Fuzzy User Modeling for Adaptation in Educational Hypermedia

Alenka Kavčič, *Member, IEEE*

***Abstract*—Education is a dominating application area for adaptive hypermedia. Web-based adaptive educational systems incorporate complex intelligent tutoring techniques, which enable the system to recognize an individual user and their needs, and consequently adapt the instructional sequence. The personalization is done through the user model, which collects information about the user. Since the description of user knowledge and features also involves imprecision and vagueness, a user model has to be designed that is able to deal with this uncertainty. This paper presents a way of describing the uncertainty of user knowledge, which is used for user knowledge modeling in an adaptive educational system. The system builds on the concept domain model. A fuzzy user model is proposed to deal with vagueness in the user's knowledge description. The model uses fuzzy sets for knowledge representation and linguistic rules for model updating. The data from the fuzzy user model form the basis for the system adaptation, which implements various navigation support techniques. The evaluation of the presented educational system has shown that the system and its adaptation techniques provide a valuable, easy-to-use tool, which positively affects user knowledge acquisition and, therefore, leads to better learning results.**

***Index Terms*—Adaptive hypermedia, fuzzy logic, learning systems, personalization, user modeling.**

I. Introduction

DATA and information play a very important role in our information society. Knowledge is becoming more and more valued and learning is a part of our everyday life. New demands of the competing world and development of new technologies have also changed the traditional educational systems, which now use better and more efficient teaching and learning methods. The integration of new technologies in the field of education offers new challenges and opportunities in distance learning, lifelong learning, and e-learning in general.

E-learning is especially important for companies and industry. According to research completed in the U.S. [1], companies use e-learning as a preferred way of education for their employees, as 40% of all education in companies in the year 2000 was realized through e-learning. The prognosis for the next years is to double the number each year. Therefore, e-learning is a trend that cannot and may not be ignored as it will play an even more important role in the future.

One area of research that brings innovative solutions to educational systems is certainly the field of adaptive hypermedia [2], [3]. Through the incorporation of more complex intelligent tutoring techniques into traditional hypermedia, it enables educational systems to recognize individual users and their needs, and consequently adapt the instructional sequence. Such systems are able to adapt information and its presentation to each individual user, and dynamically support the navigation of the user through the hypermedia material. This ability to adapt to an individual user's needs can significantly improve the teaching process since it has been shown that in many cases, individualized tutoring is one of the most successful teaching methods [4].

This paper focuses on fuzzy user modeling and the way it deals with uncertainty in the description of knowledge. An overview of adaptive educational hypermedia, together with the domain and the user knowledge representations, is given first. Fuzzy user modeling and its implementation in an adaptive educational hypermedia system follow. The adaptation of the system concentrates on navigation support techniques, which are also briefly described. In the end, the results of an evaluation of the designed system are presented and discussed.

II. Educational Hypermedia

By educational hypermedia we mean various hypermedia systems which are designed for use in education and have the ability to adapt to the individual user's needs. Such systems improve basic hypermedia functionality through incorporation of intelligent tutoring techniques to enable personalization of the system. Although they still support free browsing of the learning material and offer the freedom of exploratory learning, these systems are able to dynamically adapt the instructional sequence to the individual user knowledge level and learning goals, provide intelligent guidance, and support the user in acquiring knowledge [5], [6]. They can adapt displayed information and its presentation, and dynamically support navigation through hypermedia material.

Adaptive hypermedia, in general, are the results of investigations in the fields of hypermedia and user modeling, and represent a new research direction in adaptive interfaces, based on user modeling [7].

An adaptive hypermedia system has the following main features [8].

- It is based on hypermedia.
- It includes a domain model, which is composed of a set of elementary pieces of knowledge and their relationships in the information space.
- It maintains an explicit user model that records individual user properties.
- It is able to adapt some visual or functional parts of the system according to the user model.

Manuscript received January 31, 2003; revised April 13, 2004. This paper was recommended by Guest Editors J. L. Fernández and A. Artés.

The author is with the Faculty of Computer and Information Science, University of Ljubljana, Ljubljana 1000, Slovenia (e-mail: alenka.kavcic@fri.uni-lj.si).

Digital Object Identifier 10.1109/TSMCC.2004.833294

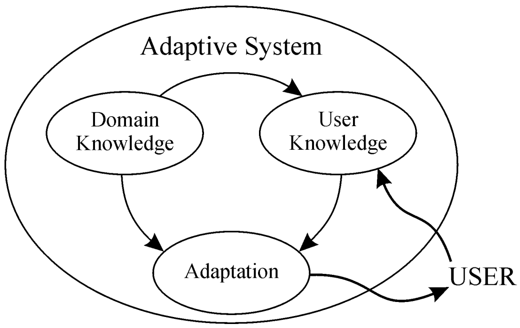

Fig. 1. Adaptive system.

An adaptive system has to model the domain knowledge to know what to teach the user. It also models the user knowledge to know whom to teach (who the user is), and uses adaptation to adjust the method of teaching to the user (how to teach) [9]. A typical adaptive system thus consists of at least three parts, as shown in Fig. 1: user module, domain module, and adaptation module.

The domain module presents the domain knowledge and is also the base for user knowledge description (the user module). Both are used by the adaptation module for altering the functionality of the system. The user module also records the user interaction with the system, which affects the user knowledge. To enable adaptation, the system has to be aware of the teaching domain, the individual users, and their knowledge, and has to monitor their learning progress. Thus, a domain model and a user model are the core parts of such a system.

A. Domain Model

The knowledge of the teaching domain is represented in the domain model, which is one of the most important components of an adaptive system.

