Computing and Informatics, Vol. 27, 28, 639–659

INTRODUCING ADAPTIVITY AND COLLABORATIVE SUPPORT INTO A WEB-BASED LMS

Natasa Hoic-Bozic

Department of Information Science, University of Rijeka
Omladinska 14, HR-51000 Rijeka, Croatia
e-mail: natasa.hoic@ri.t-com.hr

Vedran Mornar, Ivica Boticki

Department of Applied Computing
Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, HR-10000 Zagreb, Croatia
e-mail: {vedran.mornar, ivica.boticki}@fer.hr

Manuscript received 28 November 2006; revised 4 June 2007
Communicated by Giovanni Guida

Abstract. In this paper the design and implementation of *AHyCo* (Adaptive Hypermedia Courseware), a web-based learning management system based on adaptive hypermedia, is described. *AHyCo* consists of a domain model, a student model, an adaptive model and a collaborative model. *AHyCo* supports interaction between students and content by using adaptive hypermedia and online tests. Particular attention is given to the design of the collaborative functionality which enables automatic grouping of students based on various criteria. Furthermore, student to student and student to teacher interaction is supported through asynchronous communication (forum). File sharing and inter-group grading and evaluation modules were introduced into the collaborative module as well enticing interaction between students across groups.

Keywords: WWW, e-learning, learning management system (LMS), adaptive hypermedia, computer mediated communication, group forming

1 INTRODUCTION

E-learning is usually defined as a learning environment that relies on the Internet/WWW as the primary delivery mode of communication and presentation. Students acquire knowledge via the Internet, without the need to be physically present within the learning environment. In this definition e-learning has the form of the computer-centered learning with students sitting in front of a computer and reading prepared courseware. Today one other aspect of e-learning becomes important: computer as a medium for communication, research and information gathering as well as a learning tool [3, 11].

In the modern e-learning, learning management systems (LMSs), created to integrate tools needed for all activities performed by teachers and students, are becoming more and more used [21, 7]. These systems use various ways to store content for learning, information needed for learning and information regarding students. Main characteristic of an LMS are: it offers authoring tools for courseware creation, enables learning of prepared courseware and navigation through it, manages learning, provides a computer based interface and tests knowledge.

In the last few years research in the field of e-learning has intensified, especially in the area of adaptive hypermedia [5, 6]. This trend will be visible in the future years as well, and will be materialized in the form of web-based adaptive LMSs. These LMSs will, in addition to their standard features, enable the use of adaptive content and navigation in the learning process. This new generation of LMSs will not focus only on learning content creation, delivery and assessment, but will try to include constructivistic and collaborative learning and teaching methods. This new approach guarantees the rise of students' motivation for learning and leads to the better results. In addition to the domain, student and adaptation model, the collaborative model is expected to be affirmed as a standard part of adaptive systems.

In this paper we have presented *AHyCo* (Adaptive Hypermedia Courseware) adaptive LMS for development and distribution of adaptive web-based courseware, with special focus on the implementation of collaborative learning support. Our approach to the collaborative learning and asynchronous communication using *AHyCo* subsystem for organizing learning groups and forum has also been described.

The remainder of the paper is organized as follows:We briefly review related work in the field of LMSs and adaptive hypermedia in Section 2. Section 3 formally defines the domain model, student model, adaptation model and collaborative model of the *AHyCo* system. *AHyCo*'s learning and testing environment is briefly presented in Section 4. Section 5 focuses on the *AHyCo* s environment for collaborative learning. Section 6 describes the use and evaluation of the *AHyCo* system, followed by Section 7 which presents conclusions and future plans.

2 BACKGROUND AND RELATED WORK

The main disadvantage of today's classical LMSs is that they offer equal ("one size fits all") services for all students and do not adapt to their affinities, interests, communication skills and knowledge. All learners taking an LMS-based course, regardless of their knowledge, goals, and interests, receive access to the same educational material and the same set of tools, buffered with no personalized support. Even the best of today's LMSs (*WebCT, Blackboard* [4, 22]) are able to adapt to their users in a very limited manner. Some provide students with access to the courseware through their personal WWW pages or offer rudimental collection of hyperlinks. In these cases structured domain model with linked courseware concepts is missing [10].

"One size fits all" problem is present in other web-based systems as well. The effort to deal with it has enticed the development of a new concept in hypermedia: adaptive hypermedia. According to [5], the term adaptive hypermedia systems denotes all hypertext and hypermedia systems reflecting some features of the user in the user model which apply this model to adapt various visible aspects of the system to the user. An adaptive hypermedia system (AHS) adapts the presentation of content or links based on the user model. Two major technologies in adaptive hypermedia can be distinguished: adaptive presentation and adaptive navigation support. Adaptive presentation adapts either the content of a document or the style of the text. Adaptive navigation support concentrates on changing the presentation of links.

The most popular area for adaptive hypermedia research is the educational hypermedia, in which the student's goal is to study the courseware on a particular subject [5]. The most important element in educational hypermedia is the user knowledge of the subject that is being taught. Since 1993 the Web has become the primary platform for developing adaptive hypermedia educational systems [6]. Adaptive web-based collaborative learning environments became the research topic in the field of adaptive hypermedia as well and are introduced in [26, 13]. Some examples of such adaptive Web-based educational systems are *InterBook* [12], *AHM* [9], SIETTE [25], *AHA!* [10], *NetCoach* [29], *ALE* [27], COLER [8].

The problem of the current generation of adaptive Web-based educational systems is their architecture [7] since most of them do not support typical functions of an LMS. Therefore only a few of them are actually being used for teaching real courses. In short, while providing a rich set of tools, many of today's successful LMSs do not offer enough personalization and adaptation. On the other hand, adaptive hypermedia educational systems have the techniques for adaptation but show the lack of availability of standard tools.

Considering all mentioned disadvantages of classical LMSs on the one hand and adaptive Web-based educational systems on the other, we have started the development of our own system called *AHyCo* (Adaptive Hypermedia Courseware) [1] which offers both characteristics of a classical LMSs and the technology used by the adaptive systems. In addition to the classical features the development of a completely new system was primarily guided by the whish for innovation. The goal was to build a complete LMS offering learning environment with adaptive navigation, testing, course management and computer-mediated communication, all backed up with the corresponding authoring tools.

