prolog). *Professional expertise* is formalized as a set of implications that are built upon competence hierarchies. They just add some more level of abstraction.

The behaviour of our agent is described by a collection of procedures.

In the case of *study plan construction* the top level procedure, called *advice*, extracts a plan that will be executed (see Section 3).

(R1) $advice(Plan)$ **isp**
  $ask\_user\_preferences \wedge ?requested(Curriculum) \wedge$
  $generate\_goal(has\_competence(Curriculum)) \wedge$
  $plan(has\_competence(Curriculum) \ \wedge \ credits(C)$
    $\wedge \ max\_credits(B) \wedge \ (C =< B)$ **after** $combine\_courses, Plan)$.

The reasoner asks the student what kind of final expertise he wants to achieve and his background knowledge (e.g. if he already attended some of the possible courses). Afterwards, it adopts the user's goals and builds a *conditional plan* for reaching them, predicting also future interactions with the user. That is, if it finds different courses that supply a same competence, whose prerequisites are satisfied, it plans to ask the user to make a choice. *plan* is the metapredicate that actually builds the plan, in this case by extracting the executions of the procedure *combine_courses* that satisfy the user's goals as well as the further conditions that are specified (e.g. that the number of credits gained with the study plan is not bigger than a predefined maximum). [1]

[1] Note that the above formulation of the behaviour of the agent, has many similarities with agent programming languages based on the BDI paradigm such as dMARS

The way the agent combines the courses into a curriculum is specified by procedure *combine_courses* that, until the study plan is believed complete, tries to achieve the goal of acquiring still missing competences by adding a new course.

(R2) $combine\_courses$ **isp**
$?study\_plan\_complete.$
$combine\_courses$ **isp**
$?(\neg study\_plan\_complete) \wedge$
$achieve\_goal \wedge$
$combine\_courses.$

Note that the main goal of having a study plan to propose to the student is reached and the curriculum is complete only when all of the goals of getting some piece of competence are fulfilled. Indeed, we assume the behaviour of a rational agent to be driven by a set of goals, which are represented as fluents[2] having form $goal(F)$. Our agent detects its goals based on student's explicit inputs and its expert competence about learning and courses combination. Initially the agent does not have explicit goals, because no interaction with the student has been performed. The student's inputs are obtained after the first interaction phase carried on by the procedure $ask\_user\_preference$ in (R1):

(R3) $ask\_user\_preferences$ **isp**
$verify\_student\_competence \wedge$
$offer\_curriculum\_type$

$verify\_student\_competence$ allows to acquire knowledge about current curriculum studiorum of the student (if any), whereas $offer\_curriculum\_type$ allows the agent to acquire knowledge about the professional expertise the student would like to achieve. In DyLOG information is gathered from the user by means of special actions, called *sensing actions* [6]. Differently than "normal" actions, they increase (or revise) the knowledge of the agent but they do not change its environment. An example of sensing action is $offer\_curriculum\_type$:

(R4) $offer\_curriculum\_type$ **possible if** $true.$
$offer\_curriculum\_type$ **senses** $requested(Curriculum).$

After executing $offer\_curriculum\_type$ the value of the fluent $requested(Curriculum)$ (used in (R1)) will be known and the main goal

$$goal(has\_competence(Curriculum))$$

will be inferred; here the actual *goal adoption* occurs. Then, the system will exploit the competence hierarchy-based representation for generating a set of

[12]. As in dMARS, plans are triggered by goals and are expressed as sequences of primitive actions, tests or goals.

2 In action theory, *fluents* denote properties of the world whose truth value may change over the time.

subgoals that are to be achieved to compose a study plan for the selected curriculum (*generate_goal* in (R1)).

After adopting a goal $goal(F)$, the agent acts so to achieve it until it believes the goal is fulfilled.[3]

(R5) *achieve_goal* **isp**
$?goal(knows(X, \_)) \wedge (X \neq generic) \wedge$
$add\_course(X).$
achieve_goal **isp**
$?goal(knows(generic, Keyword)) \wedge$
$?(u(has\_competence(Keyword))) \wedge$
$offer\_course\_on(Keyword) \wedge$
$?course\_on(Keyword, X) \wedge$
$add\_course(X).$

The procedure *achieve_goal* allows the agent to select in a non deterministic way the goal of adding a course to the specific study plan that is being built. There are two cases. Either the goal requires to add a specific course among the available ones or, more generically, the goal requires to achieve a certain competence, given the student did not acquire it yet ($u(has\_competence(Keyword))$). In the latter case, the agent interacts again with the user to decide what specific course to add according to the user's preference ($offer\_course\_on$).

(R6) $offer\_course\_on(\_)$ **possible if** *true*.
$offer\_course\_on(Keyword)$ **suggests** $course\_on(Keyword, \_)$.

Note that $offer\_course\_on$ is a *suggesting action*. In [3] we introduced suggesting actions as a special case of sensing actions. As a difference with normal sensing actions, when performing this kind of actions the agent does not read an input in a passive way but it has an active role in selecting, after some reasoning, the possible values among which the user will make his choice. In particular only those values that lead to fulfill the goal will be selected. Figure 3 shows an example where the agent is helping a student to build a bioinformatics study plan: at a certain point of the plan construction, the agent finds four alternative courses that give competence about "programming languages", however, only two subtrees allow the student to get competence about imperative languages, necessary for a bioinformatics curriculum. The other branches are cut during the reasoning phase and the corresponding courses are not offered to the student.

Formally, the above described kind of reasoning is called *temporal projection*. Given the description of a domain and an initial situation (e.g. the student intial competences), the temporal projection task consists in predicting the future effects of actions on the basis of (possibly incomplete) information on preceding states. Briefly, see [6] for details, we formalize the *temporal projection problem* "given an action sequence $a_1, \ldots, a_n$, does the condition *Fs* hold after the execution of the actions sequence starting from the initial state?" by the query *Fs*

[3] This corresponds to adopt a *blind commitment strategy*.

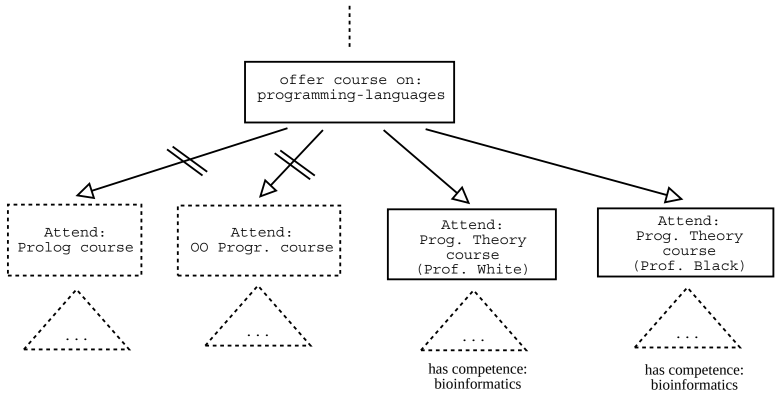


Fig. 3. An example of selection of courses to offer for consructing a bioinformatics curriculum.

after $a_1; \ldots; a_n$, where *Fs* is a conjunction of *fluents*. We can generalize this query to complex actions (procedures) $p_1, p_2, \ldots, p_n$ by:

$$Fs \textbf{ after } p_1; p_2; \ldots; p_n \tag{1}$$

where p_i, $i = 1, \ldots, n$, is either an atomic action (including sensing actions), or a procedure name, or a test. Query (1) succeeds if it is possible to find a (terminating) execution of $p_1; p_2; \ldots; p_n$ leading to a state where *Fs* holds. Intuitively, when we have a query *Fs* **after** p we look for those *terminating execution sequences* which are plans to bring about *Fs*. In this way we can formalize the *planning problem*: "given an initial state and a condition *Fs*, is there a sequence of actions that (when executed from the initial state) leads to a state in which *Fs* holds?". The procedure definitions constrain the search space of reachable states in which to search for the wanted sequence. In the virtual tutor application, the procedure p is "combine_courses" while *Fs* is the condition $has\_competence(Curriculum) \wedge credits(C) \wedge max\_credits(B) \wedge (C =< B)$.

The second kind of reasoning that we implemented is verifying whether a student-given plan will allow him to achieve some desired competence. In this case we still adopt temporal projection but in its simpler form: we have a sequence of actions and a condition that should hold after their execution. In the virtual tutor application, the sequence of actions is a sequence of courses to attend and the final condition is a set of desired competences: $has\_competence(c_1) \wedge \ldots \wedge has\_competence(c_n)$ **after** $attend(course_1); \ldots; attend(course_m)$:

(R2) $check\_plan(Plan, Competences)$ **isp**
$\quad plan((Competences) \textbf{ after } (Plan), \_)$.

3 The virtual tutor as a multiagent system

WLog, the prototype system that we developed, has a *multi-agent system architecture*. Agent technology allows complex systems to be easily assembled by

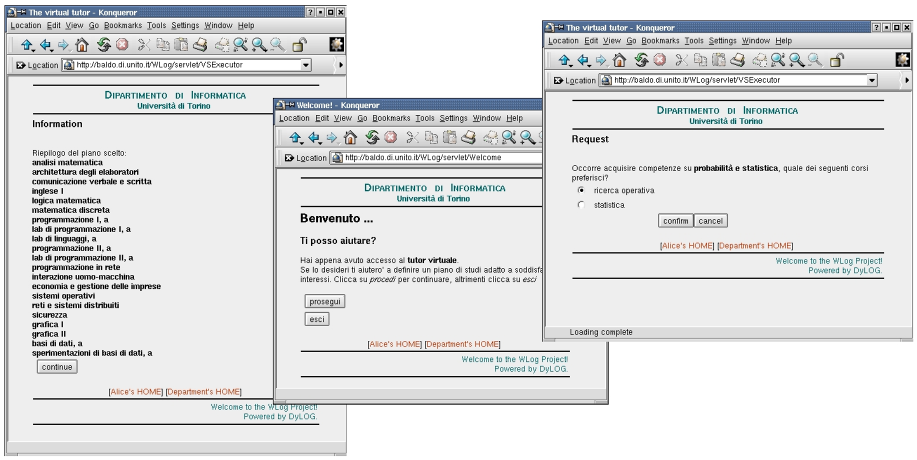


Fig. 4. Interacting with WLog.

means of the creation of distributed artifacts able to accomplish their tasks through cooperation and interaction. Systems of this kind have the advantage of being modular and, therefore, flexible and scalable. So, on one hand, each module can be developed by exploiting the best, specific technology for solving a given issue, on the other, new components can be added for supporting either new functions or a wider number of users.

WLog consists mainly of two kinds of agents: *reasoners* and *executors*[4]. We already know that reasoners are written in DyLOG, executors, instead, are Java servlets embedded in a Tomcat web server.

In the case of study plan extraction, the reasoner task is to extract a conditional plan. Such a plan is a tree, where *each path* to a leaf is actually a study plan that will allow the student to acquire the desired competence. Branches represent alternative courses to acquire a same piece of competence. The execution of the conditional plan is aimed at helping the student to choose the path that better satisfies his/her personal interests. So the student will obtain a correct study plan that is also adapted to his preferences.

The interaction between a user and a reasoner is carried on by means of an *executor*. The task of an executor consists in showing one or more web pages to the user; in particular, when a page that corresponds to a branching point is shown, a feedback from the user is requested.

The communication among the agents has the form of message exchange in a distributed system; message exchange is FIPA-like [13]. Each agent is identified by its "location", which can be obtained by other agents from a facilitator. Each agent has a private mailbox where it receives messages from other agents (see Figure 5).

[4] For a more detailed description of the system's architecture we refer to [4].

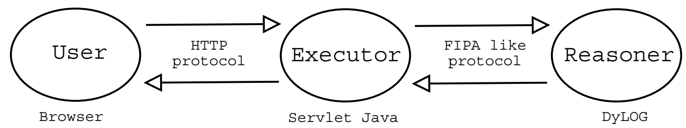


Fig. 5. A sketch of the WLOG multiagent system.

3.1 Interaction between a tutor, an executor, and a user

Users access to the system by means of a normal web browser; until the end of the interaction, the interface between the user and the reasoner will be an *executor*. First the executor looks for a free reasoner, if any is available. At the moment reasoners are not differentiated (they all can perform the different kinds of reasoning).

Supposing that the previous step was successful, the interaction between the user and the system starts with the declaration of the user's goal ("I want to become an expert of bioinformatics"). The user's goal is adopted by the reasoner, that will start a conversation aimed at collecting information about the user initial situation. For instance, the user will be asked about successfully passed exams. In the case of study plan validation, instead, the system will ask the study plan to evaluate and the competences to acquire. The leader of the conversation is the reasoner, which will send the information or the questions to the user with the help of the executor. At this point it is extremely interesting to understand how the interaction between the reasoner and the executor is carried on.

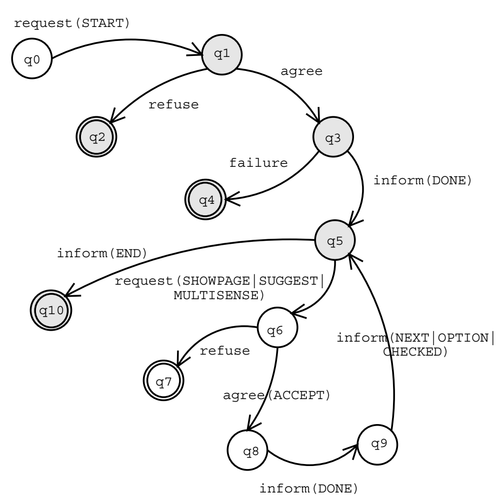


Fig. 6. Communication protocol between an executor and a reasoner.

In Figure 6, the interaction that occurs between a reasoner and an executor is specified by using a finite state automaton. Such an automaton represents the *interaction protocol* that is respected by the couples <reasoner, executor>. States are numbered and arcs are labeled with the speech act that causes the transition between states. Different shading on states are used for specifying which agent's turn it is to continue the conversation (white for the executor, grey for the reasoner). States with double border are terminating states.

The q_1-q_4 states concern the initialization phase for connecting a certain executor with a reasoner. The q_5-q_{10} states concern the *actual action execution cycle*. They state that when a reasoner executes an action (which can be part of a plan extracted after a reasoning process, or simply an atomic action in a DyLOG procedure that is being executed) the execution code associated to the action sends to the executor a request of showing a given HTML page. Both agents continually check the sender of the messages that they receive, so if, for instance, an executor receives a message from a reasoner which is not serving its user, it will refuse that execution. The same would happen if it were asked to perform an action that it is not supposed to perform in the current state. For the sake of clarity, suppose that the executor has just sent a form to the user's browser and is waiting for data. If meanwhile it receives from the reasoner the request to show another page, it will refuse to do it.

Supposing that the executor accepted to perform a requested action, it composes a proper HTML page and sends it to the user's browser. In some cases the page will contain a form to be filled. When the user finishes to consult/fill the page and asks to go on, the executor informs the reasoner that the page has been consulted. When requested, it also transmits to the reasoner the user's data, that can be used by the reasoner to update its knowledge about the user's goals (or background information); otherwise it only informs that it is possible to go on with the next action if there is any.

4 Conclusion and future work

In this paper, we have presented a web-based virtual tutor whose kernel is a reasoner, implemented in a logic programming language to reason about action and change, DyLOG. In such an approach we have, on one hand, a high-level description of the domain (e.g. the connection between two competences can be represented by means of a causality relationship), that is close to our intuition and the way in which we, human beings, handle this sort information. The advantage is that we can maintain or modify these descriptions in an easy way. On the other hand, the system interacts with the user in order to find out his goals in the context of the application framework and applies various kinds of reasoning, implemented by means of a logic derivation mechanism, for finding a solution that perfectly fits the specific user.

In our implementation, both planning (constructing a study plan) and verifying a student-given study plan are performed on-line. In the case of planning we could have followed an alternative approach: to build off-line the most general conditional plan for each professional expertise and to prune (on-line) the resulting tree during the interaction with the user. This cannot be done efficiently neither in the case in which the user asks to build a plan for achieving a generic set of competences nor in the case of plan verification. In fact, finding out if a sequence is an instance of a schema by pruning the schema tree has a high computational complexity whereas verifying its correctness by applying *temporal projection*, that is to verify whether the linear plan is compatible with the domain description, is linear in the number of the elements in the sequence.

In the case in which the study plan that the system verified resulted to be wrong, it would be extremely useful to find out the reasons for which it is wrong, so to better help the user. Basically the reasons for which a study plan is wrong are of two kinds: either the sequentialization of courses is not correct or in the end the student will not achieve the desired competence. We are currently working on the application of other kinds of reasoning (diagnosis by abduction and postdiction) to achieve this goal in the line of [7] where a formal account of diagnostic problem solving is provided in term of an action language.

Adaptation in e-learning is a open problem and is being tackled by many researchers under many perspectives. In the terminology of works like [15, 16], our work focuses on "knowledge dependencies", separating them from the actual knowledge. As a difference, while works like [15] are based upon a partial order of "knowledge items" and Bayesian networks, we base our work on a *modal logic theory of action*. In the former knowledge dependencies of the form $K1 < K2$ express the fact that $K1$ should be learned in order to learn $K2$. Therefore, the transitive closure of the "<" relation is the inferencing mechanism that allows to the system to understand the dependencies between sets of knowledge items. In our case, we identify both a set of "knowledge items" (the courses) and a set of *hierarchical competences*. Competences are abstract descriptions about knowledge, which are not necessarily contained in the index (or program) of the course. In our course descriptions we have prediconditions to attending the course (they can consist of competences as well as of other courses) and (un)conditioned effects of attending the course. The dependencies between sets of "knowledge items"/competences are based on a *logic derivation* [6].

References

1. L. Ardissono and A. Goy. Tailoring the interaction with users in electronic shops. In *Proc. of the 7th Int. Conf. on User Modeling*, 1999.
2. L. Ardissono, A. Goy, G. Petrone, and M. Segnan. A software architecture for dynamically generated adaptive web stores. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence, IJCAI'01*, Seattle, Oregon, USA, 2001.
3. M. Baldoni, C. Baroglio, A. Chiarotto, and V. Patti. Programming Goal-driven Web Sites using an Agent Logic Language. In I. V. Ramakrishnan, editor, *Proc. of the Third International Symposium on Practical Aspects of Declarative Languages*, volume 1990 of *LNCS*, pages 60–75, Las Vegas, Nevada, USA, 2001. Springer.
4. M. Baldoni, C. Baroglio, and V. Patti. Structereless, Intention-guided Web Sites: Planning Based Adaptation. In *Proc. 1st International Conference on Universal*

Access in Human-Computer Interaction, a track of HCI International 2001, New Orleans, LA, USA, 2001. Lawrence Erlbaum Associates.

5. M. Baldoni, L. Giordano, A. Martelli, and V. Patti. Modeling agents in a logic action language. In *Proc. of the Workshop on Rational Agents, FAPR'00*, London, September 2000.
6. M. Baldoni, L. Giordano, A. Martelli, and V. Patti. Reasoning about complex actions with incomplete knowledge: a modal approach. In *Proc. of the 7th Italian Conference on Theoretical Computer Science, (ICTCS01)*, volume 2202 of *LNCS*, Torino, Italy, October 2001.
7. Chitta Baral, Sheila A. McIlraith, and Tran Cao Son. Formulating diagnostic problem solving using an action language with narratives and sensing. In *Principles of Knowledge Representation and Reasoning, KR 2000*, pages 311–322, 2000.
8. P. Bretier and D. Sadek. A rational agent as the kernel of a cooperative spoken dialogue system: implementing a logical theory of interaction. In *Intelligent Agents III, proc. of ECAI-96 Workshop on Agent Theories, Architectures, and Languages (ATAL-96)*, LNAI, 1997.
9. P. Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110, 2001.
10. B. De Carolis, S. Pizzutilo, and F. de Rosis. User manuals as animated agents. In *Proc. of the AI*IA Workshop su "Agenti intelligenti e internet: teorie, strumenti e applicazioni"*, Milano, Italy, 2000.
11. B.N. De Carolis. Introducing reactivity in adaptive hypertext generation. In *Proc. 13th Conf. ECAI'98*, Brighton, UK, 1998.
12. M. d'Inverno, D. Kinny, M. Luck, and M. Wooldridge. A formal specification of dMARS. In *Proc. of ATAL'97*, volume 1365 of *LNAI*, pages 155–176, 1997.
13. FIPA. FIPA 97, Specification part 2: Agent Communication Language. Technical report, Foundation for Intelligent Physical Agents, 1997.
14. M. Gelfond and V. Lifschitz. Representing action and change by logic programs. *Journal of Logic Programming*, 17:301–321, 1993.
15. N. Henze and W. Nejdl. Bayesian modeling for adaptive hypermedia systems. In *Proc. of ABIS99, 7. GI-Workshop Adaptivität und Benutzermodellierung in Interaktiven Softwaresystemen*, Magdeburg, September 1999.
16. Nicola Henze and Wolfgang Nejdl. Extendible adaptive hypermedia courseware: Integrating different courses and web material. In Peter Brusilovsky, Oliviero Stock, and Carlo Strapparava, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems, International Conference, AH 2000*, pages 109–120, 2000.
17. L. Marucci and F. Paternò. Designing an adaptive virtual guide for web application. In *Proc. 6th ERCIM Workshop UI4All*, Florence, Italy, 2000.
18. R. Scherl and H. J. Levesque. The frame problem and knowledge-producing actions. In *Proc. of the AAAI-93*, pages 689–695, Washington, DC, 1993.

An Adaptive Web-based Tutorial of Agrarian Economy

Cristina Carmona, Eduardo Guzmán, David Bueno, Ricardo Conejo

Departamento de Lenguajes y Ciencias de la Computación.
E.T.S.I. Informática. Universidad de Málaga. Apdo. 4114, 29080 Málaga. SPAIN.
{cristina,guzman,bueno,conejo}@lcc.uma.es

Abstract. *TEA* was originally conceived as a Computer Aided Educational System for teaching Agrarian Economy. This system was composed by web pages of instructional contents and web pages of test questions. The biggest flaw of the system is the lack of control over the pages seen by the student. This contributed to the disorientation of students during the instruction. Also, in test questions, students only could know if they had answered correctly to it, but these questions did not give them any feedback about their global level of knowledge in the topics of the tutorial. In this paper, *TEA* system has been improved making it adaptive. This has been achieved by using two tools: *SIGUE* and *SIETTE*. *SIGUE* is a web-based tool that converts non-adaptive courses into adaptive, and *SIETTE* is a web-based evaluation tool using adaptive tests. The functionalities of both tools have been integrated into a homogeneous adaptive *TEA*.

1 Introduction

There are many educational systems on the World Wide Web (WWW) that allow students to learn notions about different subjects. One of the main disadvantages of web-based systems is the disorientation during the navigation. In textbooks the order of the knowledge units is defined by the book's author, but in web courses this order is not so clear. The availability of multiple links in the same web page and the freedom to follow any of them, is not always an advantage for students. They can navigate without following a rational order in the pages of a course, and they can navigate in pages not related with the course. In these cases, it is necessary to have some control technique to guide the user during the study of a subject.

On the other hand, in most systems, there are no mechanisms to infer if students have assimilated the concepts studied in a page. Some mechanism of evaluation is required to assess the students' proficiency before letting the students access the pages of new concepts.

A characteristic of adaptive tutoring systems, is that they can estimate the knowledge level of a user in relation to a knowledge unit of a course. This information can be used to recommend which unit should be studied next. Adaptive tutoring systems provide the possibility of modifying the sequencing or presentation of the course for each user, according to their goals, current knowledge and/or preferences. The drawback is that, generally, they are difficult to implement due to the high development cost. So, most of existing tutoring systems on the web are not adaptive.

TEA [1] (the acronym stands for *Agrarian Economy Tutorial* in Spanish) was originally developed in the early years of the web, as a multimedia instructor system of Agrarian Economy. It is designed as a set of static web pages structured into chapters. Each chapter is also divided into sections. Special care was taken during the definition of the chunks of information in each page. At the end of each chapter, a set of questions is proposed to the students to evaluate their proficiency. *TEA* is available at http://www.lcc.uma.es/TEA .

The main goal of this paper is to demonstrate how a static instructor system can be turned into an adaptive course. To do this we will use of the *TEA* course. Two different tools have been used to this end: *SIGUE*, that allows building adaptive web pages from static ones; and *SIETTE*, which is a web-based system to construct adaptive tests.

In the next section, the *SIGUE* system is described, indicating the procedure followed to provide adaptive feature to *TEA*. In Section 3 a brief description of the main features of *SIETTE* is presented, as well as the adaptive mechanisms which it uses. This section ends with a description of *TEA* test in *SIETTE*. Section 4 explains how the integration of both systems has been accomplished. At last, contributions obtained from this work are approached.

2 Creating contents with SIGUE

SIGUE (http://www.lcc.uma.es/SIGUE) is an authoring tool that provide some adaptive features to existing static web-based courses. It is possible to create new courses reusing existing pages from different sources. This lets authors make courses by gathering the best information on the Web about a subject. For each concept of the course the author can choose how many URLs can be associated with this concept, indicating the kind of content (theory, examples or exercises) and the importance of this URL to the concept.

SIGUE provides a student's interface with commonly used adaptive tools, like annotated table of contents, colored buttons and progress bars. The student's interface is implemented as a frame that serves as an additional navigation toolbar. Navigation on the main frame is also controlled depending on the specification of the teacher. For students that need guidance, *SIGUE* can recommend the next concept (and the best document within it) to go on, according to the student's estimated knowledge. Another aspect of adaptivity is the multilingual interface of *SIGUE*, currently available in Spanish and English. A deeper description of *SIGUE* can be found in [11]

2.1 Making TEA adaptive

To create a course the first task to do is to define the domain model associated. The domain model is structured hierarchically. The root node contains the general information about the course. The nodes represent section and subsection. Leaves nodes have the information about the concepts being taught. *SIGUE* defines two binary relationships between nodes: "*belongs to*" and "*prerequisite of*". The first defines the hierarchy. If the author defines that the concept c1 *belongs to* c2, this means that c1 is a sub-concept of c2. One concept can have many sub-concepts but only one super-concept. The author can also define that c1 is *prerequisite of* c2. This means that to learn c2 it is necessary to study c1 first. The relationship of *prerequisite* defines a relationship of partial order. This gives an idea of the order in which concepts should be visited, which is the sequence of the curriculum.

The number of pages to describe a concept is unlimited, but there should be a main one that is shown the first time that a concept is visited. Authors must indicate the type of information that each page contains (theory, examples or exercises) and the difficulty (easy, normal or hard). The difficulty is used to order the pages within a concept. The way to present the pages will be; first, the main page followed by the rest, which are shown in order from the easiest to the hardest.

Once the domain model is created and all the concepts and relations have been defined, the author has to associate pages to concepts. The author can also define a glossary of terms for the course. A definition is associated to each term in the glossary. Optionally, a list of synonyms can refer back to the same definition. When a term defined in the glossary appears in any page of the course, a link is automatically inserted to show the definition of this term. This can be done in three ways: as a *hint*, this means that when student moves the mouse over the term, a textbox will show the definition; as an URL, the link will show a URL with the definition; or as an HTML page. In the last case, the author only has to write the HTML code that the student will see when he clicks on the term.

Courses in SIGUE can have different modes of operation. The author decides how adaptive the course will be and must set the mode of operation accordingly. There are four predefined modes: (a) Disable all the links in the pages shown to the student. In this mode the student will be able to do the course by accessing documents only through the links provided by the SIGUE navigation toolbar; (b) Leave all links of the pages, this lets users navigate freely even in pages not related to the course; (c) Enable only the links that give references to pages that belongs to the course. This option avoids distracting the user with pages not related to the course; (d) Fully adaptive. The links will be enabled according to the user model. Only those links corresponding to concepts that the user is ready to learn will be activated.

Fig. 1. The interface of *SIGUEAUTOR*.

To construct an adaptive course, it is necessary to define the prerequisite relationship of the concepts. This definition must be as complete as possible. This will be used to guide the student correctly in the sequence of concepts to visit, establishing a partial order for the learning process.