Domain knowledge is logically partitioned into smaller elements or concepts [5]. Therefore, a finite set of domain concepts can represent the entire teaching domain. The number of such concepts depends on the teaching domain and selected granularity. A given domain can be described using many simple concepts or fewer, but more complex concepts.

Let C be the set of all domain concepts c_i. The teaching domain can be described as

$$\begin{aligned} C &= \{c_1, c_2, \ldots, c_n\} \\ |C| &= n \end{aligned} \tag{1}$$

where n is the total number of domain concepts.

There are some learning dependencies between the concepts, which can be represented by the ordered prerequisite relation R (the sign $\prec$ stands for this relation)

$$\begin{aligned} &R \subseteq C \times C \\ &R = \{(c_i, c_j) : c_i \prec c_j; c_i, c_j \in C\}. \end{aligned} \tag{2}$$

When two domain concepts are related $(c_i \prec c_j)$, the first concept (c_i) has to be known to understand the second one (c_j). We say that concept c_i is a prerequisite of the concept c_j. Thus, the user can start learning the second concept only after mastering the first one.

All domain concepts are named and together with their relationships form an ordered acyclic graph G_D, which we use for modeling the domain. The domain concept graph is described as

$$G_D = (C, R) \tag{3}$$

where C is the set of domain concepts and R stands for the prerequisite relation on the domain concepts, as defined in (1) and (2), respectively.

B. User Model

The information specific to each individual user is collected in the user model, which describes the features of the learner [9]. It is the foundation for system adaptation, as it saves all needed information about a particular user. Through the user model, the system can distinguish between different users and adapt itself to particular user needs. Without the information from the user model, all users would be treated equally [5].

A perfect user model would include all features of the user behavior and knowledge that affect their learning, performance, and efficiency [9]. Because the construction of such a complex model is very difficult, simplified models are used in practice.

Three aspects have to be considered regarding the user model [10]: what information about the user is included in the model and how it is obtained, representation of this information in the system, and the process of forming and updating the model.

Because we are dealing with an educational system, the key information stored in the user model is the user knowledge about the teaching domain. This information is constantly being collected during the learning process through the user interaction with the system. It is also used for updating the user model.

For the user model representation, a strict overlay model over the teaching domain is used. Therefore, the user model is based on the domain model and the same representation can be used for both models. For each domain concept, the user model saves a corresponding value, which estimates user knowledge of that concept.

The next section describes the user model representation in more detail.

III. Dealing With Uncertainty of Knowledge

The description of knowledge can be quite vague and imprecise, and may include a great deal of uncertainty. There are many mathematical theories for expressing uncertainty and which could be used to deal with the uncertainty in the description of the user knowledge.

A. Methods for Managing the Uncertainty

In general, four different approaches are used for modeling the user, considering also the uncertainty in describing the knowledge [11]:

- rules with certainty factors;
- fuzzy logic;
- Bayes probability networks;
- Dempster–Shafer theory of evidence.

Certainty factors are one of the first techniques used for solving the problem of uncertainty in the knowledge description. This model was developed in an expert system that diagnosed certain infectious diseases, called MYCIN [12], where the facts and rules had added a certainty factor (a measure of confidence). A consultant system for mineral exploration Prospector [13] uses a similar approach (rules with confidence measures). Some systems, like ELM-ART [14] that builds on rules, have developed their own heuristics for dealing with the problem of knowledge uncertainty. Certainty factors are easy to understand and enable easy realization and implementation.

Fuzzy logic offers the possibility of processing input data that are verbally imprecise. It allows natural description of knowledge and inference in the form of imprecise concepts, operators, and rules. Some examples of systems that implement fuzzy logic for describing the user and their knowledge include UNIX Consultant [15], ML-Modeler [16], Sherlock II [17], ABITS [18], and Hypernet [19]. The user modeling in UNIX Consultant is done through the KNOME component, which uses fuzzy rules for predicting user knowledge level and its updating. ML-Modeler and Sherlock II use fuzzy methods and inference rules, whereas ABITS uses fuzzy numbers for user knowledge evaluation. User modeling in Hypernet is based on neural networks, but is combined with fuzzy sets and fuzzy rules.

Bayesian probability networks are one of the most common ways of describing uncertainty and dealing with it. The probability methods for describing user knowledge are used in many systems, such as OLAE [20], POLA [21], Andes [22], HYDRIVE [23], APHID-2 [24], and KBS Hyperbook [25]. Bayesian networks offer a solid theoretical base, are consistent, and useful in any kind of problems. They are very powerful in inference (diagnostic and predictive), but need a complete model (variables and relations, all knowledge has to be coded in the model) together with the estimations of all parameters (*a priori* and *posteriori* probabilities).

Dempster–Shafer evidence theory is not so widely used compared to probability theory, but is more general (and a semantically richer mechanism). Moreover, the decision based on the analysis results is more complicated. An example system is PHI [26], which offers intelligent help to the users of electronic mail.

The main common problem of all four approaches is how to define suitable initial parameters. Although this problem is more evident in Bayesian networks, where *a priori* and *posteriori* probabilities have to be defined beside the network structure, it is present in Dempster-Shafer theory (basic assignments), fuzzy logic (membership functions), and certainty factors as well.

In our case of educational hypermedia, we want to represent and model the user knowledge in the system, considering also the uncertainty in its description. The uncertainty here does not arise from ambiguity in determining set memberships (a problem of classification), but from vague and unsharp boundaries of sets (e.g., it is difficult to draw a line between an unknown and a known concept). Whereas fuzzy measures theory (which covers the probability theory and the theory of evidence) is designed for solving the problems of ambiguity, fuzzy set theory seems more appropriate for this second kind of problem (vagueness) [27].