Similar example is the distributed architecture for adaptive e-learning *KnowledgeTree* [7] but is more oriented to the implementation of a learning portal rather than an LMS. The theoretical model of *AHyCo*, with some extensions and adjustments, is similar to the models of other *AHS* (e.g. *AHAM* [30]). It consists of three main components: the domain model, the student model and the adaptive model. System's functionality way recently extended with the collaborative model. Like *NetCoach*, [29] *AHyCo* is designed to enable the development of adaptive learning courses without any knowledge of programming. *AHyCo* is delivered as Open Source software like *AHA!* [10] and has been developed through several research projects. The system is currently used for teaching several courses and is being further developed.

3 *AHYCO* SYSTEM OVERVIEW

The model of *AHyCo* originally consists of four submodels: the domain model, the student model, the adaptive model and the collaborative model. The domain model describes structure of the learning domain as a studentet of reusable concepts (lessons and tests), linked together with prerequisite relationships. The student model encompasses the student's knowledge of lessons. The adaptive model contains rules for adaptation. The rules define how the domain model and the student model are combined together to provide adaptive navigation support. The collaborative model contains information on groups of students and rules (algorithms) that are used to divide students into groups. The system is composed of two environments: the authoring environment [18] and the learning environment [15, 17].

3.1 The Domain Model

AHyCo's domain model has a two-level structure and consists of concepts. A concept is an elementary piece of knowledge for the given learning domain (Figure 1).

The first domain level is a graph $(\mathcal{C}_k, \mathcal{LC}_k)$, where $\mathcal{C}_k$ is the set of concepts and $\mathcal{LC}_k$ is the set of arcs, $\mathcal{LC}_k \subseteq \mathcal{C}_k$ x $\mathcal{C}_k$The links represent the prerequisite relationships $\prec$. These relationships denote the pedagogical constraints. $K_i \prec K_j$, for example, means that "concept K_i should be learned before concept K_j ". In contrast to e.g. the *AHAM* model, this is the only pedagogical strategy used in our model, because we intended to make the linkage of concepts as simple as possible.

To split the domain into more manageable units, concepts are grouped into modules M_k. The second level of the domain model is a directed graph $\mathcal{D} = (\mathcal{M}, \mathcal{LM})$, where $\mathcal{M}$ is the set of modules and $\mathcal{LM}$ is the set of arcs, $\mathcal{LM} \subseteq \mathcal{M}$ x $\mathcal{M}$. The arc connecting modules M_k and M_l exists if $M_k \prec M_l$. The prerequisite relationship $\prec$ for a module pair means that a certain minimal acceptable knowledge level for

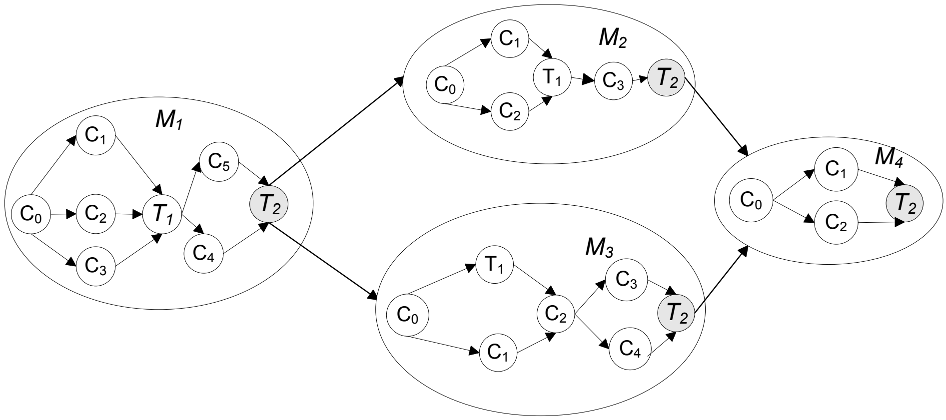


Fig. 1. An example of a domain model

module M_k should be reached before the student can start to learn lessons from M_l. The entire directed graph $\mathcal{D}$ (the domain) is equivalent to one course student has enrolled in.

The majority of concepts in the graph are lessons C_i. Some of the concepts in the graph are tests T_j that contain questions about the lessons. The main purpose of the tests T_k from the domain model is to govern the navigation within the module (mini-tests or quizzes T_k) and the navigation between modules (the final test T_f) by updating the student model after calculating students' knowledge levels.

The other type of a test is also introduced in the new *AHyCo*'s version and is used for standard students' knowledge evaluation.

A concept-lesson C_i is defined as ($\mathcal{FC}_i$, $\mathcal{PC}_i$, $\mathcal{QC}_i$, R_i, wc_i, lMy_i) where:

- $\mathcal{FC}_i$ is a set of multimedia fragments (small building blocks, e.g. a piece of text, graphics, sound, video clip...).
- $\mathcal{PC}_i$ is a set of prerequisite lessons, which are essential for a student to understand the lesson C_i.
- $\mathcal{QC}_i$ is a set of questions related to the lesson C_i. The questions may be with multiple-choice/single answer.
- R_i is the rank of the lesson C_i calculated as follows:
- $R_0 = 0$ (for lessons with no prerequisites, $PC_i = \emptyset$)
- $R_k = \max_{j|C_j \in \mathcal{PC}_k} R_j + 1$,
- wc_i is the weight of the lesson C_i with respect to the containing module M_k, determined by the author. $wc_i \in (0, 1)$ and $\sum wc_i = 1$
- lMy_i is a predefined minimal acceptable knowledge level of the lesson C_i, calculated by MYCIN formula [2, 23], $lMy_i \in (-1, 1)$.

A concept-test T_j is defined as a $(\mathcal{PT}_j, n_j, \mathcal{N}_j, R_j, u)$ where:

- $\mathcal{PT}_j$ is a set of prerequisite concepts-lessons, $\mathcal{PT}_j = \{C_i \,|C_i \prec T_j\}$. T_j will contain questions related to lessons from $\mathcal{PT}_j$.
- n_j is the total number of questions in T_j.
- $\mathcal{N}_j$ is the set of configuration rules that specify how many questions for each lesson C_i should be placed into the test T_j.
- R_j is the rank of the test T_j and is calculated analogously to C_i.
- u denotes whether the test T_j updates student model ($u = \text{true}$) or is used just for classical students' knowledge evaluation ($u = \text{false}$), $u \in \{\text{false}, \text{true}\}$, default is true.