Fig. 1 shows the *SIGUE* authoring interface (*SIGUEAUTOR*). The client window is divided in two frames. On the left frame, the *SIGUEAUTOR* navigation toolbar is shown, the table of contents tree, the prerequisites table, the URL table, the access to the glossary, the mode of operation selector and the buttons to save the course and generate the XML file. On the right frame, the web browser shows the page visited. Each page is parsed by *SIGUEAUTOR* and its links are changed so that they will invoke *SIGUEAUTOR* again when they are clicked. Navigation is intercepted by *SIGUEAUTOR*, but this mechanism is transparent to the author. In this way, Authors can freely navigate and *SIGUEAUTOR* engine can always knows which page is loaded in the right frame. When an appropriate page is reached the page can be attached to a concept of the course by using the left frame .

2.2 Studying with TEA

Once the course has been developed with *SIGUEAUTOR*, it is available to students in *SIGUE*. While connected to the web through *SIGUE*, the student will see the hierarchical structure of concepts created by the author. All links and forms are modified according to the strategy selected by the author for the course. Some of them are eliminated and others are left (apparently) unchanged. The system will also include new references for the terms in the glossary, as defined by the author.

If the course is fully adaptive, a student model is created for each user. For each concept *SIGUE* shows two indicators. (a) The estimated background of the student to visit this page, that is, if he is prepared or not to read it, according to previous pages that have been visited. (b) The second is an indicator of the percentage of pages related to that concept that the student has already visited.

SIGUE does not have the ability to evaluate, it simply makes estimations of the knowledge based on the percentage of visited pages of each concept. So, if the percentage of URLs visited for a concept is less than the minimum the status of the concept is “empty”; if this percentage of visited pages is bigger than the maximum, the status is “full”. The intermediary case is shown as “half-full”. The status of a concept is associated to the percentage of pages the user has studied within a concept. So if the user has visited all the pages the system assumes that he has already learnt the concept. If it is “empty” the pages have not been visited yet and the system assumes that the concept is not known. This information is shown to the user by a progress bar that appears next to each concept.

The level of preparation necessary for a user to visit a concept is reflected by using colors in the nodes of the concept tree. Brusilovsky [9] justified the utilization of link annotation (equivalent to our color codes) through experiments done with their system *InterBook*. This system is also used in the lisp tutorial *ELM-ART* ([8] [10]). The use of this kind of annotation makes the user feel that there is some extra help in his interactions with the system, even if a “cognitive overhead may distract users from the content” [9].

In *SIGUE* the colors used are just green, red and orange. The color of the node is decided using the status of the prerequisites in this way:

- *Green*: A concept has this status when all the prerequisites are shown to have their status as “full”. This means when all the prerequisites of this concept have already been covered, so the concept can be studied without difficulties.
- *Red*: This will be the color if at least one of the prerequisites of a concept is “empty”. This means that the user has not studied one or some of the prerequisites. Red indicates that the user is not ready to study the concept.
- *Orange*: This indication will appear when no empty prerequisites exist. The student has begun to study all the prerequisites but has not finished all of them. Nodes with this color could be studied but finishing all the prerequisites first is recommended.

The student model is updated after each interaction, that is, every time the user visits a page. The concept tree is updated modifying the status and color of nodes accordingly.

The aim of *SIGUE* is to guide the student’s navigation, and support it with adaptive annotation, but at the same time let the student move freely through the pages of the course. Access to any page is permitted for the user even if this is not recommended Another adaptive feature used to guide students is a simple planner that the student can use to get the best recommendation for the next document to view from the concept he is studying. (*SIGUE* means *continue* in Spanish.). This mechanism is designed for “two buttons users” and they are labeled BACK and NEXT.

Fig. 2 shows an example of the *SIGUE* student’s interface for *TEA*. This course is adaptive, and in the image the progress bar for each concept and the colors that indicate the preparation of the student for each concept can be seen. On the right, the

HTML document associated with the concept and the activated link of recommended concepts and glossary terms (in this case activated by a hint).

Fig. 3 Example of the student's interface of SIGUE for TEA.

3 Creating tests with SIETTE

SIETTE (System of Intelligent Evaluation using Tests for Teleeducation) [2] is an adaptive web-based test generation system (http://www.lcc.uma.es/SIETTE. It has two main parts: (1) a set of *authoring tools*, that allow teachers to add and update contents in the knowledge base of the system. (2) A *virtual classroom* where students can make tests to evaluate their knowledge in certain subject.

The knowledge base is composed by the concept domain (*curriculum*), the specifications of the tests and the item pool. Concepts are structured hierarchically. The subject is the root node and the topics, and subtopics are its children. Items (that is questions) can be associated to any node in the hierarchy. An inheritance mechanism assumes that items of lower levels can also be used to evaluate their antecessors.

Tests are configured by teachers and they are defined by topics. To create a test, teachers must indicate which topics are going to be evaluated. When a student selects this test, the test generator will show him items from all topics involved in the test, or from any descendent subtopic of these topics. Therefore, there is not any direct relation between items and tests. This relation is established through topics.[11]

SIETTE is an adaptive evaluation tool. This means that the selection of the next question to pose to the students, the finalization decision of tests and the mechanism of estimation of the student’s knowledge, are accomplished according to adaptive mechanisms. This kind of tests are known as *Computerized Adaptive Tests* (CAT) [3].

SIETTE, as well as the most of CAT systems, uses as an inference machine a psychometric theory called *Item Response Theory* (IRT) [4].

IRT is based on the hypothesis that the answer given to each item of the test, probabilistically depends on the knowledge level. As a result, conditional probabilities of the correct answer to the item by a student with a certain *knowledge level*, can be easily calculated for each item. This probability is expressed by means of a function $f : ?? ? , ? ? ?? [0,1]$, named *Item Characteristic Curve* (ICC). The calculus of the ICC can be accomplished by several models. *SIETTE* uses a model of three parameters based on the logistic function [5].

Also, in IRT, the knowledge level of the student is estimated using the response to each item of the test. There are several methods to get this value. In *SIETTE* a Bayesian method [6] is used. In this method, the probability distribution of the student’s knowledge level is calculated by the Bayes’ rule. Also, it is assumed that the knowledge level can only take *K* discrete values (from *0* to *K-1*) because of the high computational cost of the calculus. Thanks to this consideration, the Bayesian method for the estimation of the student’s knowledge level can be simplified to a vectorial product of ICC vectors with the *a priori* normalized density vector.

SIETTE does not need to establish priority relationships between topics. That is, there is no need to explicitly indicate if items from a topic must be posed to student before the items of other topics. If the characteristic curves of the items are well calibrated [7], thanks to the adaptive selection mechanisms, the test generator will show to the student the most suitable item according to his estimated knowledge level. This means that, if a student fails an item of difficulty *d*, the next item which the generator will pose him, will have a difficulty lower than *d*. Generally, the calibration process of items will make that items from the first topics of the curriculum, have lower difficulties than items of the latter topics.

The finalization criterion is configured by the teacher in each test. Through test editors, he must indicate a minimum and a maximum number of items. These values set bounds to the number of items that may be posed to the students. If a student has taken the maximum value of items, the final estimation process of his knowledge is forced. This ensures that test will finish, although the finalization criterion is not satisfied.

3.1 The test of TEA

In *SIETTE* a subject of Agrarian Economy has been defined. Its *curriculum* is composed by 14 topics. Each of this topics corresponds with one of the chapters of the *TEA* tutorial. There is a total of 84 items stored in the knowledge base. Each topic has approximately six of these items that are used to evaluate the knowledge of the student about this topic.

A test for each topic has been created. Using *SIGUE*, once the students has complete a chapter, they can directly access the corresponding tests in *SIETTE*, and evaluate their proficiency in that chapter. Additionally, a global test of the whole subject has been defined in *SIETTE* by using the inheritance mechanism.

All tests have been defined for twelve knowledge levels. Therefore, at the end, the system will give a qualification between 0 and 11. All items are multiple choice items, *i.e.*, items where students may either select only one response of a set of options, or not select anyone.

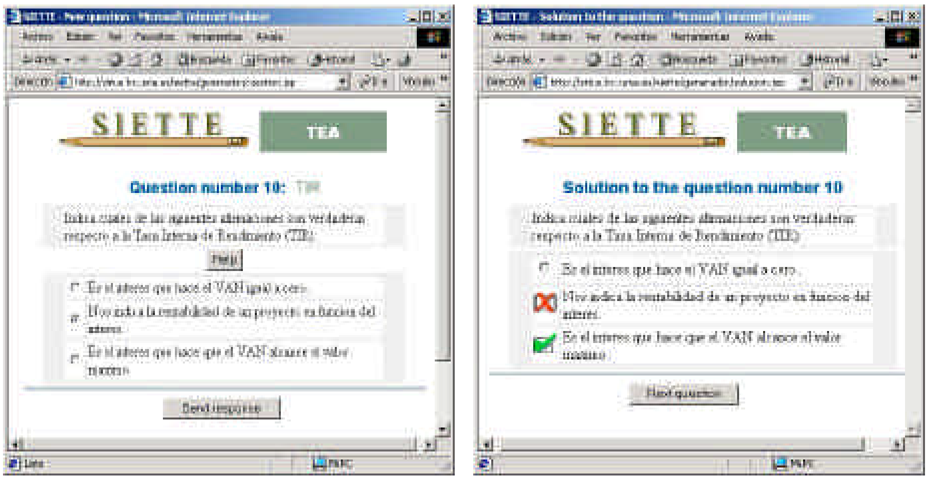


Fig. 4. An item posed by the test generator and its correction.

Fig. 4 shows an item presentation inside a test session (*a*), and its correction (*b*). Each item is composed by the title, the stem and the set of options. Once the student pushes the *Send Response* button, the system will show the correction of this item. Correct option is marked in green. If student selects a wrong option, it will be marked with a red cross. *SIETTE* can be configured to show the correct answer right after the student answer, at the end of the test or never. For educational purposes is advisable to correct the error as soon as it is detected. Different test could be make for practicing and for assessment, but currently the total number of question is not too high.

Fig. 5 shows the final qualification of the student in the test of the whole subject. As it can be seen, the system provides a detailed set qualifications for each topic. Statistics about the number of items posed to the student are given, as well as the number of items that have been successfully answered by the student. Note that the estimated knowledge level does not meet with the number of items successfully passed. This is due to the final qualification has been inferred by an adaptive criterion. At last, a pie chart with the percentage of topics posed from each topic, and the distribution curve of probabilities of that the student has each one of the knowledge levels, are presented too. Note that the value with the highest probability corresponds with the estimated knowledge level.

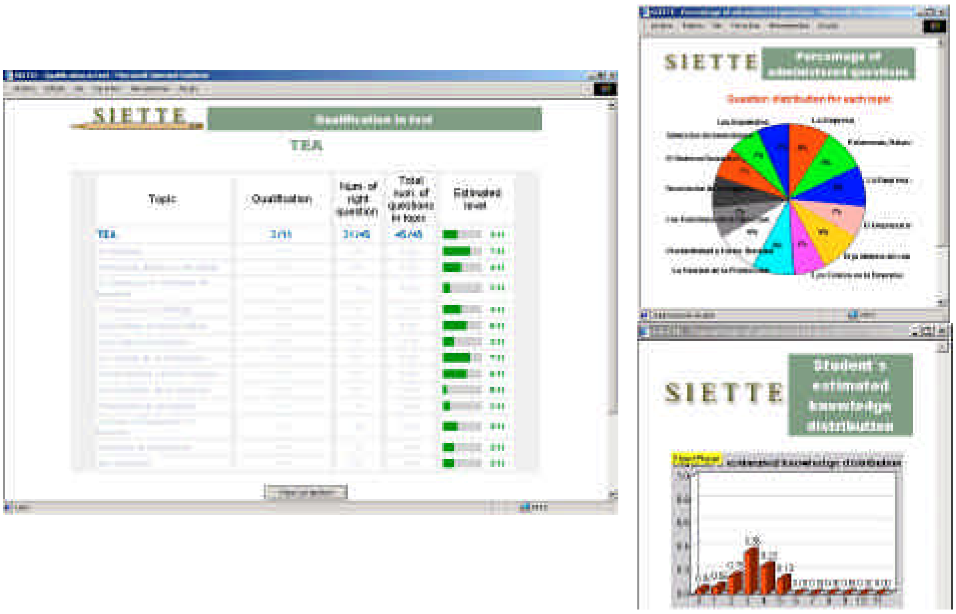


Fig. 5. Final qualification in the global test.

4 Integrating *SIGUE* and *SIETTE*

SIGUE and *SIETTE* have been integrated (it can be accesses through *SIGUE*, http://www.lcc.uma.es/SIGUE, selecting the *TEA* course) to make the access to the new adaptive version of TEA easy and homogeneous. As a result, students do not have to use separately both systems in order to have an adaptive instruction about Agrarian Economy.

In the navigation process *SIGUE* recommends the students which are the best pages to visit. .After each topic, the system will recommend to take a test in *SIETTE* to evaluate the proficiency in the topic. This test is also adaptive and is used with tutorial goals. However, at the end of the test the user has a feedback that will encourage him to continue with next topic or not.

Once the student has visited all the pages of the tutorial, the system will propose him to take a test of the whole subject. Through this test, students can check if they have assimilated correctly all the notions explained in the tutorial.

One of the advantages of the new design compared to the existing static course is that it can be easily updated. New content can be added, not only from *TEA* pages, but also from other pages related to the subject.

Figure 6 shows a hardcopy of a screen of both systems together. *SIGUE* has reach to a point where a test is recommended (see left frame). *SIETTE* has been invoked and the test is being done (right frame)

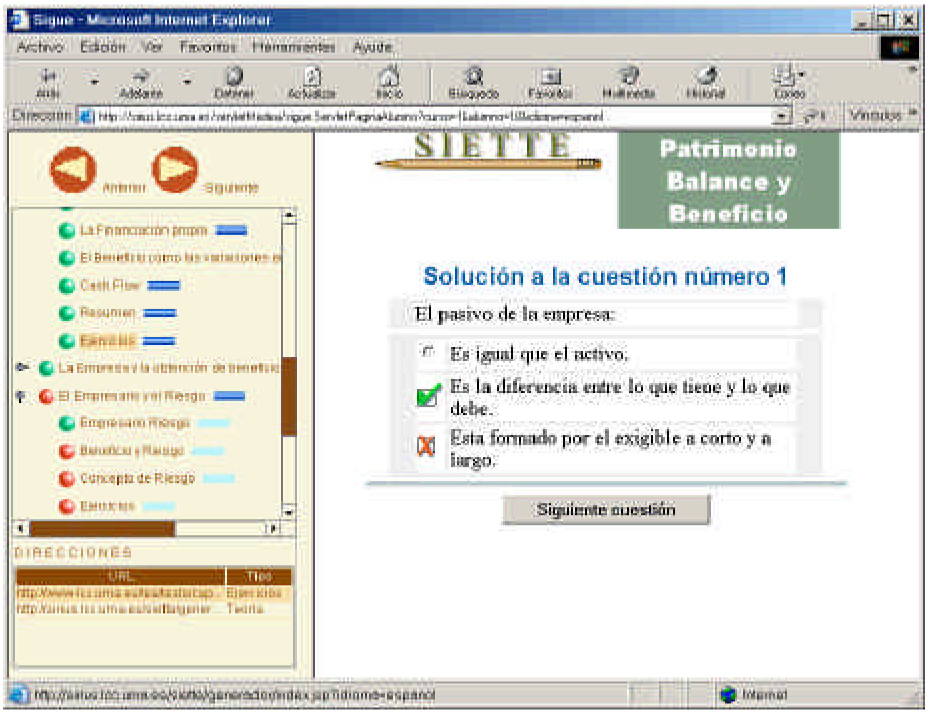


Fig. 6. Integration of *SIETTE* and *SIGUE*.

5 Conclusions

TEA was conceived as a Computer Aided Educational System for teaching Agrarian Economy. Even though it was initially designed to overcome the classical lacks of hypertext education, by using graphical and textual clues to improve the user orientation, the navigation through the course could not be controlled.

This educational system has been made adaptive. By using *SIGUE*, the navigation process of student is guided. The system annotates as the student goes through the hypertext the nodes visited and the estimated progress that the student should have at that point. This annotation is also used to guide the student upon request.

A set of adaptive tests have been include in TEA through *SIETTE*. There is a test for each topic, which allows students to evaluate their knowledge about that topic. In these tests, each item is selected in terms of the current estimated student's knowledge level.

This work has mainly supposed the integration of two web-based adaptive systems. One system to guide the student in the learning of concepts, and other system to evaluate his proficiency in these concepts. By means of this integration, students can access to both functionalities through the same web system. The student does not need to know that he is accessing two different systems. This makes easier the instruction process.

The drawback of the system is that there is no deeply integration between *SIGUE* estimations of the student's knowledge level and *SIETTE* assessment of student's knowledge level. *SIGUE* student model is basically a model of pages visited, and if the test is done, it assumes that the corresponding instruction has been completed.

SIGUE and *SIETTE* can also be integrated as components of *MEDEA* [12]. *MEDEA* is an open system to develop Intelligent Tutoring Systems. It is composed of independent educative modules coordinates by a core that controls the instruction of the student. *MEDEA* provide a communication protocol with their components that allows a deeper integration.

References

1. Belmonte, M., Berbel, J., Conejo, R. (1996). TEA: An Agrarian Economy Instructor System. *Third International Conference on Computer Aided Learning and Instruction in Science and Engineering, CALISCE'96*, San Sebastián. 322-330.
2. Ríos, A., Millán, E., Trella, M., Pérez-de-la-Cruz, J.L., Conejo, R. (1999) Internet Based Evaluation System. *Artificial Intelligence in Education:*, IOS Press, Amsterdam. 387-394.
3. Wainer, H. (ed.). (1990). *Computerized Adaptive Testing: a Primer*. Hillsdale, NJ: Lawrence Erlbaum Associates.
4. Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. Hillsdale, NJ: Lawrence Erlbaum Associates.
5. Birnbaum, A. (1968). *Some latent trait models and their use in inferring an examinee's mental ability*. In Lord, F. M. & Novick, M.R. (ed.) *Statistical theories of mental test scores*. Reading, MA: Addison-Wesley.
6. Owen, R. J. (1975). A Bayesian sequential procedures for quantal response in the context of adaptive mental testing. *Journal of the American Statistical Association 70*, 351-356.
7. Conejo, R., Millán, E., Pérez-de-la-Cruz, J. L. & Trella, M. (2000). An Empirical approach to on-line learning in SIETTE. In *Proceedings of the ITS 2000*, Montreal. Springer-Verlag.
8. Brusilovsky, P., Schwarz, E., & Weber, G.(1996). ELM-ART: An intelligent tutoring system on World Wide Web. In Frasson, C., Gauthier, G., & Lesgold, A. (Eds.), *Intelligent Tutoring Systems*, Springer Verlag, Berlin (1996) 261-269.
9. Brusilovsky, P. (1998). *A Study of User Model Based Link Annotation in Educational Hypermedia, Journal of Universal Computer Science*, vol4, n-4,429-448
10. Weber, G., & Specht, M. (1997), User modeling and adaptive navigation support in WWW-based tutoring systems. In Jameson, A., Paris,C., & Tasso, C. (Eds.), *User Modeling*, Springer-Verlag, Wien 289-300.
11. Guzman,E. Conejo, R. :Simultaneous evaluation of multiple topics in SIETTE. En *Procdeedings ITS2002*. Springer-Verlag (2002) (to appear)
12. Trella, M., Conejo, R., Bueno, D., Guzmán, E.: (2002). An autonomous component architecture to develop WWW-ITS. In Brusilovsky, P., Henze, N., Millán, E. (eds.) *Proceedings of the Workshops on Adaptive Systems for Web-Based Education.* Malaga (this volume).

An autonomous component architecture to develop WWW-ITS

Trella, M., Conejo, R., Bueno, D., Guzmán, E.

Departamento de Lenguajes y Ciencias de la Computación
Escuela Técnica Superior de Ingenieros en Informática.
Universidad de Málaga
Campus de Teatinos
Boulevard Louis Pasteur, s/n. 29071
Málaga. SPAIN
Telephone: +34 952137152
Fax: +34 952131397
{trella, conejo, bueno, guzman}@lcc.uma.es

Abstract. In this paper we describe an on-going research work (MEDEA project) whose final goal is to develop a general framework to build open ITS. We understand “open system” as a set of autonomous educational modules that communicate between themselves following high-level pre-established protocols. Each of those modules can be an intelligent component with its own instruction strategies or a support tool that leaves all the adaptive capabilities to the ITS instructor core. The architecture opens up the possibility to include new components, provided that they are able to interact within the general framework.

1 Introduction

The Web has become a great source of information and resources that are available for tutoring. The advantages of adaptive web based tutoring are well known [1]. The intrinsic structure of the web is distributed, and in most cases it is used just as a huge repository of unorganized information. Teachers collect different URLs to give their students links where they can find lecture notes, exercises, simulation programs, etc. related to the matter that they are studying. Commonly, the systems that they use have been developed for different objectives, and have been designed by different people with different approaches. Some of them might be adaptive but generally non-adaptive materials are found. Some attempts have been made to integrate two preexisting web based adaptive systems, (see for instance [2]). This paper presents a proposal of a framework for systems integration.

One of the main task of a human teacher when using the web as a teaching resource is to redirect the students to those systems that are more appropriate according to the student profile and supervise their progress. If the web pages are fixed, an intelligent behavior is still needed from the teacher, suggesting the students to skip some parts, conducting them through pages with exercises, simulators, etc, according to the specific goal of the course. On the other hand, if the web systems used have some kind of adaptive features or intelligent behavior, it should be advisable to take advantage of these features.

Following this idea, a web ITS might be constructed by gathering pages or systems, like *building blocks* in the sense of Chandrasekaran [3], some of them might be new, and some others reused from existing material. Most technologies used for Web ITS development are designed for small systems and they do not have the necessary resources to guarantee a high-level conceptual organization. This kind of problems has been traditionally treated with software engineering techniques, and more recently with knowledge engineering techniques, following the Newell's *knowledge level* paradigm [4]. Using this idea, high-level methodologies and tools for the intelligent systems development have been proposed as KADS [5], PROTÉGÉ [2], KSM [3], etc.

In this paper we describe the MEDEA system. MEDEA is a Spanish acronym of *Methodology and Tools for the Development of Intelligent Environments of Teaching and Learning*. MEDEA is not a new ITS authoring tool but a general framework for open systems development and integration [8] [9]. We understand "open system" as a set of autonomous educational components that communicate between themselves following high-level pre-established protocols. In the case of web-based open tutorial systems, this protocol is the well known HTTP. MEDEA also adds a new high level layer to communicate intelligent tutorial components, that is converted to HTTP and might be interpreted (or not) by the certain educational components.

The components used in MEDEA can be intelligent, with its own instruction strategies, or not, leaving all the adaptive capabilities to the ITS instructor core. The architecture opens up the possibility to include general-purpose components that are able to interact within the general framework. MEDEA and offers to the educators a generic environment to develop Web ITS without the limitations of a close set of utilities as in the most of the authoring systems [10].

2 MEDEA Architecture

The elements that compose MEDEA architecture can be classified in three main groups: those that contain knowledge (*knowledge modules*), those that use this knowledge for making adequate decisions along the instruction (*functional modules*) and those that serve to access and configure the system (*tools*). The base of MEDEA architecture is a core that plans the instruction based on a set of external tutorial components that are introduced in the system. The domain and pedagogical knowledge is distributed between the core of MEDEA that serves as a master index, and these components.

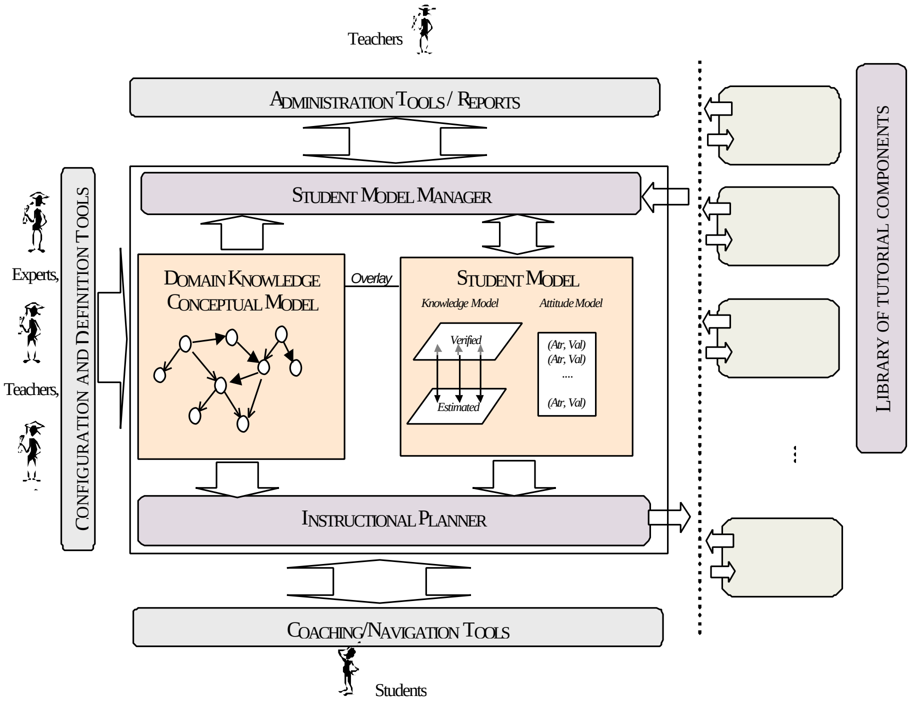


Fig. 1. MEDEA architecture

We could say, comparing the MEDEA architecture and the traditional ITS architecture, that *what* (knowledge about the learning subject) is explicitly represented by the domain model, *who* (knowledge about the student) is given by the student models (knowledge and attitudes) and *how* (teaching knowledge) is divided between the instructional planner, tutorial components and domain model. All knowledge models are represented in XML for computational use. Figure 1 shows the structure of MEDEA modules:

Knowledge modules:

? **Conceptual Knowledge Domain Model**. This module contains knowledge about the subject to be taught. Both domain concepts and relationships among them are represented. The domain model will be explained better in the next section.

? **Student model**. This module is decomposed in the *Knowledge Student Model*, and the *Attitudes Student Model*. The former contains information about the student's state of knowledge during the learning process. Not all the evidence generated during student's interaction with the system will be processed in the same way. If some kind of evaluation is performed, the results will be used to update the *verified student knowledge model*, and it will replace the *estimated* *student knowledge model*. If there is no evaluation the system estimates the students knowledge increase and actualized the *estimated student knowledge model*. On the other hand, the *Attitude Student Knowledge* contains those features that describe the student profile but are not related to his current state of knowledge, like motivation, style of learning, speed of web connection, etc. The student model is described in section 4.

Functional modules:

? **Instructional planner**. This module will provide students with the necessary guidance during the learning process. It will design and compose the tutorial sessions, that is, it will decide in each moment the adequate task to be performed by the student. To this end, the information that will be used is a) conceptual domain knowledge, b) student state of knowledge, c) student profile and d) tutorial components definitions.

The tutoring systems generated with our tool will allow students to freely navigate. The system only recommends but not imposes the next student action. The planner will also have the capabilities to justify the recommended actions, so the student has the necessary information before deciding to accept or not the suggestions made by the system.

? **Student Model Manager**. The function of this module is to create and update the student model. Once a tutorial component is executed, the student model is updated. Some tutorial components (for example a component that poses a test or an exercise) can evaluate the student knowledge about some domain concepts or subjects. The information provided by these components goes to the *Verified Student Knowledge Model.* MEDEA assumes the existence of other components that are not be able to determine exactly the variations of the student knowledge level, for example a component that display a text to be read or presents a simulation. There might be also components that do not even return any feedback to MEDEA, for example a static web page. MEDEA is just informed that the student has visited those components. This information is treated differently using the *Estimated Student Knowledge Model.*

? **The library of tutorial components**. They are external educational tools, each of which makes a concrete task (electronic books, simulation tools, exercises about the subject, making tests, etc.). From MEDEA point of view, the architecture of a component (Fig.2.) is composed by a *partial domain model*, a *development interface* to introduce contents, a *student interface*, a *temporal student model*, a *component functional description* (where the tasks of the teaching component and the way of communicating with MEDEA planner are defined) and a *control* that is the execution engine of the component.