The main advantages of using fuzzy sets are in the systems where we process an inexact user input in a verbal form, or use inference or manipulate knowledge which can be naturally described and explained in the form of imprecise concepts, operators, and rules. The use of linguistic variables and rules in natural language [27] is closer to human thinking, and hence, easier to comprehend and more straightforward to construct. It also needs little calculation and is easier to initialize.

The user model saves a system's estimation of what the user has already mastered. This knowledge is then manipulated using inference to get a broader picture of the user's knowledge of the domain. We use fuzzy set theory [28] for user model description: the user model representation is based on fuzzy sets and model updating on linguistic rules [29]. Therefore, we have to redefine the domain and user representations, considering also the uncertainty of knowledge description.

B. *Brief Introduction to Fuzzy Sets*

Fuzzy set theory was formalized by Prof. L. Zadeh [28] in 1965. It extends the limiting bivalent sets in a way that allows smooth transition between sets, especially convenient for describing natural phenomena. In fuzzy sets, an element is not strictly a member or not a member of a set, but can also be only partially in the set, which means it is present in the set to some extent. There is not only black and white, but also a series of gray shades in between. Hence, a set is called fuzzy when its membership function takes values in the unit interval [0,1] rather than in the {0,1} as in the classical logic.

Let X be the universe of discourse and its elements marked as x. A fuzzy set A with a membership function μ_A is defined as

$$\begin{aligned} &A \subseteq X \\ &A = \left\{ \frac{\mu_A(x_i)}{x_i} \right\}, \quad x_i \in X \\ &\mu_A : X \to [0,1]. \end{aligned} \tag{4}$$

The membership function μ_A gives the degree of membership to the set A.

A fuzzy set is empty if the value of the membership function for this set is zero for all elements of the universal set

$$A = \varnothing : \quad \forall x \in X : \quad \mu_A(x) = 0. \tag{5}$$

A complement, union, and intersection of fuzzy sets can be defined as

$$\begin{aligned} \bar{A} &: \forall x \in X : \mu_{\bar{A}}(x) = 1 - \mu_A(x) \\ A \cup B &: \forall x \in X : \mu_{A\cup B}(x) = \min\left[1, \mu_A(x) + \mu_B(\mathrm{x})\right] \\ A \cap B &: \forall x \in X : \mu_{A\cap B}(x) = \max\left[0, \mu_A(x) + \mu_B(x) - 1\right]. \end{aligned} \tag{6}$$

Here, we have used bold union and bold intersection. This set of operations is not the only one possible; there are many others defined in the literature. More can be found in [27].

C. *Domain Knowledge Representation*

The prerequisite relation between the domain concepts can be of two types [4]: essential or supportive. When one concept

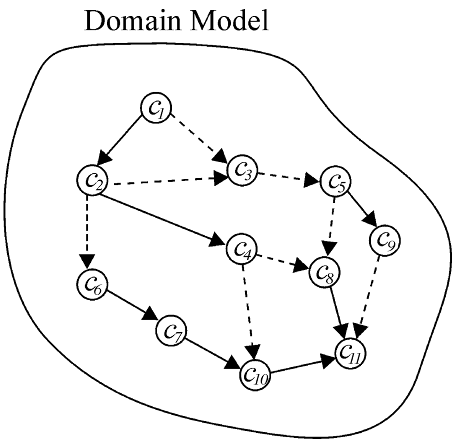

Fig. 2. Domain concept graph G_D^*.

is an essential prerequisite to the other, knowing it is necessary for learning and understanding the second concept. On the other hand, the supportive prerequisite concept just helps with understanding and learning of the related concept.

The set of prerequisite relations R is thus partitioned into two disjoint sets: a set of essential prerequisite relations E and a set of supportive prerequisite relations S, where

$$\begin{aligned} R &= E \cup S \\ E \cap S &= \varnothing. \end{aligned} \tag{7}$$

The essential prerequisite relation E can exist between two concepts only to some extent; therefore, we define it as a fuzzy relation [29]. The supportive prerequisite relation is always fully present (if it is present at all) and is therefore defined as a normal crisp relation.

The fuzzy essential prerequisite relation E is a fuzzy set, which is defined by its membership function $\mu_E(c_i, c_j)$

$$\begin{aligned} &E \subseteq C \times C \\ &\mu_E : C \times C \rightarrow [0,1] \\ &E = \left\{ \frac{\mu_E(c_i, c_j)}{(c_i, c_j)}; c_i, c_j \in C \right\}. \end{aligned} \tag{8}$$

Since the essential prerequisite relation is defined as a fuzzy relation, the domain concept graph is also a fuzzy structure. We represent it by a triple

$$G_D^* = (C, E, S) \tag{9}$$

where C is a set of domain concepts, E is a fuzzy essential prerequisite relation, and S is a crisp supportive prerequisite relation.

A domain concept graph, which models the domain knowledge, is an ordered fuzzy graph with nodes representing domain concepts and arcs connecting two interrelated concepts. The domain concept graph in Fig. 2 depicts such a fuzzy structure.

The domain in the figure consists of 11 domain concepts ($n = 11$). The two types of relations between the concepts E and S are represented by dashed and solid lines, respectively. The concept c_1, for example, is a supportive prerequisite for concept c_2 and essential prerequisite for concept c_3. Similarly, concept c_3 has two essential prerequisites: c_1 and c_2, and is itself an essential prerequisite for c_5. The corresponding values of the membership functions for essential prerequisite relation are not specified in the figure.