Each question from $\mathcal{QC}_i$ is defined as $(\mathcal{FQ}, q, \mathcal{A}, \mathcal{H}, \mathcal{B}, p)$ where:

- $\mathcal{FQ}$ is the set of hypermedia fragments that form the question (stem). The stem of the parameterized questions contains parameters.
- q is the confidence level of the fact that student has either learned the lesson if he/she answers the question correctly (in MYCIN formula $f = q$), or has not learned the lesson if he/she answers incorrectly (in MYCIN formula $f = -q$), $q \in (0, 1)$. It is predefined by the author. Easier questions have smaller q.
- $\mathcal{A}$ is the set of the offered answers. Each offered answer $\mathcal{A}_j$ is a set of hypermedia fragments. Answer $\mathcal{A}_j$ of the parameterized questions is a function $f_j(p_1, p_2, \ldots, p_n)$ that evaluates a candidate answer on the basis of parameters p_1, p_2, …, p_n. Function f_j is defined in a scripting language and is evaluated after the parameters have been randomly generated.
- $\mathcal{H}$ is the set of indices of correct answers or functions.
- $\mathcal{B}$ is the set of lower and upper bounds for parameters p_1, p_2, …, p_n.
- p with $u = \text{false}$, p is used only with the classical students' knowledge evaluation or when dividing students into groups.

A module M_k is defined as a $(\mathcal{C}_k, \mathcal{PM}_k, lm_k, wm_k, R_k)$ where:

- $\mathcal{C}_k$ is a set of concepts that exist in the module.
- $\mathcal{PM}_k$ is a set of prerequisite modules for module M_k.
- lm_k is the minimum acceptable knowledge for module, $lm_k \in (-1, 1)$.
- wm_k is the weight of the module M_k with respect to the containing course D, determined by the author. $wm_k \in (0, 1)$ and $\sum wm_k = 1$.
- R_k is the rank of the module M_k.

Prerequisite relationship can denote different things when used in learning materials from different areas. In areas like math, physics and science generally, prerequisite relationship $C_i \prec C_j$ denotes inability to start the learning of the concept C_j until knowledge level for the concept C_i has not been be raised to a certain level. In contrast, areas like art or literature use prerequisite relationship simply to denote which concept comes before other ones. Author can set prerequisite relationships according to his or her preferences and change them when needed.

3.2 Student Model

Originally, the two-level student model, a variant of the overlay model for representation of the student's knowledge [5], has been proposed for the *AHyCo* system. For the purpose of creating new *AHyCo*'s version that supports collaborative work, the student model had to be extended in order to include some additional information on students' knowledge.

The set of students $\mathcal{S}$ is introduced with:

- $\mathcal{S} = \{S_j\}$; $j = 1, \dots, NS$
- NS is the total number of students enrolled in a course $\mathcal{D}$.

The first level of the student model represents the estimate of students' knowledge about the lessons C_i, denoted by r_i and k_i.

r_i is the estimate of the fact that the student S_j has read the lesson C_i or not. Initially, $r_i = 0$ for every C_i. To set r_i to 1, the student must not only view a page containing any lesson, but chose one of the concepts proposed by the system. All hyperlinks for the concepts are shown in the bottom of the page along with the current lesson. Although not a perfect estimate, this at least proves that the student has scrolled to the bottom of the page. An alternative, such as a timed approach, even if the volume of the page is taken into account, is at least equally inappropriate. As an example, the page can stay opened even if the student is not actually reading it. Furthermore, such an approach forces the student to waste valuable time by keeping opened an argument he/she can already know well.

k_i is the estimate about the student's knowledge of the lesson C_i. It is calculated by a variant of the expert systems' model MYCIN [2, 23]. A possible approach to solve an uncertainty problem such as the estimation of the student's knowledge is the certainty factor utilization, implemented in MYCIN. This approach has been chosen for *AHyCo*, because it is simple to implement and understand. The fact that a student has correctly answered a question contributes to the hypothesis "the student knows the concept C_i." The opposite fact contributes to the negation of the same hypothesis.

The knowledge level of a lesson is set by testing and can range from -1 (student does not know the lesson) to 1 (student knows the lesson). Before the student attempts any of the tests, all lessons in the student model have an initial level of $k_i = 0$.

After answering a question related to the lesson C_i, the new knowledge level k_i' for the lesson C_i is calculated according to:

$$k_i' = \begin{cases} k_i + (1 - k_i)f, & k_i > 0, f > 0 \\ k_i + (1 + k_i)f, & k_i < 0, f < 0 \\ (k_i + f)/(1 - min(|k_i|, |f|)), & \text{otherwise.} \end{cases}$$

The new level k_i' is based on the previous knowledge level k_i and the factor q, the confidence level of the fact that the student knows or does not know the lesson if he/she knows that question. If the student answers the question correctly, $f = q$, otherwise $f = -q$.

The model asymptotically increases/decreases the knowledge level for a concept with each correct/incorrect answer, according to the previous knowledge level k_i and the question weight q from the domain model.

The second level represents the knowledge about the modules. The knowledge level km_k of the module M_k is calculated according to the formula:

$$km_k = \sum k_j \times wc_j$$

for each C_j from the module M_k, where wc_j is the weight of the lesson C_j [15]. According to the formula, the more important lessons (with higher weight wc_j) have more influence on the knowledge level km_k.

Besides student's knowledge level k_i about the lesson C_i and km_k about the module M_k, for the purpose of new *AHyCo*'s collaborative features, new summary knowledge level kd_l for a domain or course $\mathcal{D}$ has been introduced and is calculated with the following formula:

$$kd = \sum km_k \times wm_k$$

for each module M_k of the course $\mathcal{D}$.

In this version the student model was upgraded in a way that for every student who completed test T_j (which has its variable u set to false) there is value tr_j, calculated by summing up points p for every correctly answered question.

For a given student S_j the third, final value kfd_j, which will be used for group forming algorithms, can be calculated by combining together values kd_j and tr_j:

$$kfd_j = c_1 \times kd_j + c_2 \times tr_j$$

$$c_1 = c_1'/(c_1' + c_2')$$

$$c_2 = c_2'/(c_1' + c_2')$$

$$c_1', c_2' \in R.$$

kfd_j is a calculation result denoting overall success adjusted by constants c_1 and c_2. Initial constants c_1' and c_2', defined by the teacher to denote relative importance kd_j and tr_j, respectively are normalized in order to enable flexibility in the teacher's initial choice of constants. Therefore, a teacher can fine tune parameters in order to spawn optimal results.