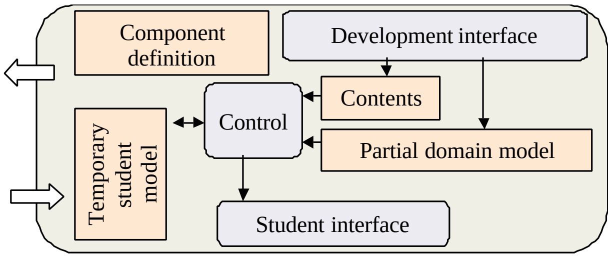


Fig. 2. Tutorial component architecture

Tools (Interfaces)

? **Configuration and definition tools**. This module will be used by domain experts, teachers and designers. They will be able to introduce the contents, define and configure the data and knowledge modules using specific interfaces.

? **Administration tools**. This module will be used by teachers to monitor the evolution of their students. It will show the progress of each student, the statistics about the course use, and average student performances, and other secretariat and administrative tasks.

? **Navigation tools**. This module is used by student to support their navigation and interaction with the whole system. It can be conceptualized as an advisor during the learning and designed as an additional bar in the browser. It is currently implemented as an additional frame of the web browser.

To describe the behavior of MEDEA we can use the simile of a school personal tutor. Suppose that a student asks help to his personal tutor (*instructional planner*) to study a subject (*domain model*). The tutor examines his academic expedient (*student model*) and taking into account other student's features (*student attitude model*), selects the topic and the style of teaching that is better for the student. After that, he consults the school staff (*tutorial components library*) and chooses the better teacher for that specific topic according to the current student's profile. The tutor must know the teaching methods of all the teachers available (i.e.: this usually does a lot of exercises, that likes better to explain theory, this wants the students makes exercises on the blackboard, etc.). The tutor send the student to the teacher selected (*tutorial component*) with a message (*communication protocol*) indicating the topic that the student must study, and may be other notices (i.e: this is the third time this student tries this topic, this student has low learning rate). This teacher will apply his effort and expertise to improve the student knowledge in the topic proposed (*intelligent component*), or he may just gives the student a good text to read (*non-intelligent component*) without asking what he is doing. When the teacher ends the class he sends back a message to the tutor explaining the students behavior (*evaluation feedback*), or the student simply returns to the tutor with no message. With this new information, the tutor (now *the student model manager*) actualizes the student expedient. This process will be repeated until the student completes the subject.

3 Domain model

The domain model represents the knowledge about the subject to be learned. Different approaches exist [11] depending of the nature of the domain to be represented. In MEDEA we have focused on declarative domains representation. The most extended models for representing this kind of domains are the semantic networks of knowledge units, used in systems as DGC [12], Eon [13] and IRIS [14].

From a conceptual point of view, the domain is defined by a) a semantic network of concepts and relations among them and b) pedagogical knowledge needed for the instruction.

From the implementation point of view, the basic elements of MEDEA domain model are a) the *concepts*, b) the *relations* between concepts, and c) the *evaluation types* used for representing the degree of knowledge.

a) *Concepts* are the basic pieces in which the subject is divided for teaching purposes. During the task of modeling a knowledge area for an ITS the pedagogical purpose of the system should be taken into account. First of all, in order to choose the net granularity, that is, it has to be decided when a knowledge unit must be discomposed in simpler units. As Anderson [11] says, sometimes to model accurately a domain requires a computational charge that it is not necessary from the tutorial point of view. Second, it is necessary to include pedagogical knowledge in the domain model to guide to the student through the knowledge units [13]. This pedagogical information is included as attributes associated to the concepts, for instance threshold marks that represents the minimum mark needed for considering the concept learned.

b) MEDEA uses a set of binary *relations* between concepts that can be used to describe the domain. These relations are: *prerequisite*, *part of*, *is a*, *belongs to*, *is useful to understand*, *is similar to* and *is opposite to*. Each relation should define an acyclic graph of concepts that is used by the functional modules to guide the instruction (*the instructional planner)* and make inferences of the student knowledge (*the student model manager*). The actual semantic of the relation is so fixed at the functional modules. However, MEDEA includes an informal description of the semantic of the relations to guide authors while creating these graphs. Currently, the domain model support all these relations, but only *prerequisite*, and part *of*, are used by the *instructional planer*. Other relations can be defined by course authors and stored for future use. The *student model manager* is still under development.

c) As we will see in next section, the knowledge student model in MEDEA is based on the overlay technique. Each concept will have a magnitude associated that represent the student degree of knowledge, the *evaluation types* in MEDEA are the types of those magnitudes. They are not fixed. but defined when the course is created among a set of *internal types* supported by the architecture. The internal types are: *enumerated* (i.e. the knowledge level about a concept can be A, B, C or D), *real* (a real number) and *distribution* (i.e. the knowledge level can be {A/0.2, B/0.3, C/0.4, D/0.1}, indicating that the probability that the student knowledge level in that concept be A is 0.2 and so on). MEDEA is a general framework that can use different components for its instructional purposes. Each of these components, can have its own internal representation of the student knowledge. Implicit conversion between internal types has been defined to support compatibility between different components.

Figure 3 shows a fragment of an XML file that contains the domain model of a course of Logic. At the beginning, the evaluation types used to evaluate each domain concept are defined. The second part is a list of concepts that includes some pedagogical information needed for the instruction as the difficulty level of each concept. At the end there is a list with the relations among concepts.

```
<!DOCTYPE DOMAIN_MODEL SYSTEM "http://sirius.lcc.uma.es/medea/dtd/DOMAIN_MODEL.dtd">
<DOMAIN_MODEL id="domain_model01" name="Logic of proposals">
  <EVALUATION_TYPES>
        <EVALENUM id="EvalEnum" default_minimum_mark="Passed">
              <ENUMERATED id="Passed"/>
              <ENUMERATED id="Failed"/>
        </EVALENUM>
        <EVALREAL id="EvalReal" lower_boundary="0" upper_boundary="10"default_minimum_mark="5"/>
  </EVALUATION_TYPES>

  <CONCEPTS>
        <CONCEPT id="t1" idref_evaluation="EvalEnum" name="Introduction" difficulty="Low"/>
        <CONCEPT id="t2" idref_evaluation="EvalEnum" name="Formal sintax" difficulty="Low"/>
        <CONCEPT id="t3" idref_evaluation="EvalEnum" name="Semantic" difficulty="Low"/>
        […]
        <CONCEPT id="c32" idref_evaluation="EvalReal" name="CDN" difficulty="High"/>
  </CONCEPTS>

  <RELATIONS>
        <RELATION id="r1" id_origin_concept="c21" id_destiny_concept="c1" type="prerequisite"/>
        <RELATION id="r4" id_origin_concept="c4" id_destiny_concept="c2" type="is_a "/>
        […]
        <RELATION id="r73" id_origin_concept="c31" id_destiny_concept="t6" type="belongs_to"/>
        <RELATION id="r74" id_origin_concept="c32" id_destiny_concept="t6" type="belongs_to "/>
  </RELATIONS>
</DOMAIN_MODEL>
```

Fig. 3. Representation of the domain model

4 Student model

The *Student Model* in MEDEA is divided in two main subcomponent. The *Student Knowledge Model* and the *Student Attitude Model*. The first represent what the student knows about the subject, the second represent other features of the student.

The *Student Knowledge Model* is an overlay model divided in two levels: *estimated model* and *verified model*. Each level is a list of *concept/mark* pairs. As we explain in section 2, in the *verified model* the concept marks have been obtained using some evaluation method. On the other hand the *estimated model* collects the indirect information and inferences of the student knowledge degree, calculated based on the student behavior during the instruction. This multilayer approach has already been used by other authors like Brusilovsky, in the last versions of ELM-ART-II [15], that uses different layer for concepts visited, evaluated, inferred, etc. At these moment we have only proposed two layer, grouping in the estimated layer all the uncertain information and inferences.

Figure 4 shows an example of the student knowledge model represented in XML. Each concept has a value associated according to the evaluation type defined in the domain model. The last concept that has been taught is also stored.

```
<!DOCTYPE STUDENT_MODEL SYSTEM "http://sirius.lcc.uma.es/medea/dtd/STUDENT_MODEL.dtd">
<STUDENT_MODEL courseid="d03" lastconcept="c5" studentid="s432">
 <ESTIMATED_STUDENT_MODEL>
     <CONCEPT id="c1" value="VERY WELL"/>
     <CONCEPT id="c2" value="VERY BAD"/>
     [...]
     <CONCEPT id="c5" value="WELL"/>
     <CONCEPT id="c6" value="VERY BAD"/>
 </ESTIMATED_STUDENT_MODEL>
 <CHECKED_STUDENT_MODEL>
     <CONCEPT id="c1" value="VERY BAD"/>
     <CONCEPT id="c3" value="REGULAR"/>
     <CONCEPT id="c5" value="REGULAR"/>
     <CONCEPT id="c6" value="VERY BAD"/>
 </CHECKED_STUDENT_MODEL>
</STUDENT_MODEL>
```

Fig. 4. Representation of the student knowledge model

There are also some relevant student's features that are important for the learning process. In MEDEA we have identified some of them that have been included in the *Student Attitude Model*.

Feature	Values
Cognitive development (formalization and abstract concepts understanding skills)	***High, Medium, Low***
Motivation	***High, Medium, Low***
Learning style	***Theory, Exercises***
Time dedicated to the subject study. It represents the student effort degree to pass the subject.	***Time in minutes***
Progress. It represents the student learning speed.	***Bad, Regular, Good***
Experience with computers.	***Very much, Some, Little, None***
Internet connection speed.	***High, Medium, Low***

This model is used by teachers or course designers to establish relations between a concrete student profile and some instruction parameters. For example, a teacher can specify in the course definition that when a student with *low motivation* level does a test it is better that he sees the right answer each time he makes a question than he sees all the right answers at the end of the test.

5 Instructional planner

The instructional planner is the core of MEDEA architecture. It is the module in charge of sequencing the domain and adapting the instruction process to each student. In order to do this task we observed human teacher behavior. Usually a teacher takes decisions in several levels. First he decides the instruction goals (concept to be taught) and then he takes the other decisions as the topics content, material to be used, which pedagogical strategy is going to be used, etc.).

There are several examples in the ITS literature in which the planner task is divided in subtasks: some oriented to concept selection and others to decide how to teach the concept selected [14] [16] [17].

The MEDEA planner takes three main decisions: 1) Does the student need to be evaluated?, 2) if YES: about which topic will he be evaluated?, if NO: which concept has the student to learn now? and 3) How will he be evaluated or taught better?. The knowledge needed to answer these questions is in the domain model, the student models and the pedagogical elements of the system (domain model pedagogical contents and tutorial components).

The modular structure of the MEDEA architecture and the separation between knowledge representation and knowledge use allows that this module could be also easily changed. The final goal of this project is to have a planner library in the system. For the first prototype a heuristic planner has been implemented. In this planner, the task of selecting a concept to be evaluated or learned will be carried on in two phases: first the planner selects an ordered set of candidate concepts. This selection is done taking into account the relations *prerequisite* and *part of* and the current student model state. Second, the planner selects the first concept of the candidates set.

The criterions to decide if a concept is a candidate or not are:

1. The student has not reached the minimum required by the teacher to pass the concept.
2. The concept is *prerequisite* of any other concept that cannot be learned because the student has not passed this one.
3. The concept is *part of* any other concept that cannot be learned because the student has not passed this one.

A weight is assigned to each candidate. The most weighted are those that are selected by the criterion 2.

As we have said before, the other important task of the planner is to decide how the student will be evaluated or taught better. Our implemented planner takes this decision using the teacher’s criterion. That is, when a teacher designs a course, he links each tutorial component registered in the system to a student profile. The planner only has to consult the attitude student model and assigns him the more adequate tutorial component according to the teacher.

6 Tutorial components

A tutorial component is an external educational tool that is able to complete a tutorial task as posing tests, presenting theory contents in hypertext, posing a game, etc.

The pedagogical knowledge in MEDEA system is distributed among several modules: the domain model, the instructional planner and the tutorial components although it is not strictly necessary for the instructional process that a component provide any knowledge. Most of the instruction tasks fall on the pedagogical system core composed by the domain model and the planner. A tutorial component can complement this task providing its tutorial strategies and a control more exhaustive over student's actions.

MEDEA classifies tutorial components as *evaluation components* or *information components*. The difference is that components of the former type are able to evaluate the student knowledge level about a concept.

The problem of the components integration in the system is very important in the development of MEDEA project. This problem can be approached as the communication between two software systems. MEDEA and tutorial components are Web systems so the communication can be established through URLs. In order to do its work, the planner needs from the component low-level knowledge about its performance (call format, parameters, etc.) and high-level knowledge about its pedagogical offer (tutorial strategies, options, user options, etc.).

Communication with a component can be established through two different interfaces: from teacher interface for course creation and from student interface for instruction session execution.

Now we are going to describe the different types of tutorial component's actions that MEDEA users (teachers and students) can request:

1. *AUTHOR*. Contents creation (tests, HTML pages, exercises, etc.). In order to do that the teacher would access the component author tool.
2. *NEWCOURSE*. If a course about a subject is already defined in the component, the teacher only has to establish the correspondence between component domain concepts and MEDEA domain concepts.
3. *NEWUSER*. If a course about a subject is already defined in the component and there are some students registered, the teacher must communicate to MEDEA system the existence of these students.
4. *EXECUTE*. Execution of a session with a component.

The component creator must specify, for each of these actions, the call and return URL formats. Not all components used by MEDEA must have all of these actions. The only compulsory action is the last one that should exist to call this component for a tutorial session. The component interface specifications are also defined in an XML file.

A library of general-purpose web based web-based authoring tools has been implemented. These authoring tools are fully compliant with the actions defined in MEDEA, but can also be used as stand-alone applications. A web based tutorial system can be constructing by developing specific components using these tools, that are also called *general-purpose components*. Currently, the general-purpose components predefined in MEDEA are:

? HERMES, which is an authoring tool to create web based electronic books. It is mainly an HTML editor that makes easier for non-programmers the creation of a set of structured web pages. The development tool is based on page templates. The result from HERMES are a set of static web pages structured hierarchically.

? SIGUE, which is a systems that allows the construction of a web course by collecting the references to existing web pages. SIGUE adds some adaptive capabilities, like adaptive links hidden, and/or reinforcement. SIGUE has his own student model that is actualized as the student goes through the course content. [18]

? SIETTE, which is an adaptive test generation system. It has a large set of item types and different ways of evaluation. [19]

The general idea is that predefined general-purpose components will works as plugs-in. Once defined they can be invoked from the MEDEA core, either for authoring or for course taking. The authoring tools of these general components is linked to the development tools of MEDEA, (action *AUTHOR*), the administration tools, are linked to the administration tools of MEDEA, (actions *NEWUSER* and *NEWCOURSE*) and the course presentation is linked to the student interface (action *EXECUTE*).

7 Conclusions

As a consequence of the increasing importance of the distance education and the advance of this field due to the new information technologies, many researchers have seen the needed of applying intelligent techniques of existing educational systems to the Web.

MEDEA is a proposal of an open architecture for Web ITS development. MEDEA is an open system that contains the traditional modules of an ITS architecture that has been designed to allow the integration and reutilization of educational tools and teaching material already developed. The idea of MEDEA is that any teacher could develop a course reusing his own material and software.

MEDEA is still under development, this presentation is just a first glance of the system. No testing has been carried out yet because it is not fully operational.

References

1. Eklund, J.; Brusilovsky, P.: The Value of Adaptivity in Hypermedia Learning Environments: A Short Review of Empirical Evidence , In *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia at Ninth ACM International Hypertext Conference, Hypertext'98*. (1998). Available on line at http://wwwis.win.tue.nl/ah98/Eklund.html
2. Brusilovsky, P.; Ritter, S.; Schwarz, E.: Distributed intelligent tutoring on the Web. In In Brusilovsky, P., Nakabayashi, K., Ritter, S. (eds.) *Proceedings of the Workshop 'Intelligent Educational Systems on the World Wide Web' at AI-ED'97*, 8th World Conference on Artificial Intelligence in Education (1997). Available on line at http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Brusilovsky/Brusilov sky.html
3. Chandrasekaran, B.: Generic Task in Knowledge Based Reasoning: High level building blocks for expert systems design. *IEEE Expert*, 1, (1986) 23-29. Reimpreso en Buchanan,B.G. y Wilkins,D.C. (eds.) *Reading in Knowledge acquisition and learning*. Morgan Kaufmann. San Mateo. CA. (1993) 170-177
4. Newell, A. The Knowledge level. Artificial Intelligence. (1982); 18.

5. Wielinga, B.J., Schreiber, A.T., y Breuker, J.A. : KADS: a modeling approach to knowledge engineering. *Knowledge Acquisition* 4, (1992). Reimpreso en Buchanan, B.G. y Wilkins, D.C. (eds.) *Readings in Knowledge Acquisition and Learning*. Morgan Kauffman. San Mateo. CA (1993) 92-116
6. Musen, M. A. Automated Support for Building and Extending Expert Models. Machine Learning. (1989); 4:347-375.
7. Cuena, J. and Molina, M. KSM: An Environment for Knowledge Oriented Design of Applications Using Structured Knowledge Architectures. Applications and impacts. IFIP'94; (1994).
8. Self, J. (1999), Open Sesame?: Fifteen Variations on the Theme of Openness in Learning Environments. Keynote speaker at AI-ED'99. Abstract published in Lajoie, S., Vivet, M. (eds.) *Artificial Intelligent in Education*: IOS Press (1999).
9. Murray, Tom: A Model for Distributes Curriculum on the World Wide Web. In *Journal of Interactive Media in Education*. (1998),5
10. Murray, Tom , Authoring Intelligent Tutoring Systems: an analysis of the state of the art. In *Intl. J. of Artificial Intelligence in Education*. (1999), 10, 98-129
11. Anderson, J. R. The Expert Module. In Polson, M.C., Richardson, J.J. (eds.) *Foundations of Intelligent Tutoring Sytems*. Lawrence Elrbaum; (1988).
12. Vassileva, J. (1997), Dynamic Course Generation on the WWW. In Brusilovsky, P., Nakabayashi, K., Ritter, S. (eds.) *Proceedings of the Workshop 'Intelligent Educational Systems on the World Wide Web' at AI-ED'97*, 8th World Conference on Artificial Intelligence in Education (1997). Available on line at: http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Vassileva/Vassileva.html
13. Murray, Tom (1998), Authoring Knowledge-Based Tutors: Tools for Content, Instructional Strategy, Student Model, and Interface Design. In The Journal of Learning Sciences. V. 7, N.1, pp. 5-64
14. Arruarte, A., Fernández-de-Castro, I., Ferrero, B., and Greer, J. (1997), The IRIS shell: How to build ITSs from pedagogical and design requisites. In International Journal of Artificial Intelligence in Education. V. 8, N.3-4, pp. 341-381
15. Weber, G., & Specht, M.: "User modeling and adaptive navigation support in WWW-based tutoring systems"; In Jameson, A., Paris,C., & Tasso, C. (Eds.), User Modeling, Springer-Verlag, Wien (1997) 289-300. http://citeseer.nj.nec.com/weber97user.html
16. Woo, Chong Woo (1991), Instructional Planning in an Intelligent Tutoring System: combining global lesson plans with local discourse control. PhD. Thesis.
17. Woolf, B. and McDonald (1987), D.Context-Dependent Transition in Tutoring Discourse. In Proceedings of AAAI.
18. Carmona, C., Bueno, D., Guzman, E. Conejo, R.: SIGUE: Making Web Courses Adaptive In De Bra, P, Brusilovsky P., Conejo, R*.: Proceedings of the AH2002 Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Springer-Verlag LNCS 2347 (2002).
19. Ríos, A., Millán, E., Trella, M., Pérez-de-la-Cruz, J. L., & Conejo, R. (1999). *Internet Based Evaluation System*. In Proceedings of the 9th World Conference of Artificial Intelligence and Education AIED'99. Amsterdam: IOS Press.

Automatic Generation of a Navigation Structure for Adaptive Web-Based Instruction

Judith Masthoff[1]

University of Brighton, UK
Judith.Masthoff@brighton.ac.uk

Abstract. A new approach to adaptive web-based instruction is outlined and advocated, which is based on models and principles that allow the course material to organize itself. This is particularly useful when course material originating from multiple authors is to be combined into a coherent whole.

1 Background

In previous research, an "artificial teacher" has been developed, which adapts instruction to the learner [1]. This artificial teacher needs an abstract description of the course material (often called metadata). Additionally, there needs to be sufficient variation in the course material to be able to reach a high level of adaptation. We are developing a system ('Authoring Coach') that coaches authors to provide both the metadata and the variation needed for the adaptation to be effective. It shall

- Provide an easy user-interface for entering course material and metadata.
- Stimulate authors to enter metadata by clarifying its purpose and consequences.
- Stimulate authors to provide variation by indicating the amount of adaptation possible with the current material and how this can be increased.
- Enable multiple authors to contribute without need for coordination. An essential aspect of the World Wide Web has been that is has organically grown: authors from around the world (without coordination) have contributed material. We use the same principle for the authoring of courseware.
- Generate personalized web-based lesson books from the material provided by the authors, which are easy to use, have a good narrative flow, and allow the artificial teacher to monitor and optimally support student learning.

This paper focuses on the last two points, and expands [8] by describing the generation of a navigation structure in more detail.

[1] The author is supported by Nuffield grant NAL/00258/G.

2 Existing Systems

To model the teaching domain, both InterBook [2] and NetCoach [3] use concepts, which are organized in a network, with links reflecting different types of relationships between them. Concepts in InterBook are "elementary pieces of knowledge for the given domain". All examples mentioned are noun phrases, like "production rule". In NetCoach concepts are "internal representations of pages", like "Chapter-2-1-2". In InterBook, the author provides an electronic textbook that is hierarchically organized (chapters, sections, etc). Each page has a set of *outcome concepts* and a set of *prerequisite concepts* associated with it (analogous to pre- and post-conditions in programming). These are used to support adaptive navigation and hyperlink annotation. NetCoach also uses prerequisites, but as relations between concepts. In MetaLinks [4], authors provide the hierarchical relationship between pages.

In authoring tools like InterBook, MetaLinks, and NetCoach, authors explicitly provide the textbook hierarchy (like page1 has subsection page1-1). In contrast, Authoring Coach will *generate* a hierarchy on the basis of the concept network and the prerequisites and outcomes associated with pages (and some additional information, see below). In a sense, it is related to the Microcosm system [10], which automatically inserts links into pages, but it concentrates more on producing the hierarchy of an educational textbook.

3 Generation of a navigation hierarchy

3.1 How to get a chapter, section, subsection like structure

We would like to generate a chapter, section, subsection like structure. The reason for adopting this kind of structure is that it is easy to learn, and has successfully been used in paper-based books. The system could in principle use five sources of information to base the structure on:

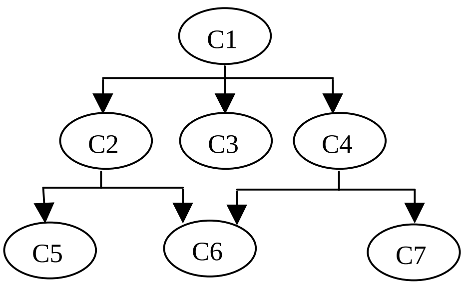
1. The *concept network* can be regarded a domain ontology. It provides information about which concepts are 'children' of which other concepts.
2. *Outcomes* describe what a student will be able to do after successfully studying a page (note: page can be explanation plus exercises). The use of concepts to express outcomes, as in InterBook, is not enough to make the outcomes sufficiently precise and unambiguous. It can result in multiple pages in an electronic textbook that cover apparently the same outcome. This reduces the opportunities for adaptive navigation support and makes it hard to automatically generate navigation structures. As advocated in [8], we express each outcome as a combination of an action verb with a concept. For instance, an outcome of a page will not be "search methods", but something like "explain search methods" or "implement search methods". This will allow authors to specify the outcomes of a page more accurately. As argued in [8], a page title will be generated from the concepts and verbs that describe the page's outcomes.

3. *Prerequisites* describe what a student should be able to do, before starting to study a page. Like outcomes, they are expressed as a combination of an action verb with a concept.
4. Other descriptive information about the pages, like author etc. (See Section 3.2.2).
5. The content of the pages, i.e., the text, images, video, etc they contain.

To generate an initial version of the structure, only the concept network and outcome information is used. The other information will be used to refine the structure, as described in sections below.

Each concept is treated as a section in the electronic book. If a concept is a child of another concept, then it will be included as a subsection. Note that this assumes no cycles in the concept network. Each section contains the pages that use the section's concept in their outcomes. In the navigation structure, the link to each page is its title, which (as in [8]) is generated on the basis of the verbs and concepts used in its outcomes.

For example, suppose a concept network, and pages with outcome and prerequisite relations as given in Figure 1. Figure 2 shows the navigation structure resulting from including the concepts (left), and their associated pages (right).



Page		**Verb**	**Concept**
P1	outcome	V1	C1
P2	outcome	V3	C1
P3	outcome	V1	C1
P4	outcome	V2	C6
	prerequisite	V3	C7
P5	outcome	V3	C7
P6	outcome	V1	C6
	prerequisite	V2	C6
P7	outcome	V4	C5
	prerequisite	V1	C1

Fig. 1. Example of a concept network and pages associated with concepts via outcomes and prerequisites. Cs denote concepts, Ps pages, Vs verbs

In the example of Figure 1, all pages have exactly one outcome, and at most one prerequisite. This does not necessarily have to be the case. A page can have multiple outcomes, and multiple prerequisites. Suppose a page has two outcomes: V8 C3 and V9 C5. In that case, the page will appear twice in the hierarchy, once below C3 and once below C5. The page title in that case would be generated from both V8 C3 and V9 C5. In theory, this approach may lead to many occurrences of the same page in the hierarchy, and very long page titles. However, our Authoring Coach will encourage authors to keep pages short (preferably, no scrolling should be necessary), and focussed (preferably, only covering one outcome). Therefore, we expect pages with multiple outcomes to be exceptions, and if a page has multiple outcomes that number to be low.

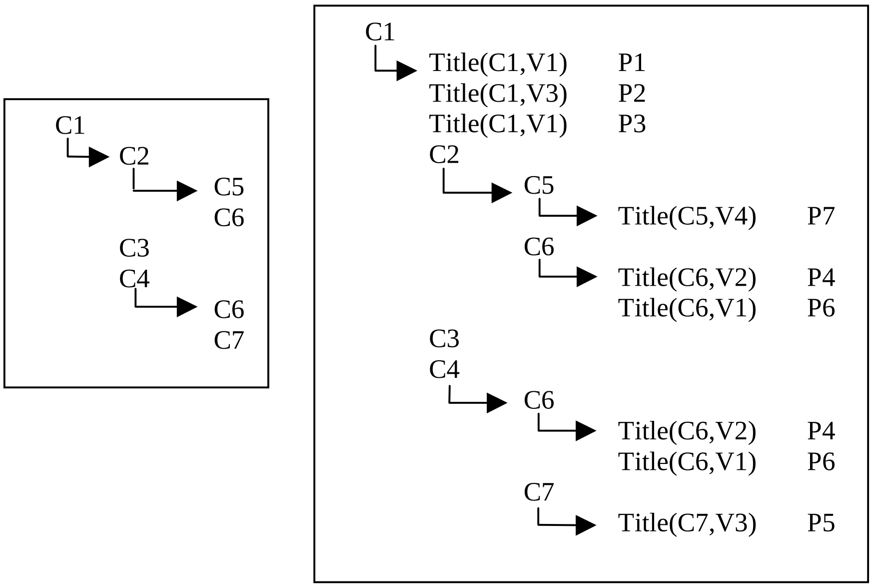


Fig. 2. Navigation structure after including concepts (left), and after including outcomes and associated pages (right). Title(C,V) denotes the title generated for a page with outcome V C.

In this paper, we do not distinguish between different kind of pages. It is, however, envisaged, that there may be a need for at least two kinds: explanation pages that teach the students the outcomes, and exercise pages that assess the student's mastery of the outcomes (a distinction with example and test pages could also be made).