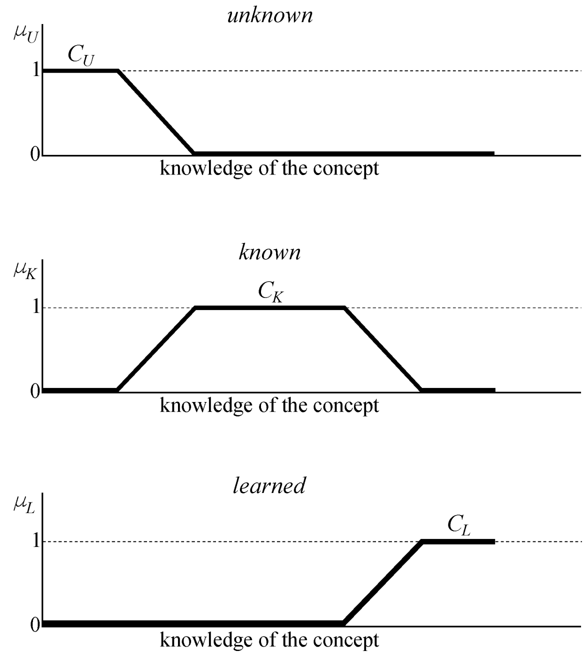

Fig. 3. Membership functions for fuzzy sets of unknown (C_U), known (C_K), and learned (C_L) concepts.

D. User Knowledge Representation

As the user model is an overlay over the domain model, the user knowledge is a subset of the domain knowledge and the domain knowledge representation can be used for describing user knowledge. The user model is a subgraph of the domain concept graph, where each concept of the subgraph has some additional properties attached, which explain the user knowledge of this concept.

The user knowledge of each domain concept can be described using a linguistic variable *concept knowledge*, which takes three possible values: *unknown, known*, and *learned*. Each linguistic term is associated with a fuzzy set, each of which has a defined membership function. We indicate the three fuzzy sets with C_U, C_K, and C_L, and their membership functions with μ_U, μ_K, and μ_L, respectively. The membership functions are shown in Fig. 3.

The value of the language variable *concept knowledge* for a certain concept, which describes the user knowledge of that concept, evolves from *unknown* to *known* and *learned* as the estimated user knowledge of that concept increases. The membership functions are partially overlapping, although the sum of their values is always one.

User knowledge representation can be associated with a graph G_U^*, which is a subgraph of G_D^* from (9)

$$G_U^* = (C, E, S, L) \tag{10}$$

where C is a set of domain concepts, E is an essential prerequisite relation on the domain concepts as defined in (8), S is a supportive prerequisite relation on the domain concepts, and L is a set of labels, which is equal to the set of values of the linguistic variable *concept knowledge*.

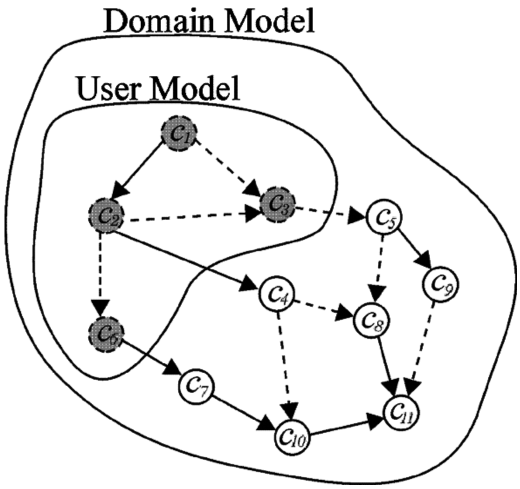

Fig. 4. User knowledge representation G_U^*.

The set of domain concepts C can be labeled by a set of labels L. Each label corresponds to a certain concept just to a certain degree, thus the labeling is a fuzzy process. The corresponding graph G_U^* is a fuzzy structure with fuzzy nodes and fuzzy relations [29].

Fig. 4 illustrates the user model as an overlay over the domain model (a subgraph of the domain concept graph). Dashed circles in the graph represent the fuzzy nodes/concepts. The concepts c_1, c_2, c_3, and c_6 are already known/learned to some extent and together with their labels (levels of knowledge) form the user model. The corresponding labels are not specified in the figure.

The three fuzzy sets of unknown, known, and learned concepts describe user knowledge of the domain concepts. User knowledge of a particular domain concept c is therefore expressed by providing values of membership functions for the three fuzzy sets, which we describe with a triple

$$(\mu_U(c), \mu_K(c), \mu_L(c)) \quad \text{or} \quad (\mu_U, \mu_K, \mu_L). \tag{11}$$

The following equations hold for the triple of membership functions for a given concept (see Fig. 3):

$$\begin{aligned} &\mu_U + \mu_K + \mu_L = 1 \\ &\mu_U > 0 \Rightarrow \mu_L = 0 \\ &\mu_L > 0 \Rightarrow \mu_U = 0. \end{aligned} \tag{12}$$

If we apply bold union and bold intersection, as defined in (6) and using (12), we get the following constraints for the three sets:

$$\begin{aligned} C_U \cup C_K \cup C_L &= C \\ C_U \cap C_L &= \varnothing. \end{aligned} \tag{13}$$

The intentional limitation of our user model is that the user knowledge in the model always expands. Even when the user performs poorly, it never reduces. For each concept, the value of the membership function for a set of learned concepts can therefore only increase, and the value of the membership function for a set of unknown concepts can only decrease, or both can stay the same. This way, the knowledge of all concepts only raises or keeps its highest level. We always change one (the most significant) component of the triple (μ_U, μ_K, μ_L). The other two components change accordingly, based on the three equations set in (12).