Apart from data for presenting knowledge, the student model contains static data [10]. Static data contains personal and administrative student data (name, surname, identification code, password, email address, study group etc.); and also data for keeping record of students' presence, questionnaires and grades (i.e. when has learning of a certain concept started and finished, how many tests have been completed and how often etc.).

To assist collaborative work with *AHyCo*, as is the case with other collaborative systems [13], the data on user interaction with the platform (number of messages sent in forum, the number of threads that the student have initiated, the number of files uploaded in file storage area) is preserved.

3.3 Adaptation Model

The adaptation model consists of adaptation rules that define how the domain model and the student model are combined together to perform adaptive navigation support.

In our system, we used the adaptive navigation, which is a combination of free and guided navigation. The student can freely follow any hyperlink within a module or graph ($\mathcal{C}_k$, $\mathcal{LC}_k$), but the list of hyperlinks suits him/her best according to the navigation plan generated for him/her. *AHyCo* uses the combination of link sorting and link annotation adaptive techniques. The navigation within a graph ($\mathcal{M}, \mathcal{LM}$) is restricted and depends on the student's knowledge value km_k.

According to the student model and currently displayed lesson C_a, concepts from the module M_k are classified into several subsets: learned concepts $\mathcal{CL}_a$, recommended concepts where all prerequisite concepts have been visited $\mathcal{CC}_a$, and not recommended concepts $\mathcal{CN}_a$. There are also completely recommended concepts $\mathcal{CP}_a$ or recommended concepts that are in direct prerequisite relationship with C_a according to the directed graph ($\mathcal{C}_k, \mathcal{LC}_k$).

All sets are sorted according to the rank R_i of the concepts C_i belonging to the graph ($\mathcal{C}_k, \mathcal{LC}_k$).

The concepts C_i from the $\mathcal{C}_k$ of the module M_k are classified according to the algorithm 1 [17]:

Input: graph ($\mathcal{C}_k$, $\mathcal{LC}_k$), active lesson $C_a \in \mathcal{C}_k$
Output: sets $\mathcal{CL}_a$, $\mathcal{CP}_a$, $\mathcal{CC}_a$, $\mathcal{CN}_a$
$\mathcal{CL}_a = \emptyset$, $\mathcal{CP}_a = \emptyset$, $\mathcal{CC}_a = \emptyset$, $\mathcal{CN}_a = \emptyset$
for each $C_i \in \mathcal{C}_k \setminus \{ C_a \}$
  if $r_i = true$
    /* C_i is visited */
    $\mathcal{CL}_a = \mathcal{CL}_a \cup \{ C_i \}$
  elseif $C_a \prec C_i$
    if $r_j = true, \forall\ C_j \in \mathcal{PC}_i \setminus \{ C_a \}$
      /* all prerequisites for Ci are visited except C_a -
      C_i is completely recommended */
      $\mathcal{CP}_a = \mathcal{CP}_a \cup \{ C_i \}$
    else
    /* C_i is not recommended */
    $\mathcal{CN}_a = \mathcal{CN}_a \cup \{ C_i \}$
  elseif $r_j = true, \forall\ C_j \in PC_i$
    /* all prerequisites for C_i are visited –

C_i is recommended */
$\mathcal{CC}_a = \mathcal{CC}_a \cup \{ C_i\}$
else
/* C_i is not recommended */
$\mathcal{CN}_a = \mathcal{CN}_a \cup \{ C_i\}$

Algorithm 1. Classifying the concepts $C_i \in \mathcal{C}_k$

Navigation within the module M_k goes on before the student completes the final test T_f. This navigation is actually the traversing of a directed graph $(\mathcal{C}_k, \mathcal{LC}_k)$, following the hyperlinks suggested by the system in the bottom of the page. The model uses mini-tests or quizzes T_j to check students' knowledge and to update the student model while he/she navigates within the module. Transition to another module is possible after the successful completion of the final test T_f. There are three possible outcomes of the test T_f that belongs to the module M_k:

- T_f is completely passed – the M_k is learned: knowledge value $km_k > lm_k$ and $k_i > lMy_i$, $\forall C_i \in \mathcal{C}_k$. The student can proceed to another module according to the directed graph $(\mathcal{M}, \mathcal{LM})$;
- T_f is partially passed – the concepts C_i with $k_i \leq lMy_i$ are offered for repetition but the student may proceed to another module since $km_k > lm_k$; M_k is partially learned;
- T_f is not passed – $km_k \leq lm_k$ the concepts C_i with $k_i \leq lMy_i$ from M_k are offered for repetition and the student should retake the test T_f in order to proceed to another module.

The student model is updated after the test T_j has been completed according to the algorithm 2 [17]:

Input: the student model, lm_k and lMy_i from the domain model,
results of the test T_j with u=true, $C_i \in \mathcal{PT}_j$
Output: updated values of k_i, rp_i, km_i
use MYCIN for calculating k_i
if $\exists\ C_i \in \mathcal{PT}_j$, $k_i \leq lMy_i$
set $rp_i = 1$
M_k is partially learned
if $T_j = T_f$
/* T_j is final test for M_k */
calculate value km_k of M_k
if $km_k > lm_k$
M_k is learned
else M_k is partially learned

Algorithm 2. Updating the student model after T_j

This algorithm degenerates into a simpler one if variable u of test T_j is set to false. In that case test results tr_j are stored in the database for future use (e.g. as an input parameter when dividing students into groups).

3.4 Collaborative Model

Collaborative model upgrades *AHyCo* in order to support group work, especially group formation based on individual student models.

AHyCo forms groups in order to enable collaborative work on a predefined project set on a course $\mathcal{D}$ in which students to be grouped participate. A project is a learning task in which students have a choice of topics and directions and whose outcome is therefore unpredictable. It requires their initiative, creativity and organizing skills: they are required to produce a report, plan or suggest a design that comprises the solution to a problem [19].

The collaborative model consists of the data about the groups of students and the rules (algorithms) that are used to divide students into groups according to the data from the student model and domain model.