3.2 What to remove?

3.2.1 Removing unnecessary sections

The navigation structure should make it easy and efficient for students to navigate and study the course material. Efficiency can be improved by removing sections that have no content; that do not contain any pages or subsections. For instance, in the example above, clicking on C3 would lead to nothing. So, C3 can be removed. It is also inefficient to have a section with only one page (or one subsection) in it. After all, after clicking on the section, the user would only have a choice of one item. For instance, C5 has only one page associated with it and no subsections, so clicking on C5 would lead to seeing only one link. The user would have to click that link, making it rather inefficient. So, C5 could be removed, and its page P7 could be included as a page of C5's parent C2.

To summarize: A concept will be included in the navigation structure if it has

1. at least two children that are included in the structure, i.e., two subsections, or
2. two pages that use it in their outcomes, or
3. one page that uses it in its outcomes and one child that is included in the structure.

Note that more sophisticated algorithms are possible, which also remove sections if they have more than one child, but their parents do not have many children. The reason for this would be to balance the navigation tree in order to improve efficiency.

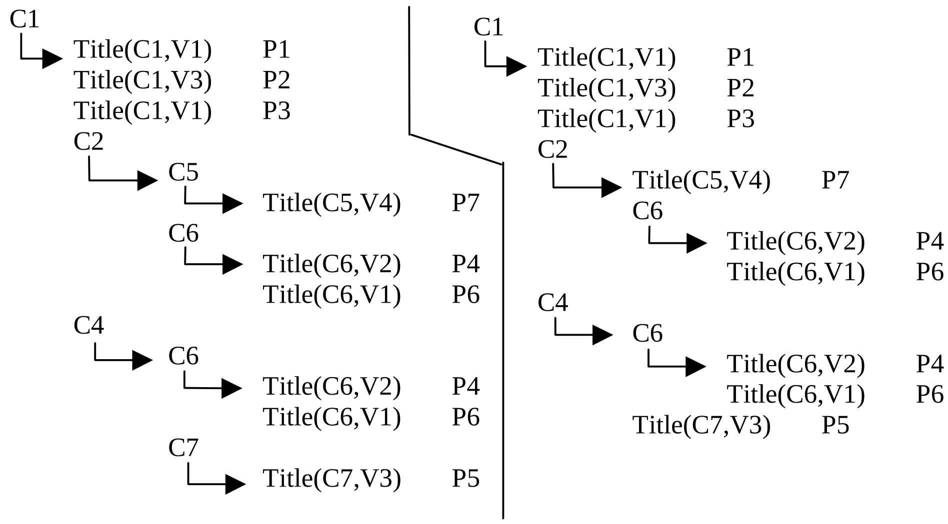


Fig. 3. Navigation structure after removing C3 (left), and after also removing C5 (right).

3.2.2 Dealing with multiple occurrences

The concept network does not have to be a tree: a concept can be a child of multiple other concepts. Additionally, a page can use multiple concepts in its outcomes. In those cases, the generated hierarchy may contain multiple instances of a concept (as section title) or page. Paper-based textbooks include all material only once. So, the question arises whether we should remove multiple occurrences, and, if so, which occurrence to leave. For instance, in the example above, concept C6 is a child of both C2 and C4. As a result, our current navigation structure contains C6 twice, including all its material (P4 and P6). In this example, there are three possibilities:

1. Remove the occurrence of C6 as subsection from C4. This results in Figure 4.
2. Remove the occurrence of C6 as subsection from C2.
3. Leave C6 as subsection of both.

Removing multiple occurrences can substantially simplify the navigation structure. For instance, in the example, removing C6 from C4, not only removes C4 and all its material (Figure 4, left), but also allows the additional simplification of removing C4 (using the rule described in the previous section) (Figure 4, right). However, it poses the problem of deciding which occurrence the leave. This could be the one resulting in the least simplification, or the first one in the hierarchy after ordering (see Section 3.3). An alternative worth considering would be to leave the one the student sees first. For instance, if the student clicks on C2 first, they will see C6. Clicking on C4 afterwards will not show C6. On the other hand, if the student clicks on C4 first, they would see C6, but not see it when clicking on C2 afterwards. Note that this may exclude the possibility of further simplifications.

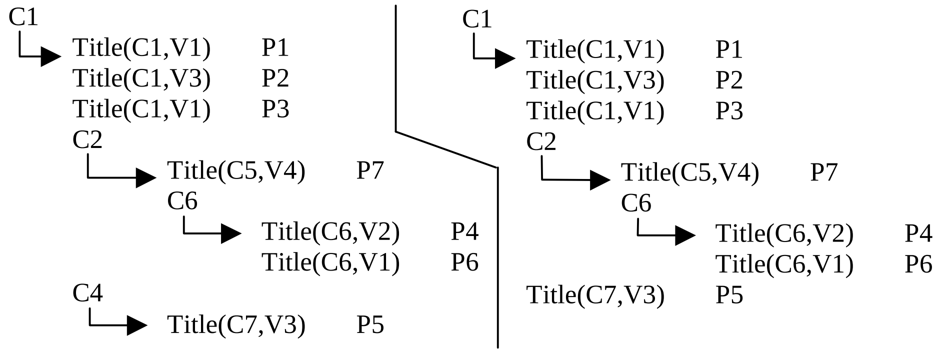


Fig. 4. Navigation structure after removing an occurrence of C6 and its children (left), and after also removing C4 when simplifying according to the rules in Section 3.2.1 (right).

We have, however, chosen the third option: include all occurrences. Removing occurrences would require the student to remember where they had seen certain material, if they wanted to consult it again. Maintaining multiple occurrences will allow the student to find material more easily, and to follow different paths through the course more easily. Whether this is the right decision needs to be investigated in an experiment.

3.2.2 Dealing with alternatives

Different authors may have different views on how a specific outcome should be taught. So, different pages can exist with the same outcomes. We call these pages *alternatives*. As alternatives have the same outcomes, they will share the same generated titles (see [8]). Pages P1 and P3 are alternatives in the example above. As alternatives share the same title, that title needs to be included in the navigation sequence only once.

The decision on which page out of a set of alternatives to include in the textbook can be postponed till the moment that the student (or artificial teacher) clicks on the title for the first time. Several strategies can be used. A page can be chosen on the basis of:

- Author: written by the same author as the previous page in the section (if exists), or by the same author as the last visited page. This may support narrative flow. Students (or teachers) may also provide an ordered list of their favorite authors.
- Its learning style. This would require the author to annotate learning style related characteristics of a page. Alternatively, deductions could be made based upon the page content, like use of images, formulas, words like "example".
- A quality measure. Experts could review (alternative) pages and indicate a quality rating. Students could also rate pages seen. Alternatively, deductions could be made based upon the time spend on the page (not very reliable), links clicks, and the students performance on exercises after having visited the page.
- Its prerequisites being most closely met. Different authors may have different views on the relative order in which outcomes need to be addressed. This can lead to pages sharing the same outcomes, but having a different set of

prerequisites. When links are annotated, like in [7], the annotation should be the most favorable one, i.e., if there is a page with the outcome whose prerequisites have been met, then the "ready to be learned" indication ought to be given.

The student can select an alternative, via links available from the page.

3.2.3 Making a custom book

We want authors to contribute to an organically growing amount of material that may cover every possible topic (as the web does). This makes it highly unlikely that a student would need or want to learn everything. Note that even when restricting the authors to write about a restricted domain, like "Java programming", there still would be large differences between students with respect to their foreknowledge and the outcomes to be reached. That is why courses on different levels tend to use different textbooks.

Different electronic textbooks can be generated to suit individual students or groups of students. Authors, teachers and students can choose the outcomes to be covered by a book. The system will only include pages that are related to achieving those outcomes, either directly or indirectly (needed for prerequisite). The foreknowledge of the student(s) could also be specified, so that pages with outcomes that have already been mastered are not included.

3.3 Ordering siblings

Pages can have outcomes that use different verbs on the same concept. For instance, a page with outcome "implement loops", and another page with outcome "explain loops". We call such pages siblings. In the example above, pages P1 and P2 are siblings, as are P4 and P6. When generating the hierarchy, siblings are ordered in such a way that a page with outcome X will precede all siblings with prerequisite X. For instance, page P4 needs to precede P6, as P4 has outcome V2 C6, which is a prerequisite of P6 (see Figure 1).

Concepts can be children of the same parent concept. We also call such concepts siblings. For example, concepts C2 and C4 are siblings. When generating the hierarchy, as far as possible, siblings are ordered in such a way that a concept with outcome X will precede all siblings with prerequisite X. The outcomes of a concept are the outcomes that contain it and the outcomes of its children. The prerequisites of a concept are the prerequisites that contain it and the prerequisites of its children minus its outcomes. For instance, concept C2 has outcomes V4 C5, V2 C6 and V1 C6, and prerequisites V1C1 and V3 C7. Concept C4 has outcomes V1 C6, V2 C6 and V3 C7, and no prerequisites. As V3 C7 is an outcome of C4 and a prerequisite of C2, C4 should precede its sibling C2.

A page can have an outcome that contains a concept that has a child concept. We call such a page and child concept siblings. For instance, page P1 is a sibling of concept C2. The same kind of rule can be used for ordering them. For instance, as V1 C1 is an outcome of P1 and a prerequisite of C2, page P1 should precede concept C2.

Within these constraints, different strategies can be used to support narrative flow.

- A page or concept with outcome X is followed immediately (or as soon as possible) by a page or concept with prerequisite X.
- A page is followed by another page by the same author.

The relative level of verbs in Bloom's taxonomy [5] can also aid the ordering process. For instance, "explain" precedes "design", which in its turn precedes "evaluate".

Note that the ordering of siblings needs to precede the simplification process of removing not-needed sections, but needs to follow the process of generating a custom book.

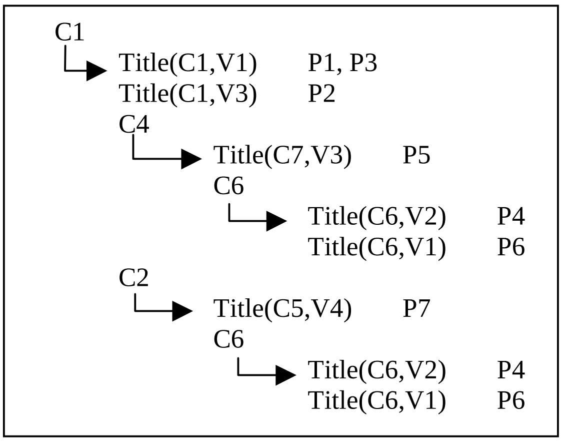


Fig. 5. Navigation structure after ordering siblings.

4 Evaluation

4.1 Criteria for a navigation structure

Before evaluating the automatic generation of a navigation structure, we need to decide what criteria such a navigation structure should meet. The literature on usability, such as [9], mentions a number of guidelines that are particularly relevant for navigation structures:

1. Consistency. The same words should be used to indicate the same objects. This means that pages about the same topics should have similar titles. Our use of generated titles based on outcomes seems to fulfill this criterion. Consistency also means that the navigation structure should not *perceivably* change over time (this does not exclude the use of link annotation). So, if the user has clicked on a link in the past, then clicking on it again should have the same result: show the same submenu, or the same page. User would not expect the sections of a chapter in their textbook to change overnight. We have taken this into account in the way we deal with alternatives (see Sections 3.2.3). If the course content could change

because of authors adding material, consistency could be served by adding a "New section added by author" annotation.

2. Visibility of system status. The user should know at any time where they are, how they got there, and where they can go from there. This advocates the use of a hierarchical, tree-like structure, where the user can immediately see where they came from (parent in the tree) and where they can go next (children in the tree, and perhaps other parts of the tree like parent and siblings). This is why we have chosen to generate that kind of structure.
3. User control and freedom. We believe that the possibility to generate custom books supports user control. However, the automatic generation of the navigation structure might make authors feel less in control. We have to investigate whether it does diminish the authors' feeling in control, and what control over the generation process authors would like to have.
4. Efficiency. Experienced users should be able to find their way around quickly. This means that once a user has seen certain information, they should be able to find it again without any problems. We believe that incorporating all occurrences (see Section 3.2.2) will increase efficiency, but we have to investigate whether that is true.
5. Error prevention. Prevent users from making mistakes rather than just producing good error messages. This means that error messages like "no pages available about this topic" should not occur, by preventing users from clicking on those topics. Our removal of superfluous sections (see Section 3.2.1) is a way of error prevention. Also, our incorporation of multiple occurrences (see Section 3.2.2) can prevent errors (users searching for information in the wrong place).

In addition to these usability related criteria, the main criteria seem to be narrative flow and educational impact (an order which is good for a student learning about the topics).

4.2 Outline of experiments

To evaluate the approach outlined above, we want to conduct a number of experiments. One such experiment could take place at the Adaptive Hypermedia conference. Various experimental designs can be used:

1. A group of subjects, named "Authors" are asked to independently write pages about a certain topic of which they are knowledgeable. At the conference, this could be Adaptive Hypermedia. They are required to provide outcomes and prerequisites for each page, in terms of concepts and verbs provided.
2. Alternatively, an initial network of concepts and an initial list of verbs could be provided, but authors could be allowed to add new concepts and verbs.
3. Alternatively, no concept and verbs would be provided, and authors would have to build up the concept network and verb list themselves.

It can be investigated whether it makes any difference whether

1. Authors can see descriptions of pages added by others, but not the content
2. Authors can see both descriptions and content of pages added by others

3. Neither descriptions nor content of pages added by others. The reason for doing this would be because it might make it more likely that alternatives will occur.

After this has taken place, a second group of subjects can be asked to (independently or as a group) look at the page descriptions (prerequisites, outcomes, author) and concept network and decide on a suitable navigation structure. This can be compared to the navigation structure generated by the Authoring Coach. It can also be investigated how looking at the page content would influence the decision of navigation structure.

6 Conclusions

This paper describes in more detail the automatic generation of navigation structures for adaptive web-based instruction systems that was first introduced in [8]. Though this work is only in its starting phases, it is clear that the choice to generate hierarchies automatically poses interesting problems and opportunities. If we want web-based instruction systems to grow organically, while maintaining narrative flow and user guidance then this seems a way forward.

Clearly, future work should address the evaluation of this approach, and we have sketched how this could be done. It will be interesting to investigate its applicability to different domains. We are also investigating how authors could be supported in producing the concept network (similar problems could occur as those faced in the semantic web). Finally, we are extending our work on the Authoring Coach in the area of giving advice to authors about the completeness, and possibilities for adaptivity of the material produced. Some initial ideas on this were discussed in [8].

References

1. Masthoff, J. An agent-based interactive instruction system. Ph.D. thesis, Eindhoven University of technology. (1997)
2. Brusilovsky, P., Eklund, J., and Schwarz, E. Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems, Proceedings of Seventh International World Wide Web Conference (1998) 291-300
3. Weber, G., Kuhl, H-C., Weibelzahl, S. Developing adaptive internet based courses with the authoring system NetCoach. Proceedings of the third workshop on adaptive hypertext and hypermedia, UM2001, TU/e Computing Science Report 01/11. (2001).
4. Murray, T. MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. Submission to the International Journal of Artificial Intelligence in Education. As accessed on http://helios.hampshire.edu/~tjmCCS/cv.html December 2001.
5. Bloom, B.S. (Ed.). Taxonomy of educational objectives: Book 1 Cognitive domain. Longman, London (1979)
6. Reiter, E. and Dale, R. Building natural language generation systems. Cambridge university press (2000)
7. Brusilovsky, P., Schwarz, E., and Weber, G. ELM-ART: An intelligent tutoring system on World Wide Web. In C. Frasson, G. Gauthier, & A. Lesgold (Eds.): Third International

Conference on Intelligent Tutoring Systems, ITS-96. Lecture Notes in Computer Science, Vol. 1086. Springer Verlag, Berlin (1996) 261-269.
8. Masthoff, J. Towards an Authoring Coach for Adaptive Web-Based Instruction. Proceedings of the Adaptive Hypermedia Conference, Malaga, Spain. (2002)
9. Nielsen, J. Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY. (1994)
10. Hall, W., Davis, H., Hutchings, G. Rethinking hypermedia: The microcosm approach (Electronic Publishing Series, No 4). Kluwer Academic Publishers, 1996.

Re-using Adaptation Logics for Personalised Access to Educational *e*-Content

Charalampos Karagiannidis and Demetrios Sampson

Informatics and Telematics Institute (I.T.I.)
Centre for Research and Technology – Hellas (CE.R.T.H.)
42, Arkadias Street, Athens GR-15234, Greece
Tel.: +30-10-6839916/17
Fax: +30-10-6839917
`karagian@iti.gr, sampson@iti.gr`
`http://www.iti.gr`

Abstract. This paper defines adaptation logics through the set of adaptation constituents, determinants and rules which underlie adaptive behaviour. It then presents the work of the KOD European project for re-using adaptation logics, through their common description which is based on (the extension of) existing learning technologies specifications and standards. The paper also discusses the possible extension of the KOD approach in personalised applications and services, in general.

1 Introduction

A number of R&D efforts are addressing the delivery of flexible and individualised access to learning material, resulting in a series of fruitful systems in the areas of intelligent tutoring systems, adaptive educational hypermedia, intelligent pedagogical agents, etc [1]. However, personalised learning (PL) technology is still not mainstream, mainly due to the difficulty in *re-using* research results [2].

In the context of this paper, PL systems are defined by their *"adaptation logic"*, which is, in turn, defined by:

- *PL constituents*: the aspects of the application that are subject to adaptations; for example, is the user interface, or the learning content being adapted? and which parts of the content are adapted/modified?
- *PL determinants*: the aspects that drive adaptations; for example, do we decide adaptations based on the learners' background? and if yes, how do we define this background?
- *PL rules*: the rules that underline adaptation, i.e. match specific constituents to specific determinants [3].[1]

[1] This scheme is in line with the "classic" ITS model, which includes a student model (PL determinants), a content model (PL constituents), and a tutoring model (PL rules).

Each PL system can be based on a different adaptation logic, i.e. it can address different constituents (i.e. modifying different aspects of the learning content), determinants (i.e. adaptations decisions can be based on different learner characteristics) and rules (i.e. adaptation rules can be based on different learning theories and instructional models, etc).

In most existing systems, however, the adaptation logic is not *explicitly* represented in a *common* format, but it is rather encapsulated in a set of adaptation rules which are usually hardwired into the system. As a result, adaptation logics (and successful design practices) cannot be easily customised and re-used across different applications and services, resulting in the "under-exploitation" of PL technology [2].

This paper presents the work of the KOD European project (see acknowledgements section) in this context. The KOD project makes use of the specific characteristics of the PL domain, for facilitating re-usability of adaptation logics (as this term is defined above). In particular, the project builds on the facts that:

- PL applications and services share some common adaptation constituents (e.g. learning content) and determinants (e.g. learner profile), and
- these aspects can be represented in a common format, through international learning technologies specifications.

The KOD project aims to facilitate the re-use of adaptation logics for personalised access to educational e-content, through (the extension of) learning technologies specifications. The aim of the paper is to outline this work, and to also discuss whether, and how it can be extended, so that it can be applicable beyond personalised learning, i.e. for adaptive and personalised applications and services, in general.

2 Current Status

The need for describing in a common format the "aspects" that characterise learning technologies has resulted in a number of international standardisation activities. The main initiatives in the area are the IEEE LTSC (Learning Technologies Standards Committee, ltsc.ieee. org), the European CEN/ISSS Learning Technologies Workshop (www.cenorm.be/isss/Workshop/lt/), the IMS (Instructional Management Systems) Global Learning Consortium, Inc (www.imsproject.org) and the US ADLnet (Advanced Distributed Learning Network, www.adlnet.org) [4].

These efforts have already resulted in a number of specifications for *e*-learning applications and services. In particular, current specifications enable the description in a common format of learning objects (through educational meta-data specifications, e.g. the IEEE LOM [5]), questions and tests (e.g. through the IMS Question and Test Interoperability – QTI – Specification [6]), learner profiles (e.g. through the IMS Learner Information Profile – LIP – Specification, [7]), etc, so that they can be easily interchanged between different e-learning applications.

Moreover, existing specifications enable users (learning material authors, tutors, publishers, e-learning platform and service providers, etc) to describe in a common format *content packages*, i.e. collections of learning objects, through the IMS Content Packaging (CP) Specification [8]. Each content package can be described in a common format, and can be "packaged" in a single zip file, which includes:

- all the learning objects included in the package;

- a manifest XML file, which describes the structure of the learning objects; the manifest file includes a number of "organizations", each one including, in turn, a number of learning resources (see Fig. 1).

However, the current CP specification enables only the description of table-of-contents like structures: "It is possible to imagine organizations that will take into account such approaches as hierarchical 'branching', indexes, custom learning paths utilizing 'conditional branching', and complex objective hierarchies. While many content organization approaches may be developed, a default approach is included as part of this specification. This default approach to content organization is referred to as a 'Table of Contents' scheme and is encompassed in a <tableofcontents> element" [8]. That is, the current version of the specification does not facilitate the description of navigation rules which specify which learning objects comprising a content package should be selected for different learners.

In summary, through existing learning technologies specifications, we can describe in a common format, and thus interchange and re-use:

- PL determinants, i.e. learner characteristics, e.g. through the IMS LIP, and
- PL constituents, i.e. learning object characteristics, e.g. through the IEEE LOM.

However, we *cannot* describe in a common format PL rules, since the IMS CP specification does not facilitate the description of adaptation rules.

3 KOD Innovation

In this context, the KOD project is developing an extension of the IMS CP specification, so that it can support the common description and re-use of PL rules (Fig. 1).

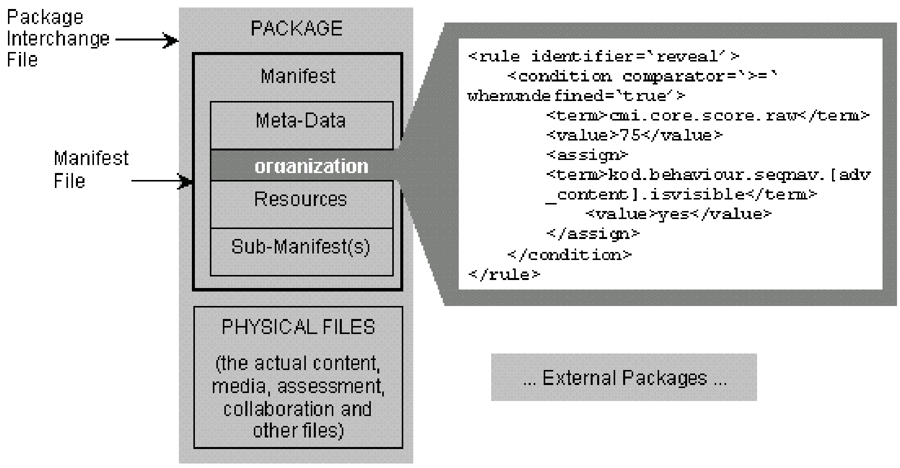


Fig. 1. The KOD Knowledge Packaging Format; an example extension over the IMS CP specification is shown, which represents an adaptation rule determining that "advanced content" is revealed only after the learner passes a specific test.

The project is experimenting with the *knowledge packaging* format for the common description of *adaptation rules*, determining which learning objects of a knowledge package (i.e. as opposed to an IMS CP-compliant content package) should be selected for different learner profiles. That is, in addition to the learning objects and their structure, we can define in a common format adaptation rules, which specify which organizations and/or learning objects should be selected for different learner profiles [9].

3.1 Developing Adaptive Educational e-Content

The KP format is used as the interchange format within the KOD-based prototype vertical learning portal, which aims to support learning material authors to design, develop, modify, publish, search, retrieve, broker, interchange and re-use adaptive educational e-content; as well as individual learners to have access to learning material in a personalised way.

The main tool available within the KOD prototype for the development of knowledge packages (i.e. adaptive educational e-content described through the KP format) is the KOD Packager [10]. Fig. 2 shows a screenshot of the KOD Packager, where the following features are shown:

1. at the left-hand side window, the KOD system user (learning material author, publisher, tutor, etc) can import existing learning objects, which can then be used for the development of knowledge packages; the learning objects can be imported both from the local file system, the KOD prototype repository or from publicly available resources on the WWW;
2. at the middle window, the on-line content author can create a number of organizations (similarly to the IMS CP specification, i.e. containers of learning objects for domain concepts) for each "knowledge unit" (or concept) related to the domain of the knowledge package; in the example screenshot, the domain of the knowledge package is tele-radiology, and the following concepts have been defined: introduction, definitions, implementation, basic parts, image acquisition and management, display systems, communication networks, interpretation, image standards and compression;
3. at the right-hand side window, the KOD system user is able to edit the learning objects that have been imported; in this example, the user can modify a text file, including a definition of tele-radiology;
4. the previous steps are in line with the IMS CP specification; in addition, the KOD Packager facilitates the definition of navigational rules, for determining which learning objects should be selected for different learner characteristics;
5. this can be done through the Rule Editor; as it is shown in the screenshot, the user can define a number of rules for the same knowledge package; also, one rule can include other rules, thus allowing nesting of rules; for each rule, the KOD system user can define:
   i) conditions: they are based on learner (profile) characteristics; examples include the learner's learning styles, interests, goals, achievements, etc;
   ii) assignments: these are the post-conditions of rules; one rule may determine, for example, that the learner's expertise in a concept is

increased, after accessing learning material on this concept, or passing a relevant test;

iii) launch request: a number of actions can be initiated; for example, after a learner accesses learning material or passes a test on a specific concept, additional learning material can be proposed (e.g. based on the results of the test), new concepts can be presented to the learner, and so on.

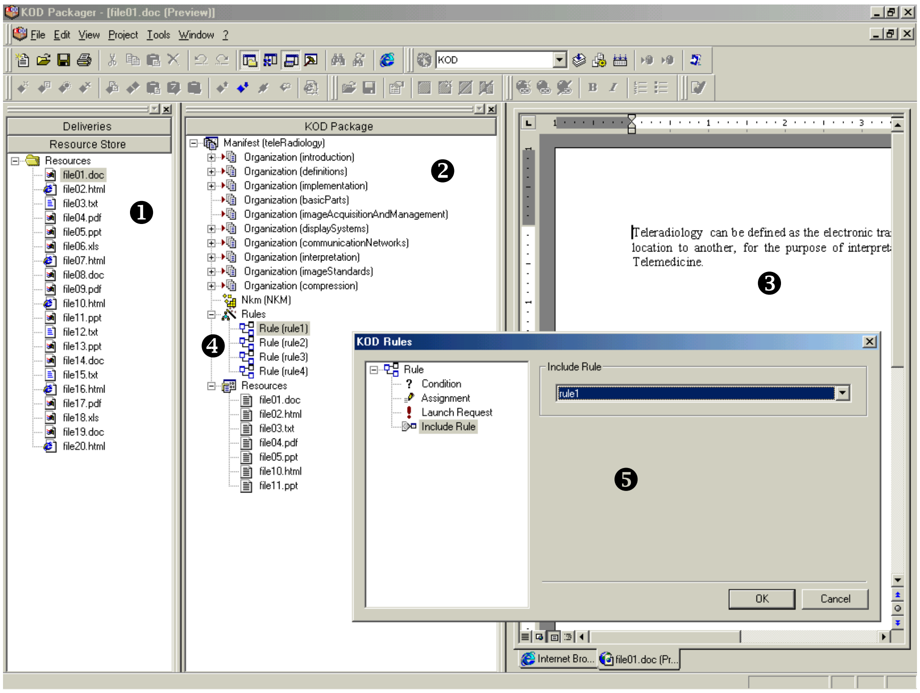


Fig. 2. The KOD Packager for defining Knowledge Packages within the KOD VLP.

3.2 Interchanging and Re-using Adaptive Educational e-Content

The KOD prototype aims to assist the interchange and re-use of adaptive educational e-content. That is, the knowledge packages that are created through the KOD Packager can be published in the KOD prototype system, so that they can be searched, retrieved and accessed by learners in a personalised way (see next section); or re-used by authors for the creation of new knowledge packages.