E. Knowledge Determination and Updating

The user knowledge of domain concepts changes (increases) during the user interaction with the system. Consequently, the user model also changes to reflect the current user understanding of the teaching domain. The main principle for gathering information about the user knowledge is checking tests results and analyzing visits to the learning units.

For describing the knowledge a particular user has about a domain concept, we use the linguistic variable *concept knowledge*, which can take three values. If we expand them with two quantifiers (*partially* and *completely*), we get the following five values, which the linguistic variable *concept knowledge* can take: *completely learned, partially learned, completely known, partially unknown*, and *completely unknown* (note that *partially known* could be *partially learned* or *partially unknown*, and is already covered in one of those terms).

1) Model Initialization: The user model is initialized using the results of a short pretest, which each new user is required to take. The user knowledge of each concept is considered *completely unknown* at the beginning, which corresponds to a triple of membership functions (1,0,0), as described in (11). After the pretest, the values are updated and the concepts that were correctly answered in the pretest become *completely learned* with corresponding triple (0,0,1). Thus, each domain concept becomes either *completely learned* or remains *completely unknown*.

2) Model Updating: Tests are used for checking how well a particular concept is learned and a set of corresponding test questions is provided for each learning unit. The most significant changes in user knowledge of the domain can be recorded, when the user takes the test associated with the learning unit.

After the user passes the test on one domain concept, this concept becomes *learned*. If the test questions are not answered satisfactorily, the value of the variable *concept knowledge* does not change. A new value of membership function μ_L for a set of learned concepts C_L is calculated based on the user answers to the test questions. It is assigned regardless of the previous knowledge level for this particular concept.

We also update the user model after each visit of the learning unit. In particular, first visits to units and visits to units which describe still not learned concepts have the biggest influence on the change of the user knowledge.

After the user visits a learning unit, the linguistic variable *concept knowledge* for the concept that is explained in the visited unit is estimated to be *known* to the extent that the supportive and essential prerequisites of the concept are *known* or *learned*.

A new value of membership function μ_K for a set of known concepts C_K is first set to 1 for this concept. Then it is modified based on the values of the prerequisites. If some of the essential prerequisites are still *unknown*, the value of μ_K decreases. If some of supportive prerequisites are not *unknown*, the value of μ_K increases.

This can be applied only to the concepts that are not *learned*, meaning their value of the membership function for a set of learned concepts C_L equals zero ($\mu_L = 0$).

3) Knowledge Value Propagation: Because the domain concepts are interrelated, we can also infer knowledge values of some concepts. This way, the knowledge of essential prerequisite concepts is deduced from the demonstrated concept knowledge. After every change of the value of *concept knowledge* for a domain concept, an inferring mechanism (knowledge value propagation) is triggered that updates the values of all essential prerequisite concepts of this concept. The knowledge value propagation algorithm is based on six rules in natural language and works recursively on all essential prerequisite concepts, until it reaches the basic concepts that have no prerequisites. The changes in the value of one concept are thus reflected in the values of all other concepts that are essential prerequisites to this concept.

As we use five linguistic terms (*completely learned, partially learned, completely known, partially unknown*, and *completely unknown*) for describing the concept knowledge, we can produce 25 linguistic rules regarding two related concepts c_i and c_j $(c_i \prec_E c_j)$. These rules determine the new knowledge level of an essential prerequisite concept c_i based on the demonstrated knowledge of the concept c_j. Because some of the rules can be merged, we end up with the following six.

Rule 1: If the concept c_i is *unknown* and the concept c_j is *completely unknown*, the concept c_i remains *unknown*.

Rule 2: If the concept c_i is not *unknown* (is *learned* or *completely known*) and the concept c_j is not *learned*, the concept c_i remains *learned/completely known*.

Rule 3: If the concept c_i is *completely unknown* and the concept c_j is *known*, but not *learned*, the concept c_i becomes *known*.

Rule 4: If the concept c_i is not *learned* and the concept c_j is *learned*, the concept c_i becomes *learned*.

Rule 5: If the concept c_i is *partially unknown* and the concept c_j is *known*, but not *learned*, the concept c_i increases its value of *known*.

Rule 6: If the concept c_i is *learned* and the concept c_j is *learned*, the concept c_i increases its value of *learned*.

The extent to which the concept c_i becomes *known/learned* depends on the values of *known/learned* for the concept c_j and the value of the prerequisite relation between both concepts. In other words, the actual values of membership functions $\mu_K(c_i)$ or $\mu_L(c_i)$ are calculated from the values of membership functions $\mu_K(c_j)$ or $\mu_L(c_j)$, and $\mu_E(c_i, c_j)$. Therefore, the rules can be described using membership functions for fuzzy sets C_K and C_L, taking in account the membership function for fuzzy prerequisite relation E, using product to combine the rules (aggregation), and applying the max function to combine the resulting values of a membership function (composition).

Rules 1 and 2 do not change the values of $\mu_U(c_i)$, $\mu_K(c_i)$, and $\mu_L(c_i)$. For rules 3 and 4, the new values of membership functions are calculated using (14)

$$\begin{aligned} \mu_K(c_i) &= \mu_E(c_i, c_j) \cdot \mu_K(c_j) \\ \mu_L(c_i) &= \mu_E(c_i, c_j) \cdot \mu_L(c_j). \end{aligned} \quad (14)$$

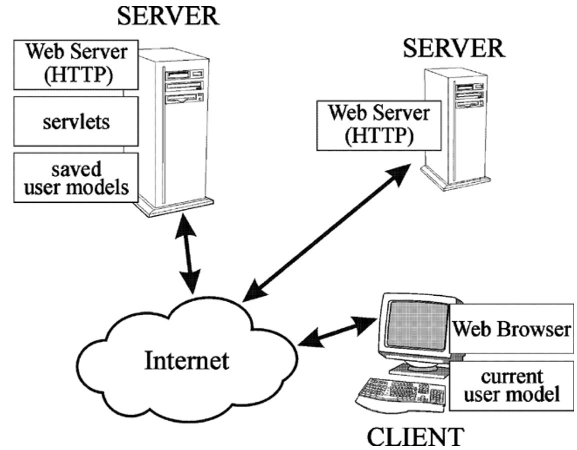

Fig. 5. Client–server model.