For the purpose of group forming, the set of groups $\mathcal{G}$ is introduced:

$$\mathcal{G} = \{G_1, G_2, \ldots, G_i, \ldots G_{NG}; 0 < i \leq NG \mid G_i \subseteq S, G_i \cap G_j = \emptyset\}.$$

NG is the number of groups G_i into which students are divided.

Since an important part of collaboration in *AHyCo* is asynchronous communication between students that use forum, the notation of the set of themes available in forum $\mathcal{TH}$ is introduced:

- $\mathcal{TH} = \{TH_1, TH_2, \ldots, TH_k, \ldots, TH_{NT}; 0 < k \leq NT\}$
- NT is the total number of the themes in forum. The themes are created by a moderator or a teacher.

Every student S_j, depending on which groups he or she is enrolled in, has access to the subset of themes from $\mathcal{TH}$. Therefore, there is a relationship between the set of groups $\mathcal{G}$ and $\mathcal{TH}$ because every theme TH_k can be assigned to a group G_j so that its members can participate in private discussions on that theme.

Each theme consists of a series of threads with a start message (prepared by a teacher or a student) and responses to it.

3.4.1 Algorithms Used in Group Creation

When preparing group work a teacher has to consider groups' size and criteria by which they will be formed. The number of students per group strongly influences interaction: the smaller the group the greater is the probability for mutual trust and good relationships inside the group [14]. On the other hand, disadvantages of smaller groups are lack of diversity, knowledge, skills and competence [28].

When grouping students, educational, emotional and social aims have to be considered as well. Although heterogeneous group of students give the best chance for interaction, in some cases intellectual differences have to be looked into as well [19]. Therefore, it is a good practice to group students to mix the more efficient ones with the less successful ones in each group. This could be an excellent opportunity for both sides: better students could learn more successfully through helping their colleagues. Nevertheless, intellectual qualities are not the only factor to be considered: social and emotional characteristics are to be included as well.

At this point *AHyCo* considers cognitive aspects only as criteria for group formation: groups are formed based on the estimation of students' knowledge and, partially, on their affinity to communicate in the forum.

Generally, the knowledge value kfd_l for every student can be used as a measure of "how good" a student is in comparison to his or her colleagues and therefore can be a criterion by which students will be grouped together.

Algorithms used to divide students into groups are denoted with $\mathcal{ALG}$:

- $\mathcal{ALG} = \{ALG_1, ALG_2, ALG_3, \ldots, ALG_i, \ldots, ALG_{NAlg} \mid NAlg$ is the overall number of algorithms used in the system; $0 < i \leq NAlg\}$
- ALG_i is an instance of an algorithm used to create groups (currently, $NAlg = 3$). For actual group forming process more algorithms can be used. Every instance of an algorithm ALG_i has this input data:

- NG – the number of groups to be created
- NSG – the number of students per a newly created group (can be used instead of NG; mutually exclusive)
- $\mathcal{D}$ – domain from which input data will be taken
- $\mathcal{S}$ – students that are to be grouped.

The following algorithms are used:

- *Best lead algorithm* – This algorithm creates groups of students in which the best of them lead. The "best" students are the ones with the highest final value kfd_l, calculated from the student model. Since a group's overall knowledge level is calculated by summing up its members' knowledge levels, each group is populated in the way that minimizes the difference between groups' overall knowledge levels.
- *Homogenous group algorithm* – This algorithm groups students so that the best of them remain together. Therefore, each group is populated in the way that maximizes the difference between groups' overall knowledge levels.
- *Interaction activity algorithm* – This algorithm groups students depending on their interaction activity by taking into account the number of messages posted in *AHyCo*'s forum. For $\mathcal{TH}$ as an input parameter, the algorithm extracts the number of posted messages for every theme TH_i in as well as the number of readings per message, combines the extracted data up and calculates the

final results in order to determine the student's overall activity for themes in $\mathcal{TH}$. The algorithm is to be upgraded with the possibility of grading individual messages/threads to give more controls to teachers.

As an example, only the best lead algorithm is presented:

Input: *NG*, *NSG*, $Kfd = \{ kfd_i \mid 1 < i <= NS \}$, $\mathcal{S}$, $\mathcal{D}$
Output: $\mathcal{G}$
$\mathcal{G}$ = createGroups(*NG*, $\#\mathcal{S}$, *NSG*,)
$\mathcal{S}'$ = sortStudents($\mathcal{S}$, *Kfd*)
/* creates a sorted set $\mathcal{S}$ with students sorted by kfd_i */
    for each $S_i \in \mathcal{S}'$
    j = i mod *NG* /* i is the rank of a student in a sorted group */
$\mathcal{G}_j = \mathcal{G}_j \cup \{ S_i \}$

Algorithm 3. Best lead algorithm

After an instance of algorithm ALG_i has finished its work, the system is supplied with newly created set of groups $\mathcal{G}$. These groups have to be attached to the domain $\mathcal{D}$ and an academic year in which they will be used (usually current academic year). It is therefore obvious that the group creation process is somewhat circular and utilizes the data from previous student efforts. When such data is missing the teacher is supposed to provide the custom input to the algorithms.

4 LEARNING AND TESTING

To use *AHyCo*'s learning environment [1], a student has to log in first. After authorization, the student has to choose the subject for learning. For the selected subject $\mathcal{D}$, a web page containing the lesson C_i is generated. This lesson is chosen in accordance to the adaptation rules and the data stored in the student model corresponding to the students' previous knowledge. The upper part of the page (Figure 2) is static and represents the content of a lesson.

At the bottom of the page hyperlinks to the continuing lessons or tests T_j are proposed by the system. The suggested hyperlinks are automatically generated before the page is shown and are annotated with various colors corresponding to concept types.

The concepts are listed in the following order:

1. Completely recommended or main concepts – green color annotates the concepts where all prerequisite concepts have been visited and these concepts are the best continuation for C_i according to the directed graph $(\mathcal{C}_k, \mathcal{LC}_k)$.
2. Recommended concepts – orange color annotates all other concepts where all prerequisite concepts have been read.