In addition, the KOD system facilitates the re-use of the "building blocks" of the knowledge packages. That is, within the KOD prototype system, authors can publish, search, retrieve, broker, interchange and re-use:

- learning objects: authors can publish their learning material in the KOD system, together with their meta-data description, so that it can be re-used; as a result, an author may, for example, search in the KOD prototype VLP, to find learning material which is related to the "definitions of tele-radiology" domain;

- organizations: as stated before, organizations in the knowledge packaging format are used as containers, for including the learning objects which are related to a specific concept of the knowledge package; these organizations can be published in the KOD prototype VLP, so that they can be re-used for the creation of new knowledge packages; for example, an author developing a knowledge package in the tele-radiology domain can search in the KOD prototype VLP to retrieve and re-use an organization for the "definition of tele-radiology" concept;
- adaptation rules: authors can publish, search and retrieve adaptation rules which have been published in the KOD prototype system; for example, one author can search and retrieve rules which are related to the tele-radiology domain, and select specific learning objects according to learner profile characteristics.

3.3 Accessing Adaptive Educational e-Content

The aim of the KOD system is to facilitate personalised access to learning material. That is, each learner can receive learning material which is adapted to his/her individual profile.

To this end, the KOD prototype maintains a profile for each learner, which is based on the IMS LIP (Learner Information Profile) Specification [7]. That is, upon entering the KOD VLP for the first time, each learner is prompted a short questionnaire, for determining his/her characteristics. This profile is automatically updated, taking into account the learner's interactions with the KOD VLP, e.g. the knowledge packages that have been viewed, the assessment questions that have been answered, etc.

Subsequently, the learner can search the KOD VLP, and retrieve knowledge packages in specific domains of interest. The KOD system adapts the results of this search according to the learner's profile.

The learner can then access the knowledge packages in a personalised way. That is, the KOD e-learning system can import knowledge packages, disaggregate them, interpret the rules included in them, and select to present the learning objects which are appropriate for the learner, according to his/her profile.

Fig. 3 shows an example adaptation of a knowledge package in the tele-radiology domain. In the left-hand side, the KOD system suggests the "acquisition systems" section to be viewed next (indicated with the red arrow), since this is the most appropriate learning object according to the learner's profile. It also suggests that specific sections and not yet ready to be viewed; while, in the right-hand side picture, the suggestions have changed, after the learner has viewed the "acquisition systems" section, and answered correctly to a specific assessment question.

Fig. 3. Personalised access to Educational e-Content within the KOD VLP

4 Discussion

As it is described in the previous sections, when an LMS imports a KOD knowledge package, it can disaggregate it, interpret the rules included in it (since they are described in a common format), match them with the individual learner profile, and present only the learning objects which are appropriate (according to the adaptation rules) for the individual learner's skills, interests, preferences, etc. That is, personalised access to educational e-content can be supported (Fig. 4).

The innovation of the KOD project is that adaptation logics can be *explicitly represented in a standard way, within a specific, common element of a universally acceptable specification* (namely, the extended CP specification). As a result, adaptation logics can be re-used across different *e*-learning applications and services.

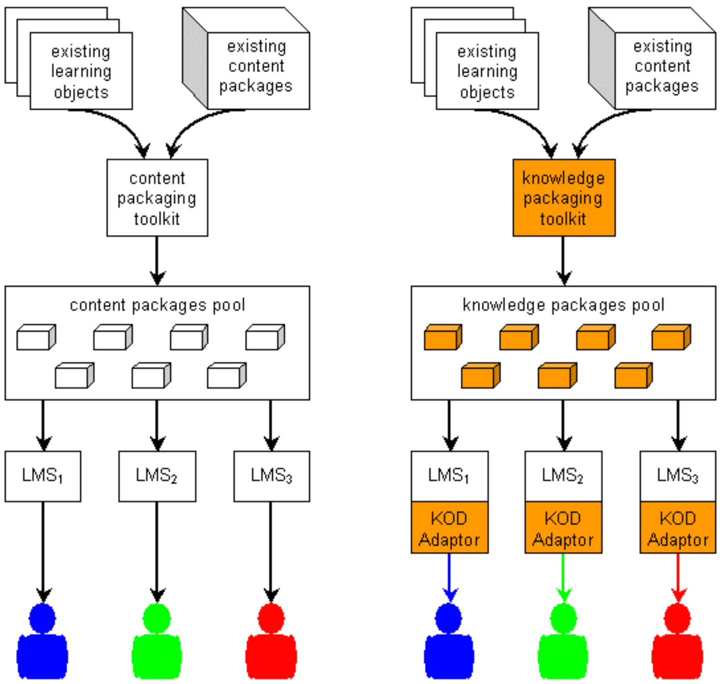


Fig. 4. The KOD approach for personalised access to educational e-content, against existing standards-based approaches. Through the use of knowledge packages (i.e. as opposed to content packages), we can select different learning objects for different learners, according to their profile.

This innovation mainly builds on the fact that we address a specific domain of adaptive and personalised applications and services: personalised learning, education and training. This domain has the distinguishing characteristics that:

- specific aspects of adaptation logic, i.e. adaptation constituents (learning content) and determinants (learner characteristics), are common to (almost all) applications, and
- there are already common learning technologies international specifications for describing these attributes; as a result, we can describe in a common format, and thus interchange and re-use PL determinants and PL constituents;
- in addition, the KOD project is proposing a way of representing in a common format PL rules, through the knowledge packaging format (Fig. 5).

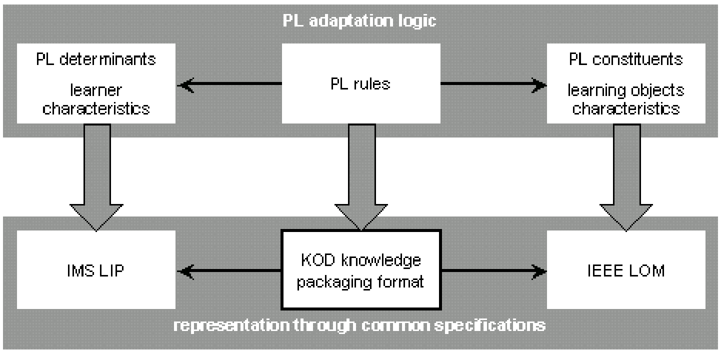


Fig. 5. Representation of PL adaptation logics in KOD, through learning technologies specifications.

The fact that we build on existing learning technologies specifications also poses some limitations to our approach. For example, re-usability and interoperability can only be realised between applications that are compliant with the proposed specification. Moreover, re-usability can only be *ensured* for adaptation logics which are defined over PL determinants and constituents which can be represented through commonly accepted specifications and standards. For example, we can define re-usable adaptation logics which include learner characteristics as an adaptation determinant, since the latter can be represented through the IMS LIP Specification. Also, we can describe adaptation logics which modify the learning objects selected for each learner, since learning objects can be represented in a common format through the IEEE LOM specification. On the other hand, it cannot be ensured that we can define adaptation logics which include adaptation determinants or constituents which cannot be represented through learning technologies specifications (e.g. learner characteristics that cannot be described through the LIP specification).

As described above, the KOD project has already developed the knowledge packaging format, as well as a number of tools for the definition and re-use of adaptive educational e-content. Currently, we are developing three demonstration knowledge packages, in the fields of e-business, tele-medicine and knowledge management.

Our future work in this field involves the investigation of the extension of the "KOD approach", so that it can be applicable to other domains as well (i.e. beyond personalised learning). From the above presentation, it follows that such an extension would require that:

- specific adaptation constituents and determinants are common to (almost) all applications in a specific domain; and
- those common constituents and determinants can be represented through commonly acceptable formats.

One possible extension would build on the Dublin Core meta-data set (dublincore.org), for representing adaptation constituents. The Dublin Core meta-data set is used for the description of *content* objects, in general, as opposed to the IEEE LOM meta-data set which is used for the common description of learning objects, i.e. material which is used, re-used or referenced in technology-supported learning [5].

Therefore, it may facilitate the description in a common format of adaptation constituents which can be used in personalised applications and services, in general.

Acknowledgements

Part of the R&D work reported in this paper is carried out in the context of the KOD "Knowledge on Demand" Project (www.kodweb.org, kod.iti.gr), which is partially funded by the European Commission, through the Information Society Technologies (IST) Programme (Contract No IST-1999-12503). The KOD Consortium comprises: CERTH-ITI, Greece (project co-ordinator); FD Learning, UK; GIUNTI Interactive Labs, Italy; CATAI, Spain; and PROFit Gestion Informatica, Spain.

References

1. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. In Brusilovsky, P., Kobsa, A., Vassileva J. (eds.). Adaptive Hypertext and Hypermedia (1998) Kluwer Academic Publishers.
2. Sampson, D, Karagiannidis, C., Kinshuk: Personalised Learning: Educational, Technological and Standardisation Perspectives. Interactive Educational Multimedia, Special Issue on Adaptive Educational Multimedia 4 (2002).
3. Stephanidis, C., Karagiannidis, C., Koumpis, A.: Decision Making in Intelligent User Interfaces. Proc. 3rd International Conference on Intelligent User Interfaces (1997) ACM Press.
4. Sampson, D., Karadimitriou, P., Papaioannou, V.: From Educational Meta-Data Authoring to Educational Meta-Data Management. Educational Technology & Society, Special Issue on Innovations on Learning Technologies (2003) In Press.
5. IEEE Learning Technologies Standards Committee: Learning Objects Meta-Data Specification. Electronically available at ltsc.ieee.org/wg12
6. IMS Global Learning Consortium, Inc.: Question and Test Interoperability Specification. Version 1.2 (2002). Electronically available at www.imsproject. org/specifications.html.
7. IMS Global Learning Consortium, Inc.: Learner Information Package Specification. Version 1.0 (2001). Electronically available at www.imsproject. org/specifications.html.
8. IMS Global Learning Consortium, Inc.: Content Packaging Specification. Version 1.1.2 (2001). Electronically available at www.imsproject.org/ specifications.html.
9. Sampson, D., Karagiannidis, C., Cardinali, C.: An Architecture for Web-based e-Learning Promoting Re-usable Adaptive Educational e-Content. Educational Technology & Society, Special Issue on Innovations on Learning Technologies 5 (2003) In Press.
10. Cardinali, F., DaBormida, G., Vaccaro, R., Grosso, F., Sampson, D.: The KOD (Knowledge-on-Demand) Packaging Toolkit: Sharable Adaptive Content Objects Packaging, Reusability and Brokerage for Learning Society. Proc. European Multimedia, Embedded Systems and Electronic Commerce Conference (2001) IOS Press.

Using StoryML for Content Representation

Petri Gerdt[1], Jarkko Suhonen[1], and Erkki Sutinen[1]

Dept. of Computer Science, University of Joensuu
{pgerdt,jsuhon,sutinen}@cs.joensuu.fi

Abstract. The Woven Stories is a tool for writing stories collaboratively. In order to represent the written stories adaptively, we need to construct description about the content of the stories and the relative information between different stories. The formal presentation of Woven Stories is called StoryML and it can be used for adaptive content representation of the stories.

1 Introduction

Adaptive IR tools in web-based education are essential to represent the content differently according to the specific needs of the students. For example the learning methods and styles between students with holistic-style of learning and students with serialistic-style of learning differs tremendously. It has been shown for example that the holistic-style of learners have flexible and easier way of solving problems than serialistic-style of learners. This means that there is a clear need for representing the learning materials according to nature of the students learning [4, ?]

The *Woven Stories* is a tool where several authors' efforts is managed in a shared writing space, where the authors may write story sections and link them together. A woven story is thus a hyperdocument or alternatively hyperspace, which consists of an arbitrary set of story sections and links between them. The Woven Stories is intended to be used to support group activities in educational settings, such as different distance education situations. The users of the tool can construct, manage and share information with it. Thus the environment works as a backbone of group work for a group of students. A prototype system for the environment labeled Woven Stories has been described in [2]. In this paper we will present our ideas that we are developing for the next implementation of the environment.

The Woven Stories can be defined as hypermedia in following way. The Woven Stories visualizes the content of a hyperspace as a graph, whose nodes are content pages, passages of text which may include hyperlinks. The arcs of the graph represent links between the content pages. Each content page may be written by a single user or several users. The Woven Stories uses an XML-based data structure to store information about the hyperspace. [3]

One of the basic requirements for the Woven Stories is the need for adapting the representation of the content for example between holistic-style and serialistic-style of learner. Hence, the important feature of the Story ML is to capture the episodic nature of stories. However, in our opinion such adaptation techniques are not available [1].

A node in a concept map is often related only to its neighboring concepts. One might say that a concept map represents only the local information of a particular concept. On the contrary stories typically represent episodes that consist of a several sub-sequent story sections connected by links. The analysis of episodes could be useful to make assumptions about the students' reasoning during the learning process. Moreover, information on different episode patterns could be stored by the StoryML specification, and actual story lines corresponding to a given pattern could be retrieved on demand. An example of episode patterns is a sequence A1-B1-A1 which represents a three step dialog, between students A and B.

Furthermore, the users of the Woven Stories can customize their views of the collaboratively constructed hyperspace. A member of the group may need a specialized view of the hyperspace to support a specific activity. If the students co-author an essay one of the students may be an editor who needs to all material produced by the group. On the other hand an author constructing the introduction of the essay will not want to see everything when she is concentrating on her own contribution.

To support the user specific customization of the view to the hyperspace the environment collects user model for each user. The user model includes data about the users' view preferences, personal settings (such as the user name and the color representing the user) and the browsing history of the user. The environment extracts much of the data for the user model from the XML-based description of the hyperspace and its contents, but it collects some of the data by examining user actions (see section 3).

There are no limits for the size of the hyperspace in the environment. Thus the users may write a single content page together, or write several, perhaps interlinked, pages. The hyperspace can become very complex if it contains many pages written by several authors. In this case a user may not find what she wants or she may not notice some input that she should. To help the users to cope with in the hyperspace the environment provides searching and information filtering tools, which are outlined in section 4.

A challenge for the environment is to adaptively support very different views as the users customize their personal views of the hyperspace and use the information filtering and search tools that the environment provides (see section 5).

2 Adapting Content Representation in Woven Stories

In educational setting the adaptive content representation in Woven Stories means that the same topic or story can be presented in multiple ways to different users. Hence the representation of the stories can be varied according to the learning style of a particular learner.

For example there are evidence that concept mapping fits best to the natural study style of the holistic [4]. It proved to be difficult to make serialistic students benefit from the concept mapping approach. Here we expect to find woven stories as a beneficial compensation for the problems holistic students have in chronological reports. In summary: woven stories and concept mapping have the natural tendency to be mutually supportive. In global terms we may expect that the serialistic students will benefit more from concept mapping, while the holistic students get a typical compensation from the woven stories approach.

In order to make the transformation between different types of representation we need a StoryML description. Figure 1 presents the general idea of transforming a particular topic to various modes of presentation. In Figure 1 CM indicates concept map and WS woven stories. Topic *t* can be for example a physical phenomenon or course contents. The change from Woven Stories representation to concept map form is straight operation. The StoryML includes all the necessary information. The transformation from concept maps to Woven Stories is a little bit elaborated, because some information needs to be collected from the authors before the transformation can be done. For example the order of the concepts does not necessarily appear in the concept-like representation.

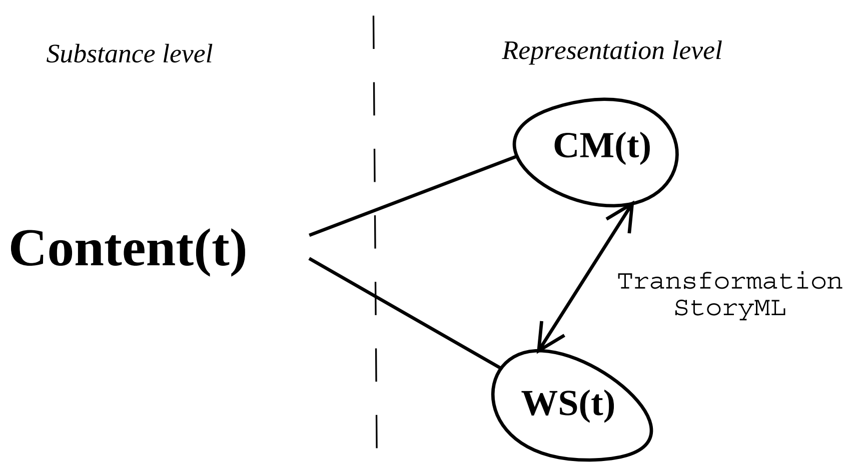


Fig. 1. An Example of Different Representations for a Given Topic t

3 Data Management And The User Model

We have defined a XML-based data structure, called StoryML, to describe a hyperspace and its contents in the Woven Stories environment [3]. The StoryML is divided into five main categories which are summarized in table 1. Each manifestation of StoryML represents a given hyperspace.

The elements presented in Table 1 can be stored in three different layers: locally, episodically and globally. The *local* metadata description deals with the single story node and its relationship to the other nodes. The *episodic* level includes two or more nodes that form a distinctive episodic structure. The episodes can be defined by the authors of the stories or by the Woven Stories environment. The *global* level includes all the single story nodes and episodes in one hyperspace of stories.

Naturally the content of a particular StoryML element can differ when the story node is treated locally, episodically or globally. Furthermore the content of the element may change when the structure of the stories is altered. For example the union of two stories probably creates a new set of metadata elements. The metadata description of different levels can be done in couple of ways. In first approach the metadata description of every story node holds the data about the local, episodic or global information. Second approach creates metadata descriptions to different layers. Hence, every episodic structure has its own metadata description.

A user model for a specific user can partly be extracted from the Author and Content elements. The Author element stores information that the user has inputted about herself: name, affiliation, color, email and preferences. The content element on the other hand has knowledge about what the users have been doing in the environment. This is a case of collaborative user modeling: the user inputs data about herself and her preferences and the environment builds a part of the user model by examining what the user has done in the environment.

To complete the user model the environment records data about the users behavior: the environment collects the browsing history of the user, and data about any searches and filtering operations that she has done.

4 Adaptive Operations Using StoryML

We plan to include various tools in our environment in order to let the users customize their views and find relevant data.

Table 1. The main categories of the StoryML core element set.

Element	Description
General	General information about the Woven Stories.
Visualization	Determines the visual appearance of a graph-like Woven Story.
Content	Descriptive information about the content (e.g. keywords).
Author	Data about the authors who have been active in the hyperspace.
Episodes	Links and time factors between story nodes.

4.1 Information Filters

Information filters provide a subset of the hyperspace. They are not intended to be used to find a specific piece of content, but to restrict the hyperspace by some constraints to make it easier to handle.

Subsets of the hyperspace defined by a subset of the authors. The most intuitive use of filtering the hyperspace by a subset of the authors would be a user checking her own contribution. By broadening the filter to include other authors the user can for example examine how the contributions of others' are related to her own ones. A teacher can examine what each student member of a group collaborating in the environment has done.

Subsets defined by creation date. The environment may present a subset of the hyperspace, which is limited by the creation or modification dates of the content. This filter may be implemented as a dynamic view, where the user can alter the subset with a slider component, which controls the date parameters. We call this filter type the time line filter, as the user can follow a time line along which the hyperspace is constructed visualizing the creation process. This filter is useful when a user wants to see what has changed to the hyperspace since she last time logged into the environment.

4.2 Search Mechanisms

Searches to the hyperspace are made to find specific content. The search can be made to a filtered subset of the hyperspace. An user can for example search for a keyword among content produced by a specific user.

Keyword search. The authors may attach keywords to the content pages. These keywords can then be used by the search engine to find certain topics.

Searching the written part of the content pages. Unfortunately keywords can be interpreted differently by different people, or they may be missing altogether. In this case free form searches are needed. The environment lets the users to search the content pages with a arbitrary string, looking for matches among all written content.

Searching among the relationships of the content pages. The relationships, or links, between the content pages can be searched. The user may for example look for all content pages linked to a specific page.

Other search mechanisms. Different context analysis heuristics can be used to analyze the content of the text. We can extract such information as subject of the matter, the intention of the text or even the emotional "tone" of the text by using Natural Language Processing (NLP) and Machine Learning (ML) [6]. With this functionality the environment can search for content pages and links based on the deep understanding of the actual content of the hyperdocument. These analysis heuristics may be used to create filters too.

5 Adapting The Results

The user model maintained by the environment makes it possible to present the results provided by searches and filters adapted to the situation of the user performing the operations.

One example of adaptation in the environment is the highlighting of such content pages in a result of a filtering or search that the user has not visited. This adaptation of results use the browsing history of the user who makes the filtering or search.

Another approach for the environment to adapt the results is to use the browsing history of other users. The environment can suggest such results of searches to the user that other users have found useful. Furthermore the environment can use the user models of others together with a specific user model to show the user possible searches or filtering options that other users in her situation have used.

A more complex way to adapt the view of a user would be the environment presenting content pages written by others to an author in order to motivate or help him. For example if a user A often collaborates with user B, then the environment can notify A about B:s behavior and writing, or the it can weight search result that include contributions from B more than other users contributions.

6 Conclusion

In this short paper we have presented our ideas for a collaborative hypermedia authoring environment that uses a XML-based representation for a hyperspace, provides different search and filtering tools and adapts the results of these tools to support its users. We are currently working on the implementation of the environment.

References

1. Brusilovsky, P. Adaptive hypermedia. User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) **11** (1/2), 2001, 87 – 110.
2. Gerdt, P., Kommers, P., Looi, C., Sutinen, E.: Woven Stories as a Cognitive Tool. **Cognitive Technology 2001**, LNAI 2117, Springer-Verlag, 2001, 233 – 247.
3. Gerdt, P., Kommers, P., Suhonen, J., Sutinen, E.:**StoryML: A XML Extension for Woven Stories**. Accepted to be presented in ITS 2002.
4. Huai, H.; Cognitive Style and Memory Capacity: **Effects of Concept Mapping as a Learning Method**. Doctoral Thesis, University of Twente, 2000.
5. Huai, H., Kommers, P.: **Concept Mapping as a Learning Strategy for Autonomous Students with a Serialistic Cognitive Style**. In International Journal of Continuing Engineering Education and Life-Long Learning, 11 (1/2), 135 – 151, 2001.
6. Paik, W. Automatic Generation of Educational Metadata Elements to Enable Digital Libraries. **Proceedings of the 9th International Conference on Computers in Education** (ICCE/SchoolNet2001), Seoul, Korea, November 12-15, 2001, 1406 – 1413.

SHAAD: Adaptable, Adaptive and Dynamic Hypermedia System for content delivery

David Mérida, Ramón Fabregat, Jose-Luis Marzo[1]
Institut d'Informàtica i Aplicacions (IIiA)
Universitat de Girona (UdG)
{david.merida, ramon.fabregat, joseluis.marzo}@udg.es

Abstract. To implement an efficient adaptation of content in the network, different variables can be analysed that involve the types of users, the devices used by customers to gain access, the types of access, the state of the network and the current load on the server. The increasing use of hypermedia objects for creating content and the fact that the variables above mentioned are not taken into consideration has led to the problem of inadequate content in many situations. In this paper, a model is suggested for the delivery of hypermedia content that considers these variables. This model, elaborated from an analysis of different research, considers the different points of view observed and has the end objective of defining a System which is Hypermedia Adaptable, Adaptive and Dynamic (SHAAD). This will be the starting point for considering the problem of content adaptation, by means of a totally modular system.

1 Introduction

At present, the enormous heterogeneity in terms of types and capacities of access devices, bandwidth of the network and needs/preferences of the users are not taken into account by a server when providing web content which is rich in images, audio and video. E.g., the server will deliver the document requested even if the terminal used (WebTV, Personal Digital Assistants (PDAs) or mobile telephones) can not access these content due to the limitations of the display, of the storage capacities, of processing or of access to the network.

To solve this problem, alternatives must be developed that allow universal access to any type of material, from any type of device and that take into account user preferences as well as the current load on the network and on the server.

The concept of *Adaptation* has been widely investigated in the field of hypermedia systems and it has been shown that in these areas it can provide better environments of use and performance. There are many groups dedicated to the task of solving the issue of adaptation of content and there are also different considerations with respect to the implementation of such adaptation.

Some of them are: *UMA (Universal Multimedia Access)* [1] takes the new classes of intelligent and portable devices into account; in *MONADS* [2-4] the adaptability of the data services to the changes of environment of the nomadic users is the main focus of this project; *Research Groups from Hewlet-Packard* [5-6] and *Microsoft Research*

[1] This work has been supported by the CICYT TEL-99-0976 and Galecia (UE-SOCRATES-MINERVA)

China [3] are working on Adaptive Delivery Systems and take into account the type of access device, the state of the network and user preferences; Paul de Bra [7] takes into account user preferences and suggests changing the content or the presentation of the nodes by altering the structure of the link.

In the studies mentioned different variables have been considered that influence content adaptation. They will all be adopted in our work:

- *The characteristics and preferences of the user* [7-9]. With this concept, we include everything referring to the preferences and information about the user. Preferences being everything related to the means of receiving the hypermedia material (e.g. explicit or summarized), the learning characteristics (e.g. if it is a textual or visual user), the personal characteristics (e.g. if it is an extrovert or introvert user), etc. Knowledge is everything to do with previous or evolving knowledge.
- *The customer's access device* [1-6]. Along with the increasing expansion of the Internet there has been wide-ranging technological development of the devices used to access the network. These devices vary enormously in terms of storage capacity, processing power, screen resolution, etc. This leads to a new problem: having to deliver the rich multimedia content available to a wide range of devices/customers which, in many cases, will not have the capacity to deal with them adequately.
- *Type of access to the network* [5-6]. Bandwidths that range from the 28.8K to 100M, mean another feature has to be considered: the speed of access to the network.
- *State of the network* [7], [10-11]. In addition to the diversity of types of connection, we have to consider the state of congestion of the connection being used at any given moment. The load on the connection is not uniform at all times. That is, a connection with a large bandwith does not necessarily mean a permanently optimised connection.
- *State of load on the server* [12-13]. Users know only too well the experience of unsuccessful or rejected access on the part of the server when it involves consulting certain multimedia material. This situation is often caused by an overload in these servers as a consequence of an unusual quantity of requests. To solve this overloading, it may be preferable to deliver content with reduced quality before having to reject or to generate a fault in the connections that are being made.

In Section 2 we take a look at adaptability versus adaptivity in order to clarify the terminology used. In Section 3 we define our SHAAD system and variants of the model are described for the different variables of adaptation. Finally a single model is presented for our system in Section 4. We conclude this paper in Section 5

2 Adaptability Versus Adaptivity

The term *adaptivity* has been used by different authors in different areas. Depending on the environment considered, the goals pursued by such an adaptation vary. For this reason, in the first place we need to agree on what we mean by hypermedia *adaptation* of content in an environment of changing variables, such as the ones mentioned in the previous section.

Abdelzaher [13] considers the adaptation of web content to be a mechanism for improving performance in the face of server overload. Wei-Ying Ma [5] looks at the delivery of content adapted in heterogeneous environments in order to improve content accessibility. Brusilovsky [9] defines an Adaptive Hypermedia System as one that builds a model of goals, preferences and knowledge, for every user. It uses this model, by means of interaction, to adapt to the needs of the user. Oppermann [14] takes the characteristics of the user into account and distinguishes between: *Adaptable Systems* and *Adaptive Systems*. On the other hand, De Bra [7], taking into account user preferences as a variable that decides the adaptation, classifies the hypermedia environments or web sites built according to their capacity to carry out some type of personalization in: *Adaptable Hypermedia*, *Adaptive Hypermedia* and *Dynamic Hypermedia.*

3 Definition of SHAAD

From what has been said in the previous section, we define our Hypermedia Adaptable, Adaptive and Dynamic System (SHAAD) as a *system that, taking into account the state of the variables we mentioned and the variety of multimedia in web content, tries to adapt the available information dynamically or statically and to deliver it in the most efficient way possible.*

The SHAAD model is made up of 4 modules:

1- Mechanisms for defining variables. The aim of these mechanisms is to define the variables we mentioned above: characteristics and preferences of the user, access device of the customer, type of access to the network, state of the network and state of load on the server.

2- Module of content. The function of this module is to deliver the content requested, either through a dynamic generation from the unit elements that make up the web page (on-line generation), or by selection from the various different static versions of this content which have been previously generated (off-line generation).