Rules 5 and 6 increase the current value of membership functions, using the following equations:

$$\begin{aligned} \mu_K(c_i) &= \max\left[\mu_K(c_i), \mu_E(c_i, c_j) \cdot \mu_K(c_j)\right] \\ \mu_L(c_i) &= \max\left[\mu_L(c_i), \mu_E(c_i, c_j) \cdot \mu_L(c_j)\right]. \end{aligned} \quad (15)$$

In (15), we use the max function for merging the two values of membership functions, according to our user model, where the user knowledge of the concept can only increase (never decreases). The max function also complies best with the constraints from (12).

IV. Fuzzy User Model Implementation

The proposed fuzzy user model was also implemented in a web-based educational system, which offered adaptation based on the described user model. The system was then tested in a real learning environment and its performance was evaluated.

The designed educational system is a web-based system and consists of two parts: the server and the client applications (see Fig. 5). All servers and clients are connected over the Internet.

On the client's side, we need only a web browser that supports Java applets. The client is responsible for forming and updating the user model during the session.

Server applications are implemented using Java Servlets [30] that take care of user authentication, processing the pretest results for user model initialization, and saving the current user model at the end of the session. Creation and maintenance of the user model is processed on the client, the server just stores the current model. This way, the client's resources are used for model updating and the network traffic is also reduced.

The system is designed for an arbitrary teaching domain. The teaching materials reside on web servers in the form of web documents, each representing one learning unit and describing one domain concept. They are accessed directly from the client. Learning units have no direct links to other units (links are separated from the page content); the interconnectivity of units is defined through the domain concept graph.

The system adaptation is carried out through different navigation support techniques [7], which help the user in navigation through the learning material. Navigation deals with the

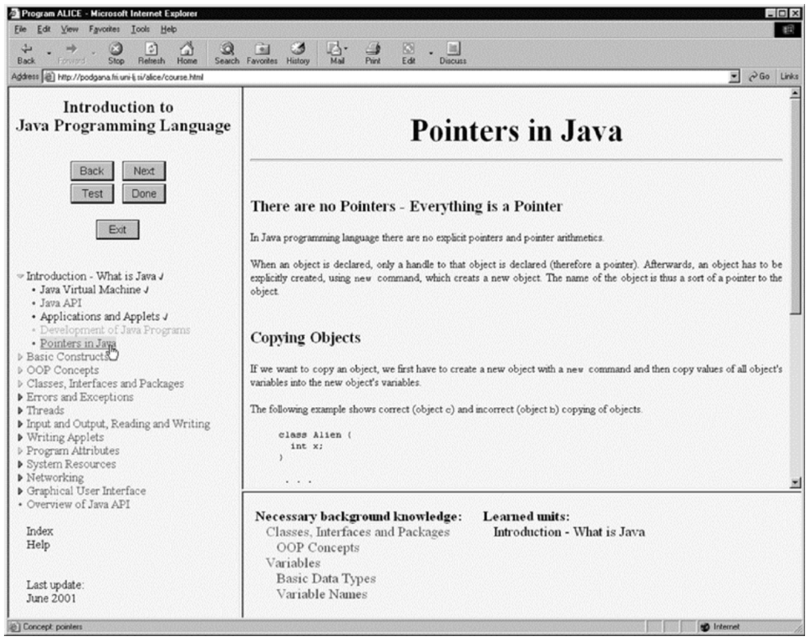

Fig. 6. Main window of the system.

problem of how to effectively access a certain hypermedia page [31] and various navigation support techniques are designed to help the user in achieving their goal.

The user always has a completely free choice in selecting learning units. The system only suggests the most appropriate options and, thus, supports users in their decisions. The user can select learning units from the table of contents (hierarchical navigation), lists of links to related units (relational navigation), using buttons to next and previous units for direct guidance (linear navigation), or through the index of all domain concepts (concept-based navigation) [29]. The main window of the system is illustrated in Fig. 6. The table of contents, all navigation buttons, and the link to the index are in the left-hand frame of the window. Links to related units are in the lower right-hand frame. The course contents are displayed in the upper right-hand frame.

All links are also of different colors, depending on the educational state of the unit the link is leading to (color annotation). The educational state of the unit is set depending on the user knowledge of the concept that the unit describes. Thus, the unit can be in one of the following states: learned, known, ready for learning, conditionally ready for learning, or not ready for learning, colored black, blue, green, orange, and red, respectively. The color annotation is used on all of the links to units: in the table of contents, in index, and in the lists of links to related units.

Relational navigation is achieved through the technique of adaptive link insertion, where the list of links to related units is composed on the fly and displayed in a separate window below the unit contents (see Fig. 6).

For each displayed unit, the system dynamically inserts groups of links to other relevant units, which depend on the currently displayed unit, the educational state of each unit, the level of user knowledge of the domain concepts, and the prerequisite relationships between the concepts. The groups of links composed in this way do not include all possible links from the current unit (hiding of links) and arrange links by their suitability (sorting of links).