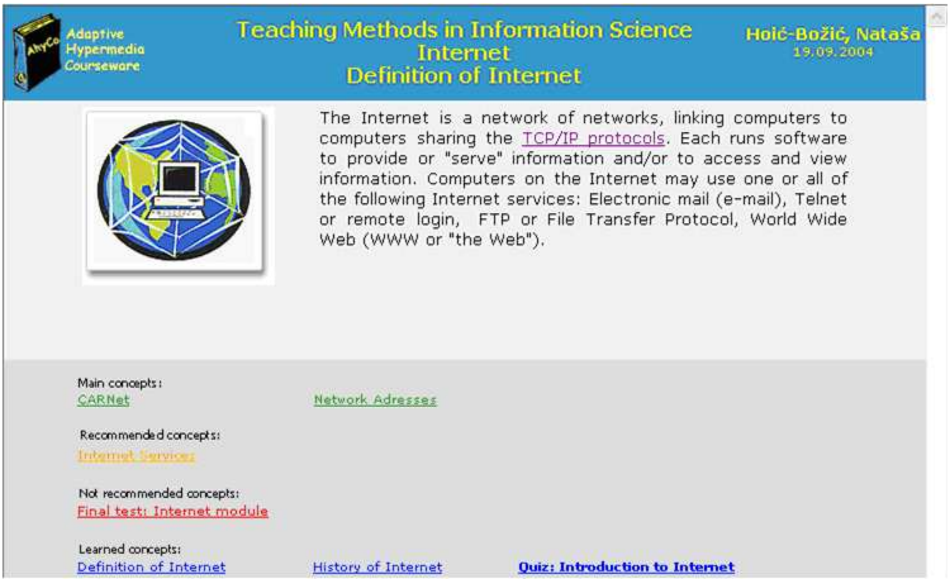


Fig. 2. The page with the content of a lesson

3. Not recommended concepts – red color annotates concepts where some of the prerequisite concepts have not been read or the knowledge level of some prerequisite $k_i \leq lMy_i$.
4. Visited concepts – blue color annotates the lessons with $r_i = true$ or $k_i > lMy_i$.

All the hyperlinks within the hypertext network are functional so the student can follow any one of them. The idea of free navigation is only to support and aid students. It is up to a student whether he/she will follow the system's suggestions. After the student has finished the learning of the module's lessons, he/she should choose the final test button. The test questions and the sequence of the offered answers are generated randomly. Transition to another module is possible after successful completion of the test T_f. Results of the test by individual lessons are displayed separately. The test of n questions consists of n WWW pages. Each WWW page corresponds to a single question. The test questions and the sequence of offered answers are generated randomly. The student selects the answer and navigates through the test pages by using the common web interface elements (hyperlinks and buttons). The button entitled "Grading" is used in the end of the testing to store the student's answers in database before the evaluation stage. Results of the tests by individual lessons are displayed separately. Furthermore, feedback is given about each question. The feedback tells whether the answer is correct, not correct or partially correct, shows the correct answer, and provides the hyperlink to the lesson C_i for remediation. This feedback is a very important addition to the learning process.

5 COLLABORATION WITH *AHYCO*

5.1 Group Formation

The critical point in the beginning of group work is considered to be the group forming process [14]. *AHyCo*'s model provides large amount of information on students' success since throughout the semester students are evaluated with various types of tests. The group creation module takes the knowledge levels kfd_l for a domain $\mathcal{D}$ and students' test results tr_j for chosen tests as input data into the group creation process. Even the data like discussion activity statistics can be used in order to create group with desired configuration.

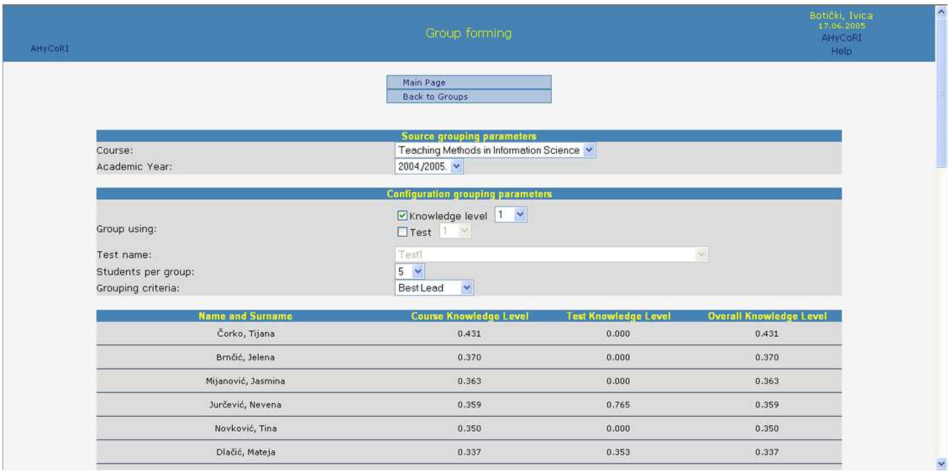


Fig. 3. Choosing source parameters for groups

Figure 3 shows a part where a teacher can choose input parameters to determine source data used in the group formation process (more specifically, a course and an academic year). After choosing source parameters, grouping parameters of the algorithm *ALG* need to be set in order to fine tune group forming process.

5.2 Asynchronous Communication Using Forum

AHyCo's data layer had to be expanded in order to enable many-to-many asynchronous communication using forum. Expansion was quite challenging because it included creating a model which was general enough to support various means of communication and, in the other hand, robust enough to enable painless integration into *AHyCo* [16]. Multi – threaded forum was chosen to be implemented since its structure satisfied user's need for easy and efficient message browsing and manipulation. *AHyCo*'s forum is implemented as a set of Web pages so that students could navigate and post forum messages more easily.

Every theme TH_k can be assigned to a group G_i so that its members can participate in the associated discussions. This means that students are presented only with information they need – groups and themes related to their course which their teacher approved. These discussions can be classified as private because they are inaccessible to the members of other groups. Since one student posts a question or an opinion and the others read it and attach their replies to it, this is considered to be a continuous process and the sequence of posts (or a thread) can go on for an indefinite period of time.

The role of the teacher is to monitor interactions, encourage students in communication and provide feedback on their questions about content or collaborative work [20, 24], and also teachers are responsible for creating groups $\mathcal{G}$ and discussion themes $\mathcal{TH}$.

5.3 File Upload Subsystem

When dealing with groups of students who are required to work on project through a longer period of time, one of the most important tasks in a LMS is to enable easy exchange of binary files, the most important being group's work progress reports. During the semester students are required to create reports and various files as a result of the given assignments which are then shared within the group. In that way teachers can monitor group work progress as well. Therefore, an intuitive interface was created to facilitate these tasks. Every student is allowed to publish a file within the group workspace(s) he or she is allowed to access. Student uploads a file into the virtual directory of his/her group G_i.