3- Decision Engine. This is the kernel of the system and the place in which the variables of decision and the available content are evaluated, from which are inferred the mechanisms for delivering the material in the form that best suits the end user.

4- Adaptation Mechanisms. Once the new web site generated by the module of content is available, this module implements the adaptation mechanisms decided upon by the Decision Engine.

We analyse the variables mentioned in the module *Mechanisms for defining variables* separately and define the model used in every case.

3.1 Characteristics and Preferences of the User

In a very general way, the concepts that allow us to describe our user are: goals versus tasks, previous knowledge or background, experience, preferences in the mode of receiving the information, interests, details of the user, etc.

These concepts take different types of characteristics into account [9]: the ones relating to knowledge (previous or evolutionary), those that consider the preferences on the mode of receiving the information (graphical or text, developed or summarized), those that take the personality of the user into account (introvert or extrovert, verbal or textual, ...), etc.

The particular modules of this model (Figure 1) are:

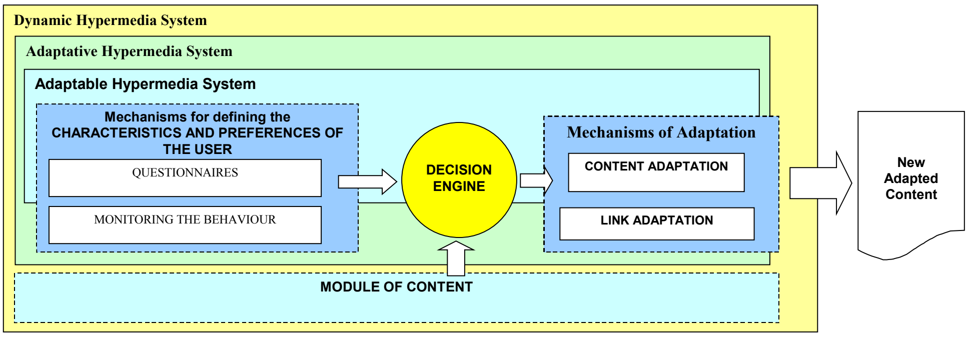


Figure 1 – Model for Content Adaptation according to characteristics and preferences of the user

- *Mechanisms for defining the characteristics and preferences of the user*:
  - *Questionnaires* for the user in order to get direct answers concerning his or her characteristics and preferences.
  - *Monitoring* the behaviour of the user in order to deduce his or her characteristics from the interaction that takes place.
- *Mechanisms of Adaptation* [7]:
  - *Adaptation of links*. The system tries to guide the user towards the most important aspects of the available information, moving him or her away from the less important.
  - *Adaptation of content*. The Adaptive Hypermedia System provides additional or alternative information in order to ensure that the user gets a complete understanding of the material offered.

The terminology used by other authors may differ what we are using, but the mechanisms involved do not differ substantially with regard to the ones we present here.

3.2 Access to the network: The Customer's Device and Type of Access / State of the network

Although these three concepts are different, we shall consider them jointly because they are intimately related in terms of technology. In the end, *the quality of the format of the content* delivered to the user will depend on these variables.

The range of existing devices capable of accessing the network is very wide. Chief among the differences are the information processing power, the availability of audio reproducers, the network access interface and the means of viewing video. This last feature plays an important role when it comes to the adaptation of the content.

Furthermore, these devices depend greatly on the type of connection available and on the speed of access. To these considerations we must also add the *state of the network* at a given moment.

As users, we know that, although we may have an *excellent access device and an excellent connection we do not necessarily have as a result, excellent content availability*. Furthermore, under certain conditions we have often experienced the frustration of not being able to enter the requested web site. This problem can arise because of the state of the network.

For this model it is necessary to describe particularly two modules (figure 2):

- *Mechanisms for defining the characteristics of the customer's device, the type of access and the state of the network*. This module carries out a function similar to the module of Mechanisms for the definition of the characteristics of the user, shown in Figure 1. The methods available and those that we are considering for determining the type of device, connection and state of the network are [5]:

  - *Http protocol*. In the protocol used for the delivery of documents, the head of the http request contains relevant and useful information about the customer (e.g. information on the screen size being used by the customer) or indirectly, from the browser used, the type of device being used to access the page. On the other hand, the World Wide Consortium (W3C) is developing a standard to discover the capacities of the customer and the preferences of the user [13].
  - *The User's Suggestions*. In this case, as in the model for the characteristics of the user, questionnaires, or templates allowing the user to personalise the page, can be implemented that will help to define the characteristics of the user's access device.
  - Furthermore, it is necessary to generate specific *tools* to determine the state of the network. The technology of intelligent agents is one of the most interesting and most important tools that can be used.

- *Mechanisms of Adaptation* [5],[13]:

  - *Data Compression Algorithms*: these present summaries of the text to be shown, thumbnail images of the available material on the page, etc.
  - *Data Transformation Algorithms*: these modify the format of the multimedia material presented e.g. lowering the resolution of the images or modifying a video file to show a succession of still images.
  - *Data Classification Algorithms*: these classify the objects that are shown on a page, giving them levels of importance in order to decide whether these objects will be shown or not.

3.3 State of Load on the Server

The problem of server overload, [12-13], has also been one of the considerations taken in the model suggested for our SHAAD, since the capacity at any given moment

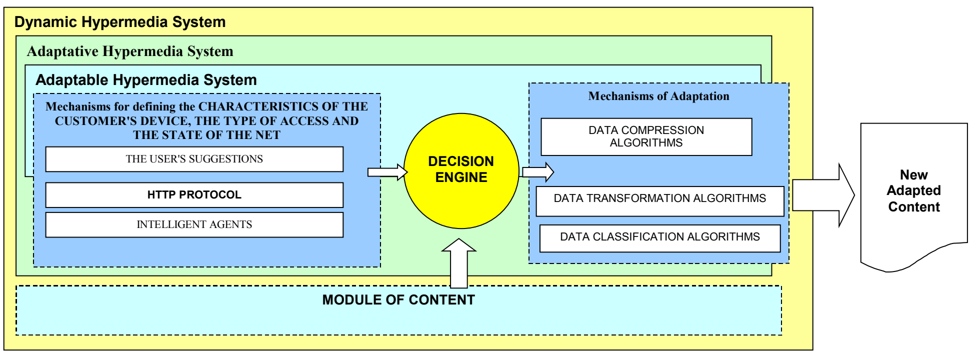


Figure 2 – Model for Content Adaptation According to Access to the network.

to deal with the quantity of incoming requests by users, will mainly depend on the server and on its current state.

In principle, these servers are sized in such a way that they can deal with all the demands made upon them. However, in practice, there are frequent cases in which, as users, we find it impossible to download a page due to the overloading on the server at that moment and to the fact that the server finds it impossible to attend to all the requests. As this overloading can, and does, take place, we need to find solutions for such situations.

The processing of the hypermedia content on the server can be carried out by *on-line or off-line transformations*. An *on-line or dynamic* process is one which is carried out when the load on the server allows the dynamic generation of the web page content from the unit elements that form it. In the *off-line or static* process, different versions of the hypermedia content are available, in different formats and qualities, in order to be able to select one of those versions when the overload conditions make it necessary. From this point on and taking into account the state of the variables mentioned in points 4.1 and 4.2, the resultant hypermedia content is adapted by means of the other mechanisms of adaptation.

The Module of Content interacts directly with the Decision Engine, to which it delivers the version of content that arises after considering the state of load on the server. This decision engine, with the hypermedia content selected and the state of the other variables defined, implements the mechanisms of adaptation to the degree that it decides is necessary.

In this case, we need to describe two modules (figure 3):

- *Mechanisms for the definition of the state of load on the server*. The tools evaluated to carry out the aforementioned definition range from the web server's own tools to the use of intelligent agents technology implemented in Java.
- *Module of Content*. This is responsible for generating dynamically, or else selecting from the static versions, the most suitable content.

  It is made up of the following blocks:

  - *Dynamic generation from unit elements*. It generates the new page dynamically from the elements that compose the original page. The Content Analyser (fig.3) is

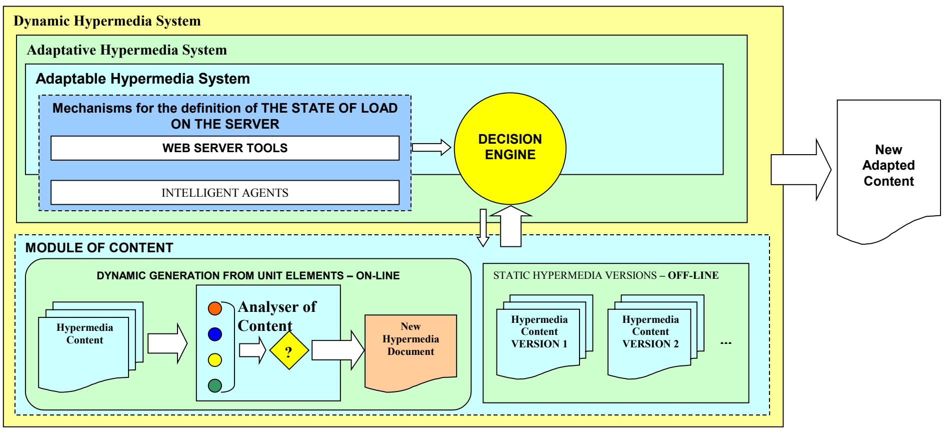


Figure 3 – Model for Content Adaptation according to State of Load on the Server.

a tool developed in XML and Java. The goal of the analyser is to convert a document from a traditional HTML format to an XML format and, with the advantages that the XML format gives us, to sort out the objects that make up the original document. Once this structure has been obtained, by means of meeting certain conditions and through the intervention of the decision engine, it selects the convenient objects to form the new HTML document.

- *Static hypermedia versions*. In this block, versions of the web pages, with different qualities in terms of format and content, are available. These static versions are available for the situations in which the Decision Engine decides to use one of these versions in the case that the load on the server so requires.

4 Architecture of the SHAAD

Through the models suggested in section 3, we have tried to explain by homogeneous blocks the obtaining of a single model. In the same way, Figure 4 represents our SHAAD proposal, with all the variables of adaptation considered taken into account.

For each of the different schematised modules we have given a corresponding analysis throughout the present paper. On the other hand, we established the prime importance, in our research, of the inclusion of the state of load on the server when considering a dynamic generation of content (on-line). A priori, this seems to be the best way of implementing our goals, but we can not lose sight of the fact that this solution cannot be the only one and the processes of our decision engine will, in the end, have to include mixed solutions, in which a combination of the states of the defined variables of adaptation are considered.

5 Conclusions

We have presented in this paper the SHAAD model (System which is Hypermedia Adaptable, Adaptative and Dynamic) for the dynamic adaptation of content. This model tries to cover, from different points of view, the wide range of related works on hypermedia adaptation of content. Thus, through the analysis of different techniques

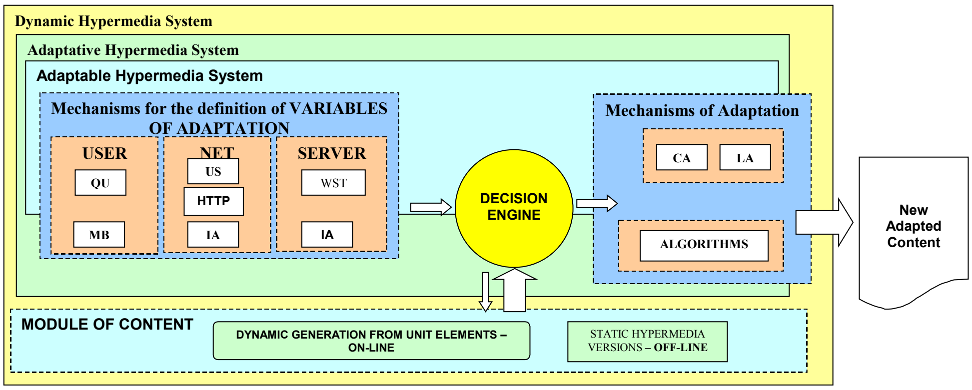


Figure 4 – Final Model for a SHAAD

for defining different variables (characteristics of the user, characteristics of the customer's access device, type of access, state of the network and load on the network) we have tried to encompass the different points of view and to define, through a single model, our starting point for the adaptation of content.

Some of the definition tools considered here are at the implementation stage. E.g., our Content Analyser, implemented in XML and Java, through which an intelligent selection of the objects that form a page can be carried out. Also, the tools investigated for an efficient testing of the conditions of the network that are being evaluated through the same http protocol as well as intelligent agents implemented in Java.

Our strictly modular model will allow us to work on the wide range of aspects we have analysed and in this way, help us to suggest partial solutions to a problem that globally can be excessively complex due to the many variables that have to be analysed.

References

[1] Perkis, A.: "UMA Project – Universal Multimedia Access from wired or wireless systems". NTNU – Department of Telecommunications

[2] Raatikainen, K., Hippeläinen, L., Laamanen, H., Turunen, M.: "Monads – Adaptation Agents for Nomadic Users". World Telecom '99

[3] Misikangas, P., Mäkelä, M., Raatikainen, K.: "Predicting QoS for Nomadic Applications Using Intelligent Agents". Impact'99 Workshop.

[4] Campadello, S., Helin, H., Koskimies, O., Misikangas, P., Mäkelä, M., Raatikainen, K.: "Using Mobile and Intelligent Agents to Support Nomadic Users". 6th International Conference on Intelligence in Networks (ICIN2000), 2000, Bordeaux, France.

[5] Ma, W-Y., Bedner, I., Chang, G., Kuchinsky, A., Zhang, H.: "A framework for adaptive content delivery in heterogeneous network environments". Hewlett-Packard Laboratories.

[6] Yang, Y., Chen, J., Zhang, H.: "Adaptive Delivery of HTML Contents". 9th International World Wide Web Conference – The Web: The Next Generation, Amsterdam, 2000.

[7] De Bra, P.: "Design Issues in Adaptive Web-Site Development". Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW

[8] Billsus, D.: "Improving User Model Acquisition from Labeled Text Documents". Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW.

[9] Brusilovsky, P.: "Adaptive Hipermedia". User Modeling and User-Adapted Interaction 11: 87-110, 2001, Kluwer Academic Publishers, Netherlands

[10] Marinilli, M., Micarelli, A., Sciarrone, F.: "A Case-Based Approach to Adaptive Information Filtering for the WWW". 7th International Conference on User Modeling, Banff, Canada, 1999.

[11] Shaha, N., Desai, A., Parashar, M.:"Multimedia Content Adaptation for QoS Management over Heterogeneous Networks". International Conference on Internet Computing (IC 2001), Nevada, USA, June 2001.

[12] Abdelzaher, T., Bhatti, N.: "Web Server QoS Management by Adaptive Content Delivery". In Int. Workshop on Quality of Service, June 1999.

[13] Abdelzaher, T., Bhatti, N.: "Web Content Adaptation to Improve Server Overload Behavior". The 8th International World Wide Web Conference, Toronto, Ontario, Canada, 1999.

[14] Oppermann, R., Rashev, R., Kinshuk: "Adaptability and Adaptativity in Learning Systems". Knowledge Transfer (Volume II) (Ed. A. Behrooz), 1997, pAce, London, UK, pp173-179 (ISBN –900427-015-X)

How to Facilitate Navigation Planning in Self-directed Learning on the Web

Akihiro Kashihara, Shinobu Hasegawa, and Junichi Toyoda

The Institute of Scientific and Industrial Research, Osaka University
8-1, Mihogaoka, Ibaraki, Osaka 567-0047, JAPAN
kasihara@ai.sanken.osaka-u.ac.jp
http://www.ai.sanken.osaka-u.ac.jp/thome2/kasihara/index-e.html

Abstract. How to help learners plan a navigation process on the Web is an important issue in web-based learning/education. Our approach to this issue is to allow the learners to preview a sequence of web pages as navigation path. In this paper, we introduce an assistant system that enables learners to plan a navigation path in a learner-centered way before navigating hyperspace provided by web-based learning resources. It generates a navigation path preview in accordance with their navigation planning context. Since the navigation path preview gives the learners in advance an overview of the contents to be learned, their learning in the hyperspace can be improved.

1 Introduction

There are currently a large number of hypermedia/hypertext-based resources available for learning/education on the Web. Such existing web-based learning resources provide learners with hyperspace where they can navigate the web pages in a self-directed way to learn domain concepts/knowledge. The self-directed navigation involves making a sequence of the pages navigated, which is called navigation path [7,10]. It also involves constructing knowledge from the contents embedded in the navigated pages, which would enhance learning [3,12]. The knowledge construction process is influenced by the navigation path. Making a navigation path is accordingly an important process of the self-directed learning in hyperspace [7,8].

On the other hand, learners often fail in making a navigation path since they would make diverse cognitive efforts [12]. The self-directed learning in hyperspace requires them to comprehend the contents of web pages they have visited, and concurrently to monitor their own navigation process such as planning the navigation path to be followed, which can be viewed as meta-cognitive activities [3,7]. The navigation monitoring particularly holds a key to success in the self-directed learning. However, it is difficult to maintain the navigation monitoring since the learners would focus on comprehending the contents of the visited pages [7]. They accordingly often reach an impasse. Although such navigation problem has been a major issue addressed in educational hypermedia/hypertext systems [2,12], the important point towards resolving the problem in the context of self-directed learning is how to help learners monitor their navigation process [7,8].

In this paper, we address the issue of how to help learners plan a navigation process in self-directed learning with web-based resources [8,11]. Our approach to this issue is to provide learners with space for the navigation planning apart from hyperspace, in which learners can preview a sequence of web pages as navigation path. The navigation path preview enables them to plan a navigation path in a learner-centered way before navigating the hyperspace. Since the planned navigation path also gives them in advance an overview of the contents to be learned, their learning can be improved [1].

This paper also demonstrates an assistant system, which is an improved version of the original [8,11]. The assistant system consists of path previewer, page previewer, and hyperspace map [11]. The page previewer extracts information attached to some HTML tags in a page, which learners select in the hyperspace map, from the HTML document. The information can be considered representative of the page. The page previewer displays it as page preview. Although the information to be extracted depends on the topic on which the learners focus in planning the navigation path, the original page previewer generates the page preview regardless of the focal topic. In the improved version, the page previewer identifies the focal topic with the contextual information, and generates a more proper page preview. The path previewer makes a sequence of previewed pages, and displays it as navigation path preview. These facilities help learners plan which page to visit and a navigation path without visiting web pages in hyperspace.

2 Navigation Path Planning

2.1 Framework

Our framework for supporting navigation planning where self-directed learning in hyperspace is done in two spaces. These are the space for planning the navigation path based on a learning goal and hyperspace for executing the plan. In the path planning space, learners preview web pages, and plan which page to visit and the sequence of pages to be visited so that the learning goal can be achieved. In the hyperspace, they are expected to navigate web pages according to the planned navigation path. The path planning and navigation are repeated during the self-directed learning. The distinction between navigation path planning and navigation allows learners to raise awareness of monitoring their navigation process, and to focus on comprehending the contents of web pages.

Demonstrating the assistant system, which has been implemented as plug-in tool for Microsoft Internet Explorer 4.0 or higher, we next explain how to support the learner-centered navigation path planning and navigation in hyperspace provided by web-based learning resources.

Let us first consider what kind of information should be presented as preview of navigation path. Although spatial maps of web-based learning resources are necessary for considering navigation paths, the maps alone may be insufficient for learners to plan the paths [4]. This suggests the necessity to give learners an informative overview of the contents. We accordingly introduce a page previewer that extracts keywords, sentences, or images to be considered representative from a web page to display them as a preview of the page. In addition, we introduce a path previewer that makes a sequence of the pages learners have previewed.

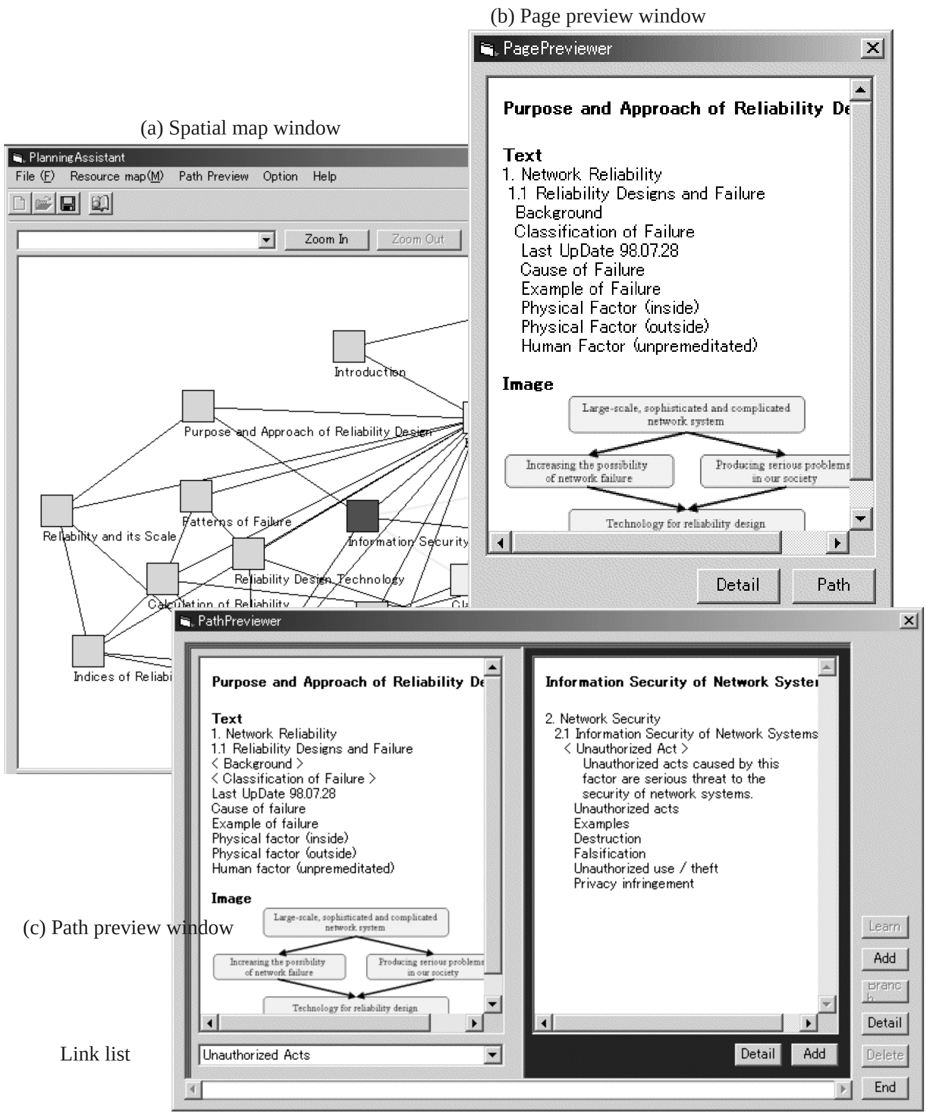


Fig. 1. User Interface for Navigation Planning.

On the other hand, web page generally includes several topics. However, every topic does not always need to be previewed when learners plan a navigation path.

Which topic should be principally previewed depends on the topic learners intend to focus on in the page. We call it focal topic. The path previewer accordingly identifies the focal topic from the navigation planning context, and the page previewer generates a preview of the page according to it.

Fig. 1 shows a user interface of the assistant system where learners can preview a navigation path. The system is composed of spatial map, page previewer, and path previewer. As shown in Fig. 1(a), the spatial map represents the hyperspace of a web-based learning resource selected by learners as a network of nodes corresponding to the web pages. It is automatically generated and displayed in the map window when they select the learning resource. The spatial map represents the web pages only within the same web site where the homepage selected by the learners is located. The links from the site to others are omitted. Nodes in the spatial map are tagged with page titles indicated by title tags in the HTML documents.

By double-clicking any node in the spatial map, learners can have an overview of the web page corresponding to the clicked node in the page preview window as shown in Fig. 1(b), which is generated by the page previewer. When the learners decide the starting point of the navigation path, they can trigger the path previewer, which changes the page preview window into the path preview window as shown in Fig. 1(c). The path previewer constructs a sequence of previewed pages starting with the current page. The path preview window has a link list, which includes anchors of the links the current page contains. Selecting any one from the list, they can have a preview of the page, to which the selected link points, next to the preview of the current page. Since the selected anchor seems to indicate the topic on which they intend to focus, the path previewer identifies the focal topic from the selected anchor to generate the page preview. The learners can then put the previewed page into the sequence, making a navigation path.

The learners are next expected to follow the navigation path to visit the web pages. In order to help them follow the path, the system provides a navigation controller, and displays the first page of the navigation path preview in the web browser. Using the navigation controller, the learners can browse the next page on the navigation path in the browser. However, they do not always need to follow the path. They can explore web pages, which are not included in the path. The navigation controller puts a warning icon when the learners run off in the path. When they also want to change or cancel the navigation path during navigation, they can return to the navigation path planning. In this way, learners are expected to repeat the navigation path planning and navigation to accomplish learning in hyperspace.

In the following, let us explain the page previewer and path previewer in detail.

2.2 Page Previewer

In this section, let us explain how to generate a page preview regardless of learners' focal topic. The page previewing in accordance with the focal topic is described in the next section.

The important point to generating an overview of a web page is how to extract information representing the contents of the page. Assuming that such information is located within representative HTML tags such as *Title* and *Heading* tags, the page previewer extracts words, sentences, or images indicated by these tags to display them as page preview. We heuristically consider that such assumption is valid in web-based resources for learning/education.

The information extracted from a web page is classified into the textual contents, images included, and the links out of it. As for the textual contents, the page previewer searches for *Title*, *Heading*, and *Font Size/Color/Face* tags, and extracts words or sentences attached to the tags. As for the images included in the page, the page previewer searches for *Img* tags in the HTML document, and displays one image whose file size is the largest. As for the links out of the page, the page previewer searches for *A href* tags in the HTML document to extract source and destination anchors of the links. These anchors are used by the path previewer.

If learners cannot grasp the page preview, they can also push *Detail* button at the lower part of the page preview window to look at the full contents of the page with web browser. However, this operation is not recommended in navigation path planning. Pushing *Path* button, the learners can also start the path previewer, which changes the page preview window into the path preview window.

2.3 Path Previewer

The path previewer sequences the pages previewed, which are put in order by learners. The order of the previewed pages represents a navigation path. It also displays a link list as drop-down menu, which is generated by the page previewer. This list includes source anchors of the links that the current page contains. The learners can select any one from the list to have a preview of the page, which is indicated by the destination anchor of the selected link, next to the preview of the current page. The page preview generation consists of the following two processes that are executed in the path previewer:

(1) Identifying the topic on which learners focus in planning, and

(2) Identifying information to be previewed according to the focal topic.

The path previewer first identifies the focal topic with keywords included in the anchor that learners select from the link list of the current page for previewing the next page.

By using *Heading* tags in the HTML document of the page to which the selected anchor points, the page previewer next divides the document into sections, which are indicated by the tags, and then identifies the section to be previewed. In the HTML document, *Heading* tags describe the topics of the sections. These tags generally constitute a hierarchy as shown in Fig. 2. The HTML document accordingly has a hierarchical structure of the topics described by the *Heading* tags. Following the hierarchy, the page previewer identifies the section whose *Heading* tag includes the focal topic, as the sections to be previewed. If no *Heading* tag includes the focal topic, it finds in which section keywords representing the focal topic appear most frequently and selects this as the section to be previewed.

The page previewer next extracts information to be previewed to generate the adaptive page preview. It first extracts not only information attached to the *Heading* tag of the identified section, but also information attached to *Title* tags, and the ancestor/descendant *Heading* tags. As shown in Fig. 2, for example, the page previewer extracts information attached to *Title*, *Heading* tags indicated by *H1-2*, *H2-2*, *H3-1*, and *H3-2* when *H2-2* is the tag of the identified section. This enables the page preview to preserve the hierarchical structure of topics embedded in the page. The page previewer then extracts key information from the identified section in the similar way as described in 2.2. The path previewer also generates the link list including anchors of all links in the page, in which the anchors included in the section previewed take precedence.