Three different groups of links are prepared for each unit. The first one is a group of links that lead to the next suitable units for learning. These units are recommended to the user as the next learning step, because they logically continue the current unit and the user is already prepared to start learning them (has all necessary prerequisite knowledge for these units). The second group contains links to units that explain needed prerequisite knowledge for the current unit. The user has to learn them before studying the current unit. In the third group are links to all units that contain already learned concepts. These units help users to refresh their knowledge, to better orient in the domain hyperspace, and to form a cognitive map of the teaching domain.

V. System Evaluation

As our system falls in educational hypermedia domain, we try to express the benefits of adaptation as an increase in examination results [32]. Thus, the evaluation of the system builds on students' performance in exams after learning with the system. We prepared different versions of the system to test different aspects of possible learning improvements. We were interested mainly in the influence of use of adaptive systems on learning success. We focused on adaptive link insertion and color annotation of links.

As the described system is just a shell of an adaptive educational system, we had to integrate the teaching content into the system, to get a suitable system for testing. Even though an arbitrary teaching domain could be chosen, we decided on an "Introduction to Java" course, because of the advantages it offered in developing the learning units and testing the system. The course is based on the teaching material on the Java programming language [33], [34], which is also available on the web.

For the evaluation, we prepared three different versions of the system: a fully functional adaptive system as described in the previous section (herein marked as version A), an adaptive system without color annotation (version B), and a nonadaptive system (version C). The content of the course (learning units) and the test questions were the same for all three versions. The overall appearance of the three versions was quite similar; they differed only in the details that arose from functional differences between the versions (like the use of colors or displayed list of links to related units). Version A offered lists of links to related units as well as color annotation of all displayed links. Version B differed from version A only in the use of colors—all links were displayed in black. Version C had neither lists of related links nor color annotation of links, and the direct guidance always offered the next unit from the table of contents.

All adaptation in the system derives from the user model. Hence, both adaptive versions, A and B, depend on the constructed user model, which represents the estimated levels of user knowledge of the domain concepts. The source data for assessing user knowledge are the analyses of self-assessment results and visited units. This knowledge of the domain concepts is then converted into visual and functional adaptation of the system: color annotated links, constructed lists of related links, and calculated the most suitable next unit to study. We did not validate the user model accurateness in the study. Therefore, no data are available on how the fuzzy user model directly contributes to the system's adaptation. The main advantage of using fuzzy logic for user modeling in our system lies in easier construction and updating of the model because of the use of linguistic rules and variables.

A group of first year computer and information science students, all beginners in the Java programming language, used the system in our experiment. All students were randomly assigned to one of the three system versions A, B, or C. First, a short pretest was given to the students. After using the system for about an hour, they received a post-test to solve. We collected the test results of both tests together with navigation paths of the students and analyzed the gathered data.

A. Analysis of Test Results

The main goal of our experiment was to find out if a way of learning affects the result of the final exam in any way or, in other words, if the experimental factor "the use of a particular version of the system" had any effect on the average score in the final exam. As other studies [35], [36] show, adaptation can improve understanding and increase the learning effect. Therefore, we expected to get better results (higher scores) for the groups, which used adaptive versions of the system. We expected that students from groups A and B would perform better in the final-assessment test, because they had learned the subject using the adaptive systems.

The experiment was as follows: we tested three teaching methods (the use of a particular version of the system) on 80 students. The students were randomly divided into three groups; each group used one of the teaching methods (one version of the system). After the process of teaching, we tested the knowledge of the students and used the results of the post-test as our experimental data.

TABLE I
POST-TEST RESULTS FOR GROUPS A, B, AND C

Group	Num.	Mean	Median	Std.Dev.
A	27	11.22	12	2.89
B	27	11.81	12	2.63
C	26	11.31	12	2.28
Total	80	11.45	12	2.59

TABLE II
ANALYSIS OF VARIANCE ON POST-TEST RESULTS FOR GROUPS A, B, AND C

Source	SS	df	MS	F	F'	%
between	5.52	2	2.76	0.40	3.11	1.04
within	526.28	77	6.83			98.96
Total	531.80	79				

Because the students were randomly assigned to a group, we did not expect big differences between the groups in the pretest results. The analysis of the pretest results [29] confirmed our expectations: there were no statistically significant differences between the groups.

On the other hand, we expected that the system version used would affect the learning and consequently the result of the post-test. The collected results of the post-test for each group are summarized in Table I. Groups are marked A, B, and C according to the system version used. The columns in the table show the number of students in the group (Num.), the mean score of the post-test (Mean), its median (Median), and its standard deviation (Std. Dev.) for each group and the total. The maximum possible score is 17.

The obtained results for each group look quite similar and the detailed analysis confirms that there are no significant differences between the groups. We performed an analysis of variance (ANOVA) on post-test results (see Table II for details) and got the F value that is smaller than the critical value F'. Hence, the null hypothesis of no differences between the groups cannot be rejected, which means that the use of the system version statistically has no effects on the result of the post-test. The calculations are summarized in Table II. The columns in the table have the following meanings: the source of variability (between groups or within groups), the sum of squares (SS), the degrees of freedom (df), the mean square (MS), the F ratio (F), the critical F value for 0.05 alpha level (F'), and the percentage of total variability (%).

As shown in Table II, the variability between the groups produces only 1% of total variability of data and is therefore irrelevant. The obtained results were a surprise to us, because we expected bigger differences between the groups. A detailed examination of the collected data revealed the reason for such results. We supposed that the use of the system versions A and B implied also the use of adaptation that was integrated in both systems. After analyzing the user logs, we realized that this was not the case and that about half of the students did not use any adaptive features of the systems. Based on this fact, we analyzed the post-test results from a different perspective: the use of adaptive features of the system. The statistical data, summarized in Table III, show that the mean values are quite different; the difference is 2.2 points or 13% of the maximum possible score.