This module enables students to share data and, again, provides them with a certain amount of adaptiveness so they are not overloaded with too much information (number of files), but only with those they need to see. Teachers benefit from this kind of a system because they have constant access to information on group progress which enables them to entice inactive groups (ones who do not fulfill their duties in time).

5.4 Group Grading and Evaluation

One of the most stressful and complicated jobs in the end of a semester is grading and evaluation of students' work. When students are grouped and given a task, the results of their work can be evaluated by their colleagues and/or by their teachers. *AHyCo* provides both possibilities combining them into a unified way of determining overall final grade for a group's project.

After the group G_i has finished its work for current course (and it is possibly published using *AHyCo*'s upload subsystem) every student S_j can evaluate its results. The group can be evaluated through the set of criteria proposed by a teacher: Every criterion is weighted and final scores are shown to every student at any time so he or she can be informed of his or her group's success. Criteria depend on the kind of group project. Usually the teacher determines a set of criteria together with their weights through the *AHyCo*'s authoring module.

6 LEARNING ACTIVITIES AND EVALUATION

AHyCo is currently being used for teaching the course "Teaching Methods in Information Science". This course is a fourth year undergraduate course within a mathematics and information science study program at the University of Rijeka. Students are trained for teachers in elementary and high schools and in the context of this course they can learn how to implement various teaching and learning approaches for different information science lessons. They are also introduced to the use of information and communication technology (ICT) in education. The number of students taking the course in one term is between 15 and 25.

AHyCo modules have been generated for a part of the course since we utilize a mixed model of face-to-face and online learning. Main activities for the course are: online presentation and testing using *AHyCo* system, students' seminar papers, online discussions, and students' group project.

In order to evaluate the students' attitude concerning new e-learning approach using *AHyCo* the questionnaire about the effectiveness and quality of *AHyCo* as a teaching resource was conducted. The survey was anonymous and was conducted using *AHyCo* surveying subsystem. According to the questionnaire results, students accepted the new way of online collaborative learning with *AHyCo* quite well. For example, 72.73 % of students did not lack any educational aspect included in traditional teaching. About 18 % of students could not decide about this and the rest missed f2f lecturing and "live contact" with the teacher and the colleagues. If they could choose between a new way of learning using *AHyCo* and a traditional lecturing, about 90 % of students would prefer *AHyCo*. There were 36.6 % of students who would like to use *AHyCo* for some other courses.

The results have shown that students consider forum (46 %) and *AHyCo* modules (36 %) to be the most usable components of the system. 91 % of students considered that they learned the most from the collaborative work on courseware development. The rest preferred learning from *AHyCo* modules rather than focusing on seminar papers or discussions.

All students passed the exam from the first try on the first exam session in July. The students' average mark was quite high (3.68; the range is from 1 to 5).

7 CONCLUSIONS AND FUTURE DEVELOPMENT

In this paper we have presented *AHyCo* (Adaptive Hypermedia Courseware) system for development and distribution of the adaptive web-based courseware, with special focus on the implementation of collaborative learning support. Our approach to the collaborative learning and asynchronous communication using *AHyCo* subsystem for organizing learning groups and forum has been described.

Currently, we are working on further development of the subsystem for group work and CMC. Although *AHyCo*'s collaborative subsystem has reached a level which satisfies its current users (both teachers and students), many further enhancements are planned in order to make it even more attractive:

1. Internal messaging system – another form of asynchronous communication which gives students an opportunity to communicate individually without pressure.
2. Synchronous whiteboard and chat – these very popular means of communication will be incorporated into *AHyCo* to extend its interactivity.
3. Enhancement of the existing group creation algorithms and research into the possibility of combining existing group algorithms.
4. Adaptive regrouping based on students' knowledge level – with this feature *AHyCo* will be able to monitor students' progress and adaptively regroup students. New groups will reflect current knowledge levels changes in a way that parameters set by teacher allow.
5. Progress supervision and deadline monitoring – the aim of this feature is to help teachers when dealing with large number of students and deadlines for multiple tasks. Its main purpose is to track students' progress and help manage various activities and events.

AHyCo is currently being used in teaching "Teaching Methods in Information Science" course by utilizing a mixed model of face-to-face and online learning. In order to explore students' attitude concerning online collaborative learning, a questionnaire about students' acceptance of *AHyCo* as a teaching resource was conducted and the results show students accepted the new way of online collaborative learning with *AHyCo* quite well.

All students passed the exam from the first try with quite high average mark. Therefore, we can conclude that collaborative online learning was successful, probably because it requires continuous active participation during the academic year and more personal responsibility and concentration when learning. In that way, this approach to learning reduces the time needed for preparing the exam, contributes to successful passing of the exam and ensures deep level learning.

The students who had used the system have evaluated it favorably so we are quite satisfied with the results. The engagement of other teachers at University of Rijeka who would prepare the courseware from diverse areas (math, physics, pedagogy, psychology, art etc.) is planned. Based ontheir suggestions, the work on the *AHyCo* development will continue.

Acknowledgment

The research and development of the *AHyCo* system has been conducted under the projects "Adaptive Hypermedia Courseware" (reference number 2002-085), "Computing Support to Education" (reference number 0036041), and "Teaching Methods of Information Science Online" (reference number 2004-119), supported by Croatian Ministry of Science, Education and Sports.

REFERENCES

[1] *AHyCo* Web Site (in Croatian). Availaible on: `http://ahyco.ffri.hr/ahyco/`.

[2] Anjaneyulu, K.: Concept Level Modeling on the WWW. Proceedings of AI-ED 97, 8th World Conference on Artificial Intelligence in Education, Knowledge and Media in Learning Systems. Kobe, Japan, 1997.

[3] Baumgarthner, P.: Communication and Interactions in eLearning. In: Proceedings of eLearning Conference: Towards a Learning Society, Brusseles, Belgium, 2005, pp. 120–130.

[4] Blackboard and WebCT Web Site. Availaible on: `http://www.blackboard.com/webct/`.

[5] Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction. Vol. 6, 1996, pp. 87–129.