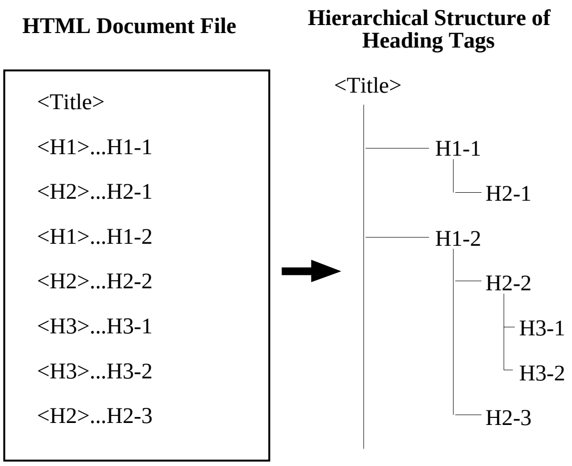


Fig. 2. Hierarchical Structure of HTML Document.

If the learners want to add the previewed page into the navigation path, they are required to push *Add* button in the right corner of the path preview window. If they do not add it, they try to select another link from the link list again. Repeating the page previewing and the *Add* operation, the learners can make a sequence of previewed pages.

The learners can also delete any page in the navigation path by pushing *Delete* button, make branches of the path by pushing *Branch* button, start making new navigation path by pushing *End* button. They can then start navigation according to the navigation path plan with the navigation controller, which appears as *Learn* button is selected.

2.4 Example

Let us demonstrate the navigation path planning with a simple example. In this example, we consider a learner whose learning goal is to explore factors reducing the reliability of computer networks. He/she first tries to plan a navigation path for achieving the goal from the web page titled *Purpose and Approach of Reliability Design*. The starting page in the navigation path is generated in the same way as described in 2.2, which is shown in Fig. 1(b). Next, he/she selects the anchor *Unauthorized Acts* from the link list in the path previewer shown in Fig. 1(c). In this case, he/she seems to focus on unauthorized acts in the web page *Information Security of Network Systems*, to which the anchor points. The path previewer accordingly identifies the focal topic with *Unauthorized Acts*, and then generates the adaptive preview of the page as shown in Fig. 1(c). It principally includes information related to the focal topic.

3 Related Work

Current navigation aids, which support the navigation planning, can be divided into two types, which provide global and local view of hyperspace. As representative aids for global view, there are spatial and concept maps. Spatial maps represent nodes and links that compose the structure of hyperspace [5]. Concept maps consists of nodes and links representing the structure of domain concepts to be learned, which nodes are mapped on the corresponding pages in hyperspace [6]. In both spatial and concept maps, nodes are tagged with their titles, which are intended to represent the contents of the corresponding pages. In concept maps, links are also tagged with descriptions representing the semantic relationships between the nodes. Although the spatial and concept maps can provide learners with a space, apart from hyperspace, for considering navigation paths, the tagged information may be insufficient for learners to plan a navigation path [4]. In addition, web-based learning resources mostly have no maps.

As representative navigation aids for local view, there is adaptive hypermedia, which supports navigation in hyperspace by adapting the contents of the page to be visited next, annotating pages and links to be visited next, etc. [2], and visual preview which shows a thumbnail of the next page [9]. The main purpose of these aids is to help learners select the page next to the current page. However, the thumbnail may be unsuitable for having an overview of the page in case the contents of the page are invisible in the thumbnail. The adaptive navigation aids call learners' attention to comprehending the contents of pages, but decrease their awareness of navigation path planning, which is required in the self-directed learning. In addition, the adaptive aids are not always applicable to existing web-based learning resources since they generally have no clear description of semantic relationships among web pages, which is indispensable for executing the adaptation. In order to apply the adaptive aids to existing web-based learning resources, it is necessary to analyze semantic structure of the domain concepts/knowledge beforehand. This is not an easy task.

In this paper, on the other hand, we address the issue of how to help learners navigate hyperspace provided by existing web-based learning resources. Considering that making a navigation path is an important process of self-directed learning in the hyperspace, we particularly focus on how to help learners plan a navigation path. The approach to this problem is to generate the adaptive preview of navigation path, which can be viewed as the middle of the global and local views. Such navigation path previewing is an important aid for self-directed learning in hyperspace.

4 Conclusions

This paper has described a navigation planning aid for self-directed learning in hyperspace provided by web-based learning resources. The key idea is to provide learners with adaptive preview of navigation path to help them plan their navigation process, which is an important process of the self-directed learning. The adaptive previewing enables them to plan the navigation path in a proper way before navigating hyperspace. The self-directed learning can be also improved since the distinction between navigation path planning and navigation allows them to focus on comprehending the contents of web pages navigated in the hyperspace.

This paper has also demonstrated an assistant system executing the adaptive navigation path previewing. It allows learners to plan which page to visit and make a navigation path plan without navigating hyperspace.

In future, we need a more detailed evaluation of the adaptive navigation path previewing with the system. We would also like to improve it according to the results.

References

1. Ausubel, D. & Fitzgerald, D. The role of discriminability in meaningful verbal learning and retention, Journal of Educational Psychology, 52, pp.266-274 (1961).
2. Brusilovsky, P. Methods and Techniques of Adaptive Hypermedia, Journal of User Modeling and User-Adapted Interaction, 6, (1996) 87-129.
3. Cunninghan, D.J., Duffy, T.M., and Knuth, R.A. The Textbook of the Future, in McKnight, C., Dillon, A., and Richardson, J. (eds): HYPERTEXT A Psychological Perspective (1993), Ellis HorwoodLimited, 19-49.
4. Dillon, A., McKnight, C, and Richardson, J. Space-the Final Chapter or Why Physical Representations are not Semantic Intentions, in McKnight, C., Dillon, A., and Richardson, J. (eds): HYPERTEXT A Psychological Perspective (1993), Ellis HorwoodLimited, 169-191.
5. Domel, P. WebMap - A Graphical Hypertext Navigation Tool, in Proc. of Second International WWW Conference (1994),
6. Gaines, B.R. and Shaw M.L. G. WebMap: Concept Mapping on the Web, in Proc. of Second International WWW Conference (1995),
7. Hammond, N. Learning with Hypertext: Problems, Principles and Prospects, in McKnight, C., Dillon, A., and Richardson, J. (eds): HYPERTEXT A Psychological Perspective (1993), Ellis HorwoodLimited, 51-69.
8. Kashihara, A., Suzuki, R., Hasegawa, S., and Toyoda, J. A Navigation Path Planning on the Web, Proc. of IEEE International Workshop on Advanced Learning Technologies (2000), 25-26.
9. Kopetzky, T., and Muehlhaeuser, M. Visual Preview for Link Traversal on the WWW, Proc. of WWW8 (1999), http://www8.org/w8-papers/4b-links/visual/visual.html.
10. Nielsen, J. The Art of Navigating Hypertext, Communication of the ACM, 33, 3 (1990), ACM Press, 297-310.
11. Suzuki, R., Hasegawa, S., Kashihara, A., and Toyoda, J. A Navigation Path Planning Assistant for Web-based Learning, Proc. of World Conference on Educational Multimedia, Hypermedia & Telecommunications (2001), 851-1856.
12. Thuering, M., Hannemann, J., and Haake, J.M. Hypermedia and Cognition: Designing for Comprehension. Communication of the ACM, 38, 8 (1995), ACM Press, 57-66.

Interactive Tutoring Model Using Information Cycling

Jan Smid[1], Petr Svacek[2]

[1] Morgan State University
1700 East Cold Spring Lane
Baltimore, MD 21251
e-mail: jsmid@jewel.morgan.edu
[2] S-DATA, Prague, Czech Republic

Abstract. The goal of this paper is to describe a user oriented system framework that provides user tutoring and testing. This tutoring system is suitable for classroom use. It helps to introduce definitions and simple relations among concepts. The amount of information delivered by the system varies for different users. The way information is delivered and the amount of information can be characterized by a set of parameters. The parameters can be used to minimize the time of the learning process. We chose for our experiments only parameters that describe the number of presentations that allow the user to pass a test. The system was developed using PHP and XML programming environment. We present preliminary classroom results along with statistical simulations of the system. Typical application domains were simple procedures from discrete mathematics, probability, computer science terminology, statistics and language elementary vocabulary.

1 Introduction

In this paper we will deal with a description of an information cycling system. The task of the intelligent tutoring system (ITS) can be briefly characterized as a problem of instructional material sequencing, diagnosing and responding to (specific) user requests. In the ideal case, the material is constructed on demand, and intelligent re-mediation can be performed at any level. The material sequencing and diagnosing has two aspects. A logical mistake should be corrected by providing the appropriate information. The strategy of presentation may depend on the user. The extent to which the content and the presentation strategy should be controlled by the user profile is one of the crucial question of ITS. Weber [7] reports moderate to no effect of single adaptation technique.

In our Information Cycling and Diagnosing System (ICD) system, each unit is presented by cycling elements of information, in the form of definitions, statements and quizzes. A final test allows the user to leave the unit and to pass to the next one. A user, and its path through the system, can be characterized by a set of parameters. We have limited our attention to only two adaptive parameters for each information unit. These parameters control the number unit cycles and the unit exit criterion. It means that each user is associated with a recommended number of cycles for each unit. The other meaningful parameter is the amount of information presented at a time. Parameters that describe the scoring information to the user have important psychological value and should be analyzed rigorously. The knowledge domains we have investigated were algorithms for solving linear algebraic equations using the Gauss-Jordan method (GJ), theoretical computer science statements such as the Euler Circuit Theorem and elementary French vocabulary. Our goal is to compare tutoring mini-systems that can be effective in the introductory college classroom. We are interested in using interactive educational tools to introduce key vocabulary and concepts. From our classroom experience, we have seen that students can benefit substantially from interactive information cycling. The targeted audience was the lower level undergraduates. When students are exposed to vocabulary training before an instructor-led lesson, difficult concepts can be introduced more easily. This idea has been applied in classrooms in which the majority of the students are required to take mathematics or computer science but do not need to understand deeper principles such as proofs. This system is useful for students who need to master an introductory understanding of the basic vocabulary and concepts. Even within the same classroom, there are differences in presentation needs. The differences among users in such a classroom setup is the skill level rather than the background level. In accommodating a variety of presentation needs, the user oriented system reflects skill level in both reducing redundancy for more skilled users and maintaining appropriate level for those less skilled. The targeted users for our ICD experiments were undergraduate students of mathematics and computer science classes at Morgan State University. Five classes, ranging from 10-20 students, participated in the experimentation.

1.1 Flexible and User Oriented Systems

Flexible systems let a user to proceed to another unit as soon as a test question(s) is correctly answered. If we use other exit criterion, a basis for this criterion must be established. The purpose of this criterion can be to prepare the user for information that requires the unit in question as a prerequisite. We consider systems that keep information about which units were passed as flexible systems. We name user-adaptive systems as systems that utilize more than merely pass-fail data. The defining characteristics of an user-adaptive system is its optimal utilization of the user profile data. The user profile is established through collection of data intrinsic to the user such as the number of cycle necessary to memorize a set of information. The profile data is acquired during a user session in which the individual is classified in a category based on the performance data. One type of adaptivity is based on the actual knowledge of units (flexible systems). Type two are user-adaptive systems based on user intrinsic features. In literature, the adjective adaptive usually refers to type two.

1.2 Evaluation of User Path

A number of systems for learning have been developed. The task of measuring complexity of adaptive and non-adaptive systems has been addressed at the Empirical Evaluation of Adaptive Systems Workshop [1], [8]. While some of these systems use adaptive parameters, they are not clearly evaluated in the literature against non user-based systems. Regardless of the nature of those parameters, the goal of a user oriented system is to lower the complexity of the user path, using the user model. The complexity of the user behavior as a path between information gathering and knowledge testing is difficult to evaluate. Therefore, in this paper we restrict our study to a simplified description where the cycle path is a record of number of cycles per each unit. The goal is to analyze and evaluate the difference between the number of operations of the user adaptive systems (that usually require no more operations than non-adaptive) and flexible systems. In some instances, such as GJ elimination, the number of operations estimated based on the user profile can be significantly higher and closer to user's needs. Consequently, the user adaptive system is more efficient because the user passes only one exit test whereas the non-adaptive system user must remediate through entire units including the exit test. The complexity of implementing a user oriented system is much higher compared with implementing flexible systems and must be always considered in the overall analysis.

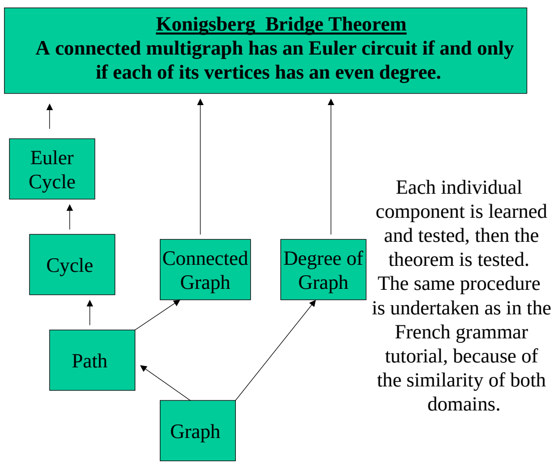


Fig. 1. The Euler theorem domain

2 User Path Optimization

Flexible interactive systems provide information until test questions are successfully answered. In the ICD system, the user profile specified number of cycles is provided. Depending on the nature of the mistake, the user either repeats an information cycle at the same unit or is returned to some of parent (prerequisite) units. One example of a flexible system is the Gauss-Jordan elimination training tool experiment [2]. In this experiment, the user is offered GJ elimination steps as a quiz until a criterion is satisfied. No user modeling is required because the number of cycles is added one by one as necessary. However, if a learning goal contains the GJ procedure as one of its components then a user model may be required. More extensive training may be needed for the units that have the GJ as a prerequisite. In an average situation, one unit has several prerequisite units i.e. a typical graph of a recommended path is not a mere series of units, but a directed acyclic graph [3]. Therefore, the test questions (more precisely, user's wrong answers) should also point to a particular prerequisite knowledge. The exit test involves a task that requires all prerequisites. For example the ability to correctly apply the Euler theorem can be used as an exit test. Tests for prerequisites involve definitions of graph, connected graph, cycle, Euler cycle and degree. For a typical situation, see Euler theorem diagram on Fig.1.

This approach can be applied for different domains. The similarity of domains such as Konigsberg bridges and French grammar is based on the representation of information in the form of a tree where each vertex represents a statement, definition, example, sentence or a sentence component. We will consider simplified situations which are still pedagogically important. In the simplest scenario, the micro-strategy within a unit cycles items of the unit (definition, quizzes) until a criterion is satisfied assuming that each definition and quiz have been seen once successfully. The macro-strategy is to follow the network (recommended sequence) of units, which in some instances (as in the case of return to a parent unit described above) may be repeated. Each cycle of the unit is recorded. The numbers of cycles generate a sequence which we call the cycle path c_1, c_2, c_3,...,c_n. The cycle path is not the actual path through the unit items. We ignore this potentially useful information in our simplest model because we feel the number of cycles itself is the most relevant information. A set of definitions, statements and quizzes is used in each unit. The number of included components is higher than the number of actually used components for each user. This is to assure sufficient variability of presentations.

Case 1: First, we will assume that units are independent. Each unit is self-contained and consists of several definitions, statements and tests. Units are visited by the user in a sequence. Each user is assigned an initial number of cycles for each unit. This initialization is based on a pre-test or on a constant sequence $(1, 1, 1, .., 1)$, i.e. each unit is cycled one time. A unit exit test follows. When the exit test is failed, one cycle is added to the cycle path vector and the unit is presented. The user type is a cycle path vector $(c_1, c_2, c_3, .., c_n)$, where n is the number of units. The user types are initialized to constant vectors

$(i, i, i, .., i)$, $i = 1, 2, 3, \ldots$. The i represents the best estimate of the number of cycles. This number adjusted by the system based on the user answers. Using one of the presentation strategies, we repeat items within the same unit, never returning to the previous unit (in this simplest scenario). Examples of presentation strategies are:
(1) present definitions/statements and then ask question in the multiple choice form,
(2) present definitions/statements and then ask questions in the form of a missing word
The distance from a new partial or complete cycle path to a known user types is measured and the cycle path is assigned to the closest user type. This membership determines the number of cycles needed to complete the given unit. The unit is completed when the exit criterion is met. The criterion is either defined by an exit test or by a score level. All cycle paths are clustered and resulting clusters are used as user types. The variations of the cycle path vector components $(c_1, c_2, c_3, ..., c_n)$ indicates the degree of difficulty of a unit. In the ideal case of uniformly authored units, the components would be constant for each user type. If there is more than one presentation strategy, we can utilize all strategies in order to minimalize the number of operations. Assuming that we have more than one strategy available, we can alternate them and match sub-cycle paths with user types.

Case 2: We assume that one unit depends on a set of other units-prerequisites. Dependent units can be visualized as children nodes of parent nodes in a graph. For dependent units, we made a requirement about test questions. Each test questions about a child node is phrased so that it points to the remediation unit(s). The frequency of visits of prerequisite units is used for modifying the number of cycles estimated from individual units.

A user can be either incorrectly under or over-classified. If a user is incorrectly classified to require not enough cycles a penalty is paid due for additional cycles and exit tests when remediation occurs. The penalty for finding the correct remediation also applies. The over-classification produces a penalty for redundant exits test and operations within a unit. The main objective of the cluster analysis is the classification of users. This means that each future user can be, after several units, classified into a certain class. Likewise, his path through the system can be optimized. By optimal, we mean that the cost of the user's learning counted in the number of operations (cycles) and unsuccessful tests is minimal for a given teaching strategy and stopping criterion.

3 Examples

One of the learning strategies allows the user to return to the prerequisites. This strategy assumes that authoring provides test questions that allow us to identify the cause of incorrect answer. We assume that each unit is in the form of a small tree with 3 or 4 parents. A good example is the Euler Circuit Theorem again, Fig.1.

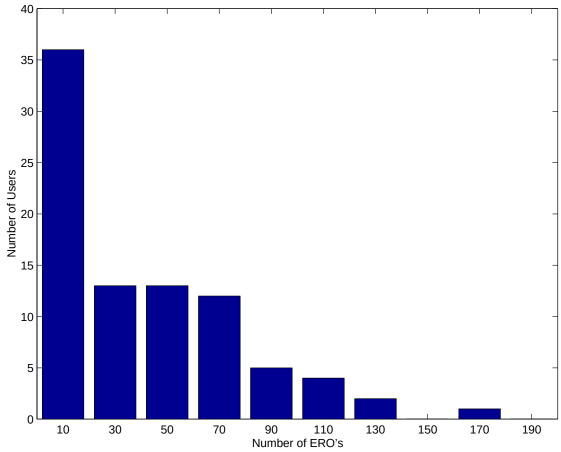


Fig. 2. Numbers of cycles required by users, in Gauss-Jordan experiment

The system can ask the user to list the properties of a presented graph. Possible answers might have the following outcome:

1. An incomplete list of properties (degree of vertices, connected graph) indicates a definition for remediation.

2. An incorrect statement about the existence of an Euler cycle indicates that more work at the theorem unit level should be done.

3.1 Types of experiments

The current system is concerned with several types of experiments. The first style is based on abstract presentations and example based presentations. The abstract presentation provides more compressed information than the example based presentation. The goal of the second type of experiments is to collect sequences of the number of cycles. These sequences are then clustered into several typical sequences. The third type deals with dependencies of units and the goal is to collect the number of cycles under the assumption that some units may have one or more prerequisite units.

3.2 Numerical experiments

We have performed classroom experiments and simulated users mathematically. The number of users was chosen to be approximatelly the same for each cluster. Data simulation were provided by random variables based on heuristic from in class experiments either with system that collected the data or with stateless systems (no memory).

The GJ experiment, Fig. 2, shows the variation in the number of cycles for groups of users. More details on this problem can be found in [1]. The significance of this experiments consists of demonstrating that a user can require a dramatically more prerequisite information than the average. In this experiment, the goal is to be able to apply the GJ method for solving a system of linear algebraic equations. The prerequisite for learning the GJ method is that the user is trained to do only one elimination step at a time. In the numerical experiments of the user we generated cycle paths. In the simplest scenario (no remediation using previous units) the penalty for incorrectly identified numbers c_i is $op * (c_i - c)$ when the correct number of cycles c is smaller than c_i, and $p * (c - c_i)$ for c greater than c_i. The term op stands for the number of operations for one unit and p is the penalty for additional unit exit tests.

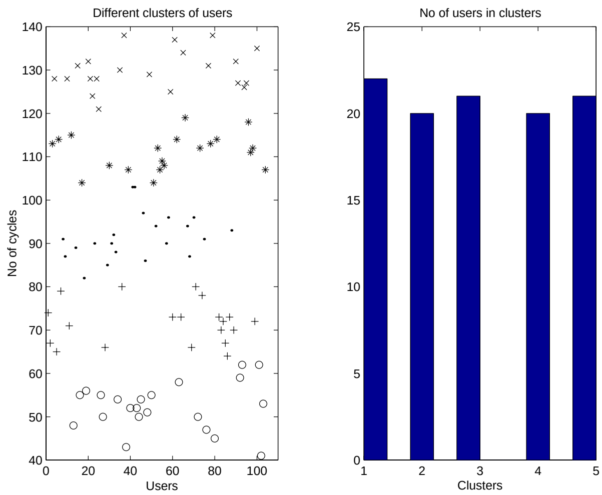


Fig. 3. Clusters of users

3.3 Identification of user types, simulation results

In the numerical simulations of the users, we defined a sequence of units and then generated a number of cycle paths with different probability distributions for passing each unit. For each user, these probabilities were mutually dependent, and slightly increasing after each additional information (cycle) provided to the user. This process created cycle paths that were clustered. Clustering algorithm used a variant of the K-means procedure, where the distance was defined as a weighted Euclidean distance in the space of paths. The optimal number of clusters was selected to minimize the sum of all distances of paths from corresponding means. In order to reduce the final number of clusters, the criterion was penalized with the Schwarz's BIC penalty term. Figure 3 shows the result of one of simulation experiments, which selected 5 clusters of cycle paths. The left figure shows the distribution of the total number of path cycles and their classification. The boxplots of Figure 4 compares, for all 5 clusters, the distribution of numbers of cycles used for one of the units. All the numbers of cycles differ. These distributions, along all units, actually split the set of paths into 5 subsets of cycle path. Each new user (after passing several units) can be classified on the basis of weights measuring the distance to different clusters. In subsequent units, the user is provided with the information (the number of cycles) optimal for his class. The optimal number of cycles can be set, for instance, as 75% quantile of the corresponding distribution. In the boxplot, this number corresponds to the upper bound of the box. An incorrect classification of a user can increase either the time (number of cycles) spent at the unit when the unit has already been mastered, or it can increase the number of attempts needed to pass the final unit test while the knowledge of the user is not yet sufficient. In the simplest scenario (no remediation), the overall penalty for incorrectly identified user is the sum of penalties for all units. Thus, as the model is described with the aid of probability distributions (and identified by methods of mathematical statistics), we cannot avoid the error of the first kind, i.e. even if a user is classified properly, he will need more operations than recommended for his type (actually, in the example described the probability of such an error is 25%). Nevertheless, the mean cost

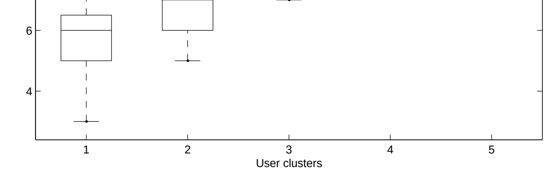


Fig. 4. Example of distribution of cycles per one unit, for different clusters of users

of such a mistake is, as a rule, much smaller than the cost of incorrect classification. Moreover, the classification is a dynamical process and the user can be re-classified while still operating within the system. As it has already been said (in Section 3.1) there are two types of penalties $P = PE + PR$ that are assessed for sub-optimal behavior of the system. The PE term represents the penalty for a unit exit test due to a return to the unit. The return to a unit follows a failed test in a child unit. The PR term represents the penalty for finding out the correct remediation unit.

4 Current implementation issues

One of the requirements of implementation is that the system need not have any proprietary components and can run on the web under the TCP/IP protocol. The current implementations are relatively simple and are based on static html, html/php documents. The documents are retrieved either by a PHP control module or using PHP/Mysql control module. For the simplest implementations, with no permanent record of the user activities, we used javascript. A simple system that writes user information on a server uses PHP. We experiment with a simple system that can be modified without intensive software skills and the developer can experiment with a variety of user scoring and material sequencing procedures. All components have been available free of charge on both platforms (Win, Unix/Linux). In our experiments we have used a simple graphical user interface design at this stage of system development since our primary goal is to collect data. The user data analysis will be utilized as a tool for additional refinement of the model for in-classroom use.

The following is an example of a unit when the information is marked up using XML. The XML information can be parsed and then it can be searched for required content.

```
<?xml version="1.0"?>
<unit id="unique unit identification">
<children>
unit id’s that are children of this unit
</children>
<parents>
unit id’s that are parents of this unit
</parents>
<friends>
unit id’s that are related but neither children or parents
</friends>
<definition>
This is a placeholder for one or more definitions of the problem, variable
or concept.
</definition>
<example id = "unique example identification">
```

```
The example can be placed in one unit, or
it can be divided into two parts: a problem description and
a solution for the problem.
<problem>
This a placeholder for the problem description.
</problem>
<solution>
This is a placeholder for a solution
</solution>
</example>
<exitTest id = "unique exit test identification">
This is a placeholder for one of the tests
</exitTest>
</unit>
```

This xml unit is parsed and a set of html/php pages is generated. This set is then presented by a control module.

5 Conclusion

We have considered a class of flexible information cycling systems and associated user adaptive systems. In the latter class, the cost of the user path is lowered because of the user-orientation and adaptation of the system. The optimization is achieved with the help of probabilistic modelling of the user chance (and its development) to master the units, and the statistical evaluation of user data. The optimization is important for classroom applications where the time the user must spent practicing is minimized. We are currently in the process of setting up a procedure for extensive user data collection along with both on-line and off-line data processing. In the future, on demand information systems will have evolving natural language interface and applications will run with the semantic web framework.

Acknowledgement

P.Volf is thanked for his support with data analysis and programming.

References

1. Smid J., Obitko M., Volf P.: Model Parameters and Model Performance. *Proceedings of the UM2001 - Workshop on Empirical Evaluation of Adaptive Systems*, Sonthofen, GE, 2001, 25-32.
2. Smid J., Svacek P., Smid J. Jr.: Processing User Data for Intelligent Tutoring Models, *Workshop within KDD-2000: Post-Processing in Machine Learning and Data Mining: Interpretation, Visualization, Integration, and Related Topics*, Boston, MA, August 2000.

3. Volf P., Smid J.: Randomized Evaluation of States of an ITS, *Modeling and Simulation of Systems Conference MOSIS 01,* MARQ Ostrava, CZ, 2001, 227-232.
4. Weber G., Specht M.: User modeling and adaptive navigation support in WWW-based tutoring systems (1997),
http://www.psychologie.uni-trier.de/projects/ELM/elmart.html
5. Kobza A.: Generic User Modeling: User Modeling and User-Adapted Interaction, 11, 2001, 49-63.
6. Brusilovsky P.: Adaptive Hypermedia: Modeling and Use-Adapted Interaction, 11, 2001, 87-110.
7. Weber G.: Adaptive learning Systems in the World Wide Web, UM99 - User Modeling 1999,Proceedings of the Seventh International Conference, Judy Kay (Ed.), Springer Verlag, 1999. 371-377.
8. Weibelzahl S.: Evaluation of Adaptive Systems,UM01 - User Modeling 2001, Proceedings of the 8th International Conference, Springer Verlag, 2001.
9. Wahlster: Semantisches Web und Wissenmanagement.
http://www.dfki.de/ wahlster
10. Cobza: User Modeling, AI & Society 4, 1990, 214-240.

Using Genetic Algorithms for Data Mining in Web-based Educational Hypermedia Systems

Cristóbal Romero[1], Sebastián Ventura[1], Carlos de Castro[1], Wendy Hall[2], and Muan Hong Ng[2]

[1]Universidad de Córdoba, Campus Universitario de Rabanales, 14071, Córdoba, España
{cromero,sventura,cdecastro}@uco.es
[2]University of SouthSampton, Highfield, SO17 1BJ, SouthSampton, UK
{wh,mhn99r}@ecs.soton.ac.uk

Abstract. In this paper we show how to apply genetic algorithms for data mining of student information obtained in a Web-based Educational Adaptive Hypermedia System. The objective is to obtain interesting association rules so that the teacher can improve the performance of the system. In order to check the proposed algorithm we have used a Web-based Course developed for use by medical students. First, we will describe the proposed methodology, later the specific characteristics of the course and we will explain the information obtained about the students. We will continue on with the implemented genetic algorithm and finally with the rules discovered and the conclusions.

1 Introduction

Nowadays there is a growing trend of web-based technology applied for distance education. Particularly, Web-based Adaptive Educational Hypermedia Systems have many advantages because they can adapt the course for each specific student. But usually, the methodology used to elaborate them is static, that is, when the course elaboration is finished and published on the Internet it is never modified again. The teacher only accesses the student evaluation information obtained from the course to analyze the student's progress. We propose, a dynamic elaboration methodology, where the evaluation information is used to modify the course and to improve its performance for better student's learning. Our approach is to use a knowledge acquisition method (machine learning and data mining) to discover useful information that might help the teacher to improve the course. Nowadays data mining researches are beginning to use techniques such as Web Data Mining to evaluate web-learning activities [7]. We propose a genetic algorithm for data mining to evaluate the student information obtained from a Web-based Adaptive Hypermedia System. We have used a Web-based Hypermedia Course that was designed to be used by medical student as an example to evaluate our algorithm and to obtain association rules [4]. These rules could then be shown to the teacher in order to help him decide how the course could be modified to obtain best performance.

2 Methodology

The dynamic construction methodology of Web-based Hypermedia Courses that we propose is recurrent and evolutionary (Fig 1) and while the number of students who use the system increases, more information is available to the teacher to improve it.

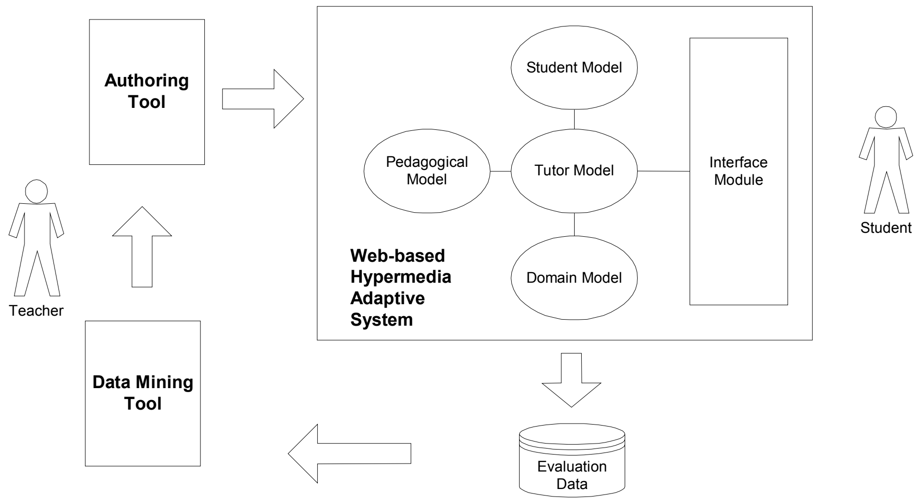


Fig. 1. Dynamic development methodology of Web-based Hypermedia Adaptive Systems.

In our methodology we can distinguish four main steps:

- Construction of the course. The teacher builds the Hypermedia Adaptive Course providing information of the domain model, the pedagogic model and the interface module. An authoring tool is usually used to facilitate this task. The remaining information, tutor model and the student model usually is given or acquired by the system itself. Once the teacher and the authoring tool finish the elaboration of the course, then, the full course's content may be published on a web server.
- Execution of the course. The students execute the course using a web navigator and in a transparent way the usage information is picked up and stored in the server in a huge database of all the students.
- Application of Data Mining. The teacher applies data mining algorithms [8] to the database to obtain important relationships among the data picked up. For this, he uses a graphical data mining tool.
- Improving the course. The teacher using the discovered relationships carries out the modifications that he believes more appropriate to improve the performance of the course. To do it, he again uses the authoring tool.

The process of execution, application and improvement can be repeated as many times as the teacher wants to do so. Although it is recommendable to have a significant amount of new students usage information before repeating it.

3 Web-based Hypermedia Medicine Course

We have used the data obtained from the evaluation of a Web-based Hypermedia Course in the study of Rheumatology. The system used to develop the course is an adaptive hypermedia system [6], but the evaluation data used for this paper was obtained from a usability study and there was no attempt to use the result to redesign the Course. As part of this evaluation the system was used by 30 users, of which 20 were medical and 10 were non-medical users. All the information was stored in a single database in the following tables: USER: String value that represents a system user, in our case they are 30. PERFORMANCE: Real value that represents the users performance in the 7 case studies in this application. AVEP_AH: Real value that represents the average performance of the users in the 7 case studies, adaptive application version. AVEP_NOAH: Real value that represents the average performance of the users in the 7 case studies, but in the version of the application without adaptation. CASETIME: Integer value that represents the time that a user takes in visualizing a complete case study. CASESCORE: Integer value that represents the score that an user has obtained when undertaking a case study. ACCESSTIME: Real value that represents the number of times a user has accessed the application. CONCEPT: Real value that represents the user's effort spent in the different concepts. QUESTIONSCORES: Integer value that represents the score obtained by the users in the relating questions to the case studies.

The data was preprocessed so that it will be easier to obtain relationship rules from them. This transformation consisted of a discretization, which mapped from continuous values (usually real values) to discrete values (strings that represent values groups) and integer values only needed to be labeled. In this way the modifications made to the tables are as follows: PERFORMANCE, AVEP_AH, AVEP_NOAH, CASETIME, ACCESSTIME and CONCEPT have been discretized to the labels VERYHIGH, HIGH, MEDIUM, LOW and VERYLOW. The values of CASESCORE and QUESTIONSSCORES have been named with the labels SUCCESSFIRST, SUCCESSSECOND, SUCCESSTHIRD and SUCCESSFOUR, which means getting the answer correct at the first attempt, second attempt and so forth. USER does not need modification.

4 Genetic Algorithm for Data Mining

Some of the main data mining tasks are [8]: classification, clustering, discovery of association rules, etc. We have used a genetic algorithm to obtain association rules from the user evaluation data. The association rules relate variable values. They are more general than classification rules due to the fact that in association rules any variable may be in the consequent or antecedent part of the rule. The classical problem of discovering association rules is defined as the acquisition of all the association rules between the variables [4]. Genetic algorithms are a paradigm based on the Darwin evolution process, where each individual codifies a solution and evolves to a better individual by means of genetic operators (mutation and crossover). In general the main motivation for using genetic algorithms for rule discovery is that they perform a global search and cope better with attribute interaction than greedy rule algorithms often used in data mining [3]. Most data mining methods are based on the rule induction paradigm, where the algorithm usually performs a kinds of local search (hill climbing). Also the fitness function in genetic algorithm evaluates the individual as a whole, i.e. all the interactions among attributes are take into account. In contrast, most rule induction methods select one attribute at a time and evaluate partially constructed candidates rule, rather than full candidate rule. The Genetic Process we have used consists of the following steps [5]: The first step is Initialization, next Evaluation, Selection and Reproduction steps are repeated until the Finalization condition is fulfilled.

4.1 Initialization

Initialization consists of generating a group of initial rules specified by the user (50 - 500 rules). Half of them are generated randomly and the other half starting from the most frequent values in the database. We use a Michigan approach in which each individual (chromosomes) encodes a single rule. The format of the rules we are going to discover is: IF Variable1 = Value1 (AND Variable2 = Value2 …) THEN VariableX =ValueX

Where:

- Variable1, Variable2, VariableX: Are the database's field names. (P0..P6,AVEPH,AVEPNOH,CASETIME0..CASETIME6,CASESCORE0.. CASESCORE6,ACCESSTIME0.. ACCESSTIME6,C0..C37,MCQ0.. MCQ6).
- Value1, Value2, ValueX: Are the possible values of the previous database fields (VERYLOW,LOW,MEDIUM,HIGH,VERYHIGH,SUCCESSFIRST, SUCCESSSECOND, SUCCESSTHIRD, SUCCESSFOUR).

We use value encoding in which a rule is a linear string of conditions, where each condition is a variable-value pair. The size of the rules is dynamic depend of the number of elements in antecedent and the last element always represents the consequent.

4.2 Evaluation

Evaluation consists of calculating the fitness of the current rules and keeping with the best ones. To calculate the fitness we count the precision of the rule, the number of patterns in the database that fulfill both antecedent and consequent and do not fulfill both antecedent and consequent. That is, we obtain very strong association rules [2] that fulfill [A=a] → [C=c] and [C≠c] → [A≠a]. So a rule is very strong if the previous two rules are strong, that is, both rules have greater support and confidence than a minimum values set by the user (0.5-1). Our formula detects both statistical negative dependence and independence between antecedent and consequent.

4.3 Selection

The selection chooses rules from the population to be parents to crossover or mutate. We use rank-based selection that first ranks the population and then every rule receives fitness from its ranking. The worst will have fitness 1, second worst 2, etc. and the best will have fitness N (number of rules in population). Parents are selected according to their fitness. With this method all the rules have a chance to be selected. We also use an elitism method, which first copies a few best rules to new population. Elitism increases performance of the genetic algorithm, as it prevents losing the best found solution.

4.4 Reproduction

Reproduction consists of creating new rules, mutating and crossing current rules (rules obtained in the previous evolution step). The crossover and mutation probability is set by the user. A higher crossover rate (50-95%) and a lower mutation rate (0.5-2%) are recommended. Additionally it is good to leave some part of population survive up to next generation. Mutation consists of the creation of a new rule, starting from an older rule where we change a variable or value. We randomly mutate a variable or values in the consequent or antecedent. Crossover consists of making two new rules, starting from the crossing of two existent rules. In crossing the antecedent of a rule is joined to the consequent of the other rule in order to form a new rule and vice versa (the consequent of the first rule is joined to an antecedent of the second). So it is necessary to have two rules to do the crossover.

4.5 Finalization

Finalization is the number of steps or generations that will be applied to the genetic process. The user chooses this value (10-500 steps). We could also have chosen to stop when a certain number of rules are obtained.

5 Rules discovered and Conclusions

We have carried out different execution proofs with the described genetic algorithm and the data obtained from the medical Web-based Hypermedia Course, to discover association rules. We have applied the algorithm to the whole data, only to the medical students and only to the other users. For each case, we have obtained different rules both in the content and in number and fit. For example, one of the rules obtained when using 100 initial rules and 100 steps is: IF CASESCORE2=SUCCESSFIRT AND CASESCORE4=SUCCESSFIRT THEN CASESCORE1=SUCCESSFIRST (with support = 0.73 and confidence=1). This can be interpreted as if a user gets the answer for case 2 and case 4 correct, he/she is likely to do well in case 1. The support of a rule gives the importance of a rule and the confidence of a rule gives its predictability power. All the rules discovered are showed to the teacher in order he can obtain conclusion about the course functionality. The teacher has to analyze them and he has to decide what are the best modifications that can improve the performance of the course. Summarizing the main conclusions that we obtained starting from the discovered rules are: We obtained expected relations, for example: between CASESCORE and CONCEPT, and between MCQ and CONCEPT, due to the fact that the questions are about the concepts. We obtained useful relations, for example: between CASESCORE and MCQ. This could be because the questions are the same ones, or they refer to the same concepts, or they have equal difficulty. We obtained strange relations, for example between AVEP_AH and P, it is probable that this relation takes place by chance. And we didn’t find any other relation, for example with ACCESSTIME or CASETIME. This could be because user access times were completely random and they did not determine any other variable as it might be expected.

Having tested our genetic algorithms on the Web-based Hpermedia Course as described above, and shown that they can produce potentially useful results, we plan to apply them to this and other adaptive hypermedia courses to test how they can be used to improve the adaptive features. The preliminary results reported in this paper are promising and they show that our genetic algorithm is a good alternative for extracting a small set of comprehensible rules, which is important in the context of data mining. We are now developing several courses with AHA system [1] and we expect to obtain interesting rules to improve the course adaptation. We have chosen AHA system because it is a generic model of hypermedia adaptive system and it has a high degree of adaptation. We are also developing a more sophisticated evolutionary algorithm with genetic programming [3], to obtain more complex and interesting rules.

References

1. De Bra P., Wu H., Aerst A., Houben G.: Adaptation Control in Adaptive Hypermedia Systems. International Conference on Adaptive Hypermedia. (2000).
2. Delgado M., Sánchez D., Martín-Bautista M.J., Vila A.: Mining association rules with improved semantics in medical databases. Artificial Intelligence in Medicine. 21. (2001).
3. Freitas, A. A.: A survey of evolutionary algorithms for data minig and knowledge discovery. To appear in: Advances in Evolutionary Computation. A Springer-Verlag, (2002).
4. Hipp J., Güntzer U., Nakhaeizaeh G.: Algorithms for Association Rule Mining – A General Survey and Comparison. SIGKDD Explorations. (2.000).
5. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
6. Ng. M, Hall W., Maier P., Armstrong R.: History-based Link Annotation for Self-Exploratory Learning in Web-based Hypermedia. 3rd workshop on Adaptive Hypertext and Hypermedia. Arhus, Denmark. (2001).
7. Osmar R. Zaïene.: Web Usage Mining for a Better Web-Based Learning Environment. Technical Report. (2001).
8. Witten I., Frank E.: Data Mining. Practical Machina Learning Tools an Techniques with Java implementations. Morgan Kaufmann Publishers (1999).

Simprac – Teaching system for management of chronic illness

Douglas Chesher[1], Judy Kay[2], and Nicholas King[1]

[1] School of Pathology
[2] School of Information Technologies
University of Sydney
Australia 2006

Abstract. Simprac is web-based medical case simulation for the diagnosis and management of chronic disease. Users have the opportunity to manage patients in the simulated environment over a series of consultations that vary based on the management plan selected at the end of each consultation. It has been designed with learner reflection as a central goal. This is supported by providing the learner with opportunities to review their progress at each consultation as well as at the end of a case compared to an expert or their own peer group.

1. Introduction

Simprac is simulation-based learning environment designed to run on the web. It tackles the problem of medical management of chronic illness. This is an important area of medical activity. Its very nature means that it involves a sequence of consultations. In the first such meeting, the medical practitioner needs to take a history and perform examinations. These provide a basis for a set of possible hypotheses for the underlying illness. The practitioner needs to request tests which can reduce this hypothesis space, especially taking care to exclude serious illness. At the same time, the practitioner needs to formulate a management plan which the patient should follow up to the time of the next consultation. At that time, the patient is reviewed, for several purposes, one of the important ones being to review the management regime and, if necessary, adjust it. With much chronic disease, there will be many consultations over a long period of time. This is typical in the case of such widespread and long-term illnesses as diabetes mellitus and cardiovascular disease.

A simulation-based teaching system is attractive in that it can give medical practitioners an opportunity to practice at their own convenience and in a short period of time. This can serve as an important complementary learning opportunity to be combined with the typical approaches that are widely used: rounds, educational meetings, conferences, refresher courses, programs, seminars, lectures, workshops, and symposia [1].

One of the worrying and striking observations made by Davis et. al. [1] is that traditional formal didactic continuing medical education (CME) had little influence on physician behaviour. At the same time, their work has provided inspiration for the approach taken in Simprac because they suggest that "interactive CME sessions that enhance participant activity and provide the opportunity to practice skills can effect change". In an important area like the management of chronic illness, we conclude that it is not sufficient to provide current information about best practice. This is true even for the case of adaptive hypertext that might be customised to the knowledge and learning preferences of the medical practitioners. It is important to offer an interactive experience.

2. Related Work

There is a considerable body of work on teaching medicine with the aid of simulations [2],[3]. Tellingly, it has also been used for assessment [4]. These are appearing as stand-alone simulations available on a CD-ROM as well as on the web. Actors have been used in research into medical problem solving by clinicians [5]. This offers a high fidelity simulation of a consultation. Our goal has been to identify the salient element of a simulation, capture these in our system and then enhance the whole experience by adding opportunities for learners to reflect and abstract.

Friedman [6] has provided an excellent outline of the features to be considered when developing a computer based clinical simulation. Combining this with the description by Melnick [7] of the system used by the American National Board of Medical Examiners (NBME), a minimum feature set can be developed. Briefly, we can summarise these as:

- Menu vs. Natural Language Requests for Data
- Interpreted vs. Uninterpreted Clinical Information
- Deterministic vs. Probabilistic Progression
- Natural Feedback vs. Instructional Intervention
- Scoring
- Single v multiple consultations

In these terms, Simprac offers limited natural language interaction with the practitioner being able to question the simulated patient in natural language. The clinical information is not interpreted. At this stage, the system is deterministic and it provides natural feedback in the form of the patient’s progress and state of health. Rather than employ a scoring system, Simprac has been designed to encourage learner self-assessment and reflection as discussed below. Finally, since Simprac has been designed to explore ways to support learning of chonic illness, it needs to simulate a series of consultations. Simprac’s design has been significantly influenced by the observations of Anderson, Reder and Simon [8] that learning is enhanced by combining abstract learning with concrete situations which reify the theory.

3. Overview of one example case

Simprac contains a simulation of a moderately difficult task for general practitioners: the diagnosis and management of hyperlipidaemia in a patient with significant co-mobidities that make management difficult. Figure 1 gives an overview of the case. As shown, it involves four consultations. This case models four possible states which the learner can meet at the second consultation. The full set of four consultations spans six months of simulated time.

The case is realistic and was modelled on anonymised data from a real patient. It is also of considerable practical interest. As the leftmost path in Figure 1 indicates, an incorrect management programme can lead to serious implications for the patient and the doctor.

Simprac simulates normal practice in that the first consultation is when the practitioner takes a patient history and performs initial examinations. From these, the practitioner formulates a set of potential hypotheses about the likely disease processes. These drive both the choice of additional tests and longer term investigations as well as the formulation of a plan for management.

In this case, there is very tight linkage between the patient states modelled. For example, if the user takes a sub-optimal management option at the initial consultation, they might come to the second consultation at the state modelled by the leftmost box of the second row of Figure 1. This reflects *no change from baseline*, meaning that the management programme did not effect any improvement and the serious illness of this patient continues to cause problems. If the learner recognises their error at this stage, they can select the correct management option and move to the optimal state by the third consultation, shown as the third row of the figure. This case exemplifies the tightly connected influence of the various management options with the ongoing disease outcomes. At the first and second consultations, either optimal or non-optimal management options lead to states which can subsequently move to any of the states at the next level. By the third consultation, this changes: from this stage, some of the states of the fourth row are inaccessible.

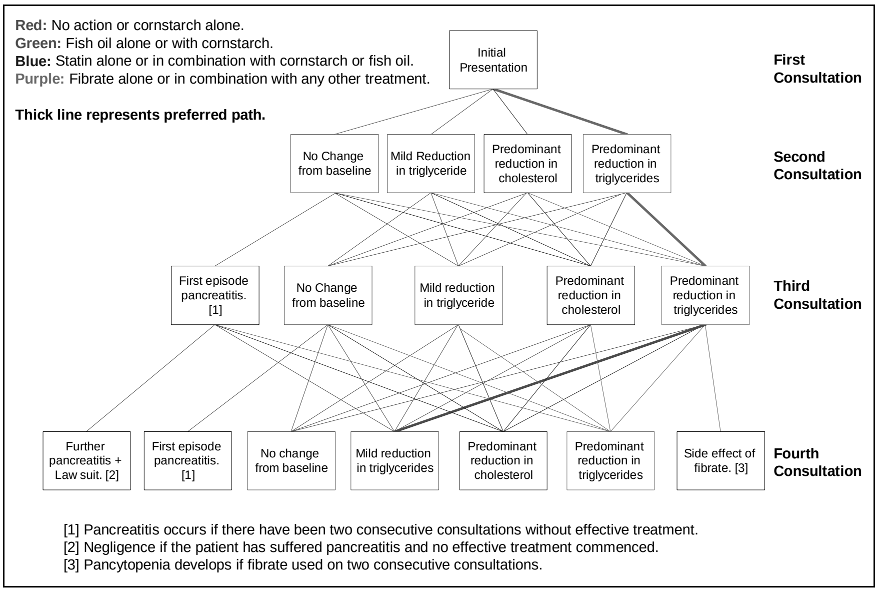


Fig. 1. Overview of four consultation sequence in simulation

4. Overview of interface

The first interface provides the learner with details of the case. From this, the user has several possible actions. The user would normally start by asking questions to collect the medical history.

This involves a limited natural language interface where the free text questions are matched against a coded set of questions. Since this is an inexact process, the user is offered the set of matching questions so that they can select the one which best captures their intent. This is clearly far less desirable than a more sophisticated natural language interface but it has the merit that it is less likely to lead the learner than a pure menu-based system.

The next stage of the first consultation will usually involve a physical examination via the interface illustrated in Figure 2. The practitioner can perform a virtual physical examination by selecting a variety of "tools" or actions and applying these to different parts of the body.

From the beginning of the consultation, the practitioner would make use of the *Medical Notes* interface to record their findings. This is selected at the left of the interface shown in Figure 2. Also, the learner can explicitly record the hypotheses they have formulated.

In addition to the interviews and examinations, the learner can request a wide variety of investigations. Of course, in line with reality, the results of the investigations are not available until a predetermined time later in the case. For example, results of a CAT scan will only be available at the next consultation.

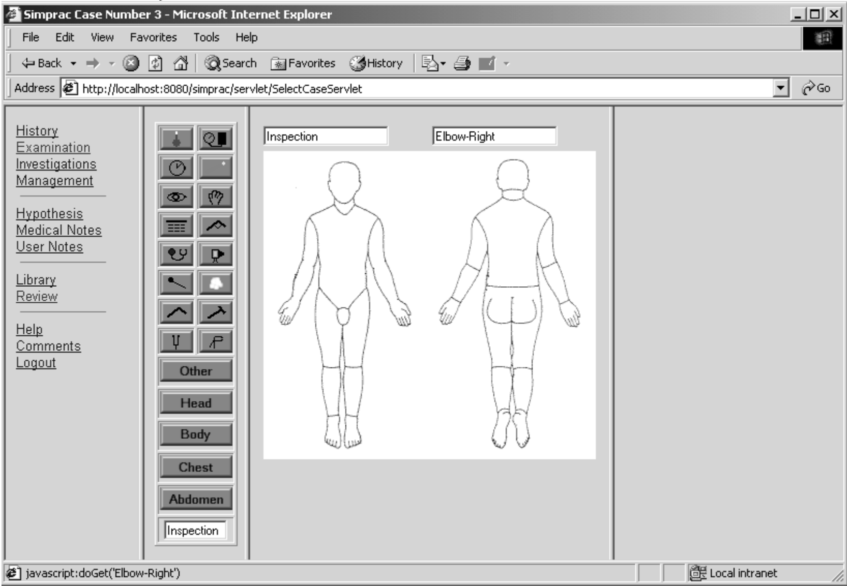


Fig. 2. Example of a screen for performing examinations. In this case the practitioner has chosen menu options to select the patient's right elbow. The system will report what they would be able to see. In some cases, this will provide an image of the area; in others, the user is provided with text stating what they would be able to observe.

Once the consultation is complete, Simprac offers the user the opportunity to review their progress. This takes the following form. The user can review their questions, examinations, investigations and management options they have chosen. They can simply review this as a basis for reflecting about what they have done in this consultation. The current implementation provides this level of support for reflection.

We have designed Simprac so that we will collect data as practitioners use the system. The goal is that the user be able to compare their own performance against their peers or an expert in the domain. We are currently developing the interfaces for this. The options to be supported are illustrated in Figure 3. As these indicate, we propose to enable the learner to compare their own performance with that of an expert. In addition, as we build up models of the diagnostic and management practices of peer groups, we want to allow the learner to compare their performance with their own peer group. This is important for our envisaged plans for the system. We will evaluate it with general practitioners as well as with medical students. Depending on the particular case, a general practitioner may want to judge their own performance in relation to other general practitioners rather than experts. Similarly, the medical student may want to be able to compare their own performance against that of other medical students as well as that of the expert. This range of group models is important for helping the learner make a realistic assessment of their own performance: it may not be reasonable for a learner to expect to perform at expert level on some cases.

A second level of review is available at completion of a full case. This provides a more holistic overview of performance over the full treatment path. At this level too, we envisage that reflection on performance should be based upon both study of the outcomes and of comparisons against expert and peer groups.

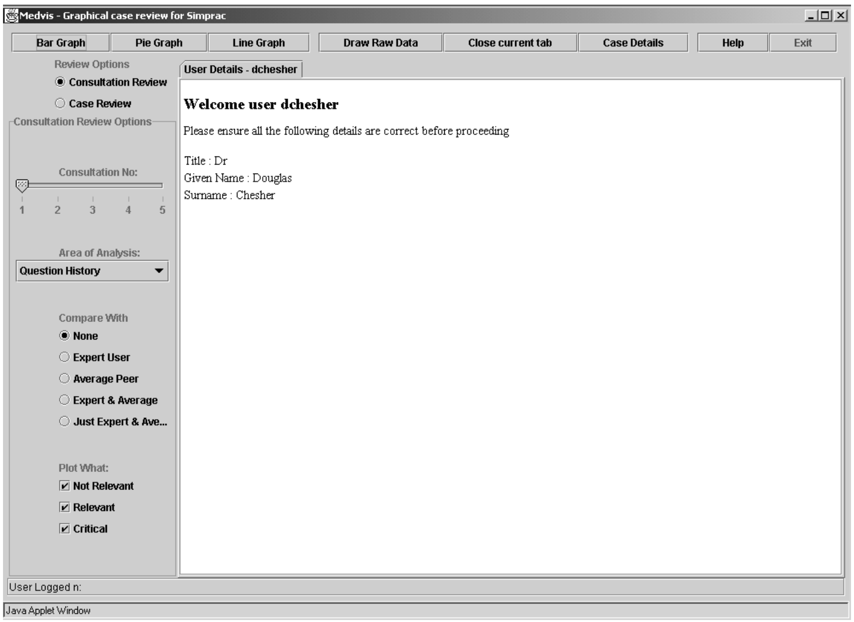


Fig. 3. Screen for selection of comparison options

5. Conclusions

Simprac is an exploration of several innovative aspects in a teaching system. It is web-based so that it can be used flexibly both by students and practicing medical practitioners. It is also significant in its exploration of teaching the management of chronic illness. Finally, it has been designed with learner reflection as a central goal. This is supported by providing the learner with opportunities to review their questions, examinations, investigations and management at each consultation as well as at the end of a case. It is also unusual in its approach to supporting reflection by enabling the learner to compare their own performance with that of an expert as well as against a peer group.

6. References

1. Davis D, O. B. M., Freemantle N, Wolf FM, Mazmanian P, Taylor-Vaisy A. "Impact of formal continuing medical education: Do conferences, workshops, rounds and other traditional continuing education activities change physician behaviour or health care outcomes?" Journal of the American Medical Association. **282** (1999) 867-874.
2. Bryce DA, K. N., Jeremy RW, Myers JH. Facilitating clinical reasoning through computer based learning and feedback. Australian society for computers in learning in tertiary education. (1997)
3. Hayes KA, L. C. "The interactive patient: a multimedia interactive educational tool on the world wide web." MD Computing **13** (1996) 330-334
4. Clyman SG, O. N. "Status report on the NBME's computer-based testing." Academic Medicine. **65** (1990) 235-240
5. Elstein ES, S. L., Sprafka SS. Medical Problem Solving: An Analysis of Clinical Reasoning. Cambridge, Harvard University Press. (1978)
6. Friedman CP. "Anatomy of the clinical simulation." Academic Medicine **70** (1995) 205-209.
7. Melnick DE. "Computer-based clinical simulation: State of the art." Evaluation and the Health Professions. **13** (1990) 104-120
8. Anderson JR, Reder. RL, Simon HA. Applications and misapplications of cognitive psychology to mathematics education. (1995) http://act.psy.cmu.edu/personal/ja/misapplied.html