[6] Brusilovsky, P.: Adaptive and Intelligent Technologies for Web-Based Education. Künstliche Intelligenz, Special issue on intelligent systems and teleteaching, Vol. 4, 1999, pp. 19–25.

[7] Brusilovsky, P.: KnowledgeTree: A Distributed Architecture for Adaptive E-Learning. In: Proceedings of the Thirteenth International World Wide Web Conference, New York, 2004.

[8] Constantino Gonzalez, M. A.—Suthers, D.—Escamilla De Los Santos, J. G.: Coaching Web-Based Collaborative Learning Based on Problem Solution Differences and Participation. International Journal of Artificial Intelligence in Education, Vol. 13, 2003, Nos. 2–4, pp. 261–297.

[9] Da Silva, D. P.: Concepts and Documents for Adaptive Educational Hypermedia: a Model and a Prototype. In: Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, HYPERTEXT '98, Pittsburgh, USA, 1998.

[10] De Bra et al.: AHA! The Adaptive Hypermedia Architecture. In: Proceedings of the ACM Hypertext Conference, Nottingham, UK, 2003.

[11] Dohmen, F.: Internationalising eLearning. In: Proceedings of eLearning Conference Towards a Learning Society, Brusseles, Belgium, 2005, pp. 120–130.

[12] Eklund, J.—Brusilovsky, P.: Individualising Interaction in Web Based Instructional Systems in Higher Education Presented at The Apple University Consortium's Academic Conference, Melbourne, Australia, 1998.

[13] Gaudioso, E.—Boticario, J.Ġ.: Towards Web-Based Adaptive Learning Communities. In: Proceedings of Artificial Intelligence in Education, Sydney, Australia, 2003.

[14] Gross Davis, B.: Tools for Teaching. Jossey-Bass Publishers, San Francisco, 1993.

[15] Hoic-Bozic, N.: Adaptive Hypermedia Courseware (Ph. D. Thesis in Croatian). Zagreb, Croatia, 2002.

[16] Hoic-Bozic, N.—Mornar, V.—Boticki, I.: Collaborative Learning in *AHyCo* Online Learning System. In: Proceedings of ITI 2005, Cavtat, Croatia, 2005, pp. 247–252.

[17] Hoic-Bozic, N.—Mornar, V.: *AHyCo*: a Web-Based Adaptive Hypermedia Courseware System. Journal of Computing and Information Technology – CIT, Vol. 13, 2005, No. 3, pp. 165–176.

[18] Hoic-Bozic, N.—Mornar, V.: Authoring of Adaptive Hypermedia Courseware Using *AHyCo* System. In: Advances in Web-Based Education: Personalized Learning Environments, Hershey, Idea Group, 2005, pp. 253–276.

[19] Jaques, D.: Learning in Groups: A Handbook for Improving Group Work. 3rd ed., London, Kogan Page, 2001.

[20] Kemery, E. R.: Developing On-Line Collaboration. In: Web-based Learning and Teaching Technologies – Opportunities and Challenges, Hershey, Pa: Idea Group Publishing, 2000, pp. 226–244.

[21] McCormack, D.—Jones, D.: Building a Web-Based Education System. New York, Wiley, 1997.

[22] *Moodle* Web Site. Availaible on: `http://moodle.com/`.

[23] Ng, K.—Abramson, B.: Uncertainty Management in Expert Systems. IEEE Expert, Vol. 2, 1990, No. 5, pp. 29–47.

[24] Purcell-Robertson, R. M.: Interactive Distance Learning. In: Distance Learning Technologies: Issues, Trends, and Opportunities, Hershey, Pa, Idea Group Publishing, 2000, pp. 16–21.

[25] Rios, A. et al.: Internet Based Evaluation System. In: Artificial Intelligence in Education: Open Learning Environments, IOS Press, Amsterdam, Netherlands, 1999, pp. 387–394.

[26] Soller, A.—Lesgold, A.: A Computational Approach to Analysing Online Knowledge Sharing Interaction. Artificial intelligence in education: Shaping the future of learning through intelligent technologies. IOS Press, Amsterdam, 2003, pp. 253–260.

[27] Specht, M. et al.: Authoring Adaptive Educational Hypermedia in WINDS. In: Proceedings of ABIS2001, Dortmund, Germany, 2001.

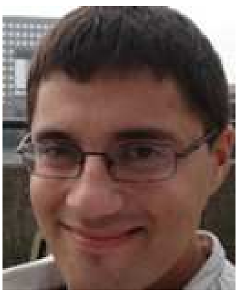
[28] Tuckman, B. W.—Jensen, M. A. C.: Stages of Small Group Development Revisited. Group and Organizational Studies, Vol. 2, 1977, pp. 419–427.

[29] Weber, G.—Kuhl, H. C.—Weibelzahl, S.: Developing Adaptive Internet Based Courses with the Authoring System NetCoach. In: Proceedings of Third Workshop on Adaptive Hypertext and Hypermedia, Sonthofen, Germany, 2001.

[30] Wu, H.—Houben, G.—De Bra, P.: AHAM: A Reference Model to Support Adaptive Hypermedia Authoring. In: Proceedings of the Conference on Information Science, Antwerpen, 1998.

Natasa Hoic-Bozic is an assistant professor at the Department of Information Science, University of Rijeka. She received Ph. D. degree in computer science at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. She currently lectures the courses Multimedia Systems, Seminar on Hypermedia, and Teaching Methods in Information Science. Her main research interests include information and communication technology (especially hypermedia and Internet) in education. She participated in several research projects and was a principal researcher of the project "Teaching Methods in Information Science Online".

Vedran Mornar is a Professor of Computer Science at Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia, where he currently teaches several graduate and undergraduate computing courses. He himself graduated and received his Ph. D. degree in computer science at the same university. As a Fulbright scholar, he studied at University of Southern California, Los Angeles for an academic year. His professional interest is in application of operational research in real world information systems, database design, development and implementation. He is an editor of international journal "Computing and Information Technology". In the period from 2002 to 2006, he was the vice dean for education. Since October 2006, he has been the dean.



Ivica Boticki is an assistant at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia, where he is currently involved into several undergraduate courses: Programming, Algorithms and data structures and Programming paradigms and languages. At this point he is working on his Ph. D. in computer science. His main professional interests include information systems, databases, software engineering and social computing. He is currently engaged in several projects in the fields of e-learning, intelligent transportation systems and on connecting developers with the academic society.