TABLE III
POST-TEST RESULTS REGARDING THE USE OF SYSTEM'S ADAPTIVE FEATURES

Adapt.	Num.	Mean	Median	Std.Dev.
yes	25	12.96	14	2.28
no	55	10.76	11	2.45
Total	80	11.45	12	2.59

TABLE IV
TWO-WAY ANALYSIS OF VARIANCE ON POST-TEST RESULTS REGARDING THE USE OF ADAPTATION AND GROUPS

Source	SS	df	MS	F	F'	%
adaptation	82.91	1	82.91	14.54	3.92	15.59
group	21.07	2	10.53	1.85	3.07	3.96
interaction	0.21	1	0.21	0.04	3.92	0.04
within	427.61	75	5.70			80.41
Total	531.80	79				

TABLE V
POST-TEST RESULTS FOR GROUPS A1, A2, B1, AND B2

Group	Num.	Mean	Median	Std.Dev.
A1	13	12.69	13	2.14
A2	14	9.86	9	2.88
B1	12	13.25	14	2.49
B2	15	10.67	11	2.19
Total	54	11.52	12	2.75

TABLE VI
ANALYSIS OF VARIANCE ON POST-TEST RESULTS FOR GROUPS A1, A2, B1, AND B2

Source	SS	df	MS	F	F'	%
between	103.41	3	34.47	5.78	2.76	25.76
within	298.07	50	5.96			74.24
Total	401.48	53				

However, are the differences in the means significant enough to be the effect of the use of a particular system version and its adaptation abilities? The analysis of variance (ANOVA) gives the answer. We use two-way analysis of variance, because there are two main effects in the model, the groups, and the use of adaptation techniques. Table IV summarizes the calculations and displays the two-way ANOVA results. The first column in the table denotes the source of variability (adaptation factor, group factor, adaptation-group interaction).

We see from Table IV that the critical F value (with 5% alpha level) for the adaptation factor is 3.92, which is much less than the calculated F value of 14.54. The null hypothesis that there is no significant main effect for the factor adaptation is thus rejected. In the second case, the critical F value of 3.07 is greater than the calculated value of 1.85 and the null hypothesis that there is no significant main effect for the factor group cannot be rejected in this case. Similarly, the interaction of both factors does not affect the data distribution.

Our goal was to estimate the usability of an adaptive system (i.e., to find out the differences between learning with adaptive system and learning with nonadaptive system, if any). Hence, the students from each of the groups A and B were divided according to the use of adaptive features of the system into two subgroups; we marked them A1 (the students that did use the adaptive features of the system), A2 (the students that did not use the adaptation), B1 (using adaptation), and B2 (not using adaptation). The results of the pretest and post-test were analyzed again according to the use of adaptive features of the system (considering the described four groups A1, A2, B1, and B2).

Again, the analysis of variance on pretest results showed no difference between the four groups. On the other hand, the analyses of variance on the post-test results revealed significant differences between the groups of users A1, A2, B1, and B2.

The differences are already seen from the means and medians of the four groups (see Table V): the means (and the medians) of groups A1 and B1 (the users that took advantage of the adaptive system features) are much higher than those of groups A2 and B2 (the users that used the system as a nonadaptive one); on average the difference is 2.7 points or 16% of the maximum possible score.

When performing analysis of variance on groups A1, A2, B1, and B2, we get the F value of 5.78, which is much greater than the critical F value of 2.76 (with $\alpha = 0.05$). Because the calculated value of F exceeds the critical value, the grouping variable does have an effect on the post-test result and we can reject the hypothesis of the equal means of the populations with the 0.05 probability of error. This means that there are significant differences between the four groups and that the use of a system version and its adaptive features does have a certain effect on the post-test results. The ANOVA results are summarized in Table VI.

The analysis of variance (F test) shows that overall the groups are related to the post-test results; thus, multiple comparison tests of significance have to be used to explore just which groups have the most to do with the relationship and estimate the size of the differences.

For post-hoc pair-wise comparison, which compares all possible pairs of group means, we used Fisher's least significant difference test (LSD) method [37] with individual 95% confidence limits (0.05 alpha level). As there are four groups, we get six comparisons of group means (group pairs). The calculated critical point is 2.0086. The estimated values of the comparisons are the differences of means of the two groups. Using the estimated values of the comparisons, the estimated standard error of the comparison, and the calculated critical point, we can calculate the confidence intervals for each pair of groups (lower and upper bounds of the confidence interval). The results of the multiple comparison tests are shown in Table VII.

When we test the hypothesis of equal group means (meaning the difference of two group means is zero), the confidence interval has to include our hypothetical point (zero) to confirm the hypothesis. Hence, the hypothesis of equal group means can be rejected for all those group pairs where the confidence interval does not cover the value zero. There are four intervals excluding zero, corresponding to the pairs of means A1–A2, A1–B2, A2–B1, and B1–B2. Differences in these group pairs can be identified as statistically significant. On the other hand,

TABLE VII
PAIR-WISE COMPARISON ON POST-TEST RESULTS FOR GROUPS A1, A2, B1, AND B2

Group pair	Value of comparison	Standard error	Lower bound	Upper bound	Interval excluding 0
A1 - A2	2.840	0.940	0.95	4.73	yes
A1 - B1	– 0.558	0.977	– 2.52	1.40	no
A1 - B2	2.030	0.925	0.17	3.89	yes
A2 - B1	– 3.390	0.961	– 5.32	– 1.46	yes
A2 - B2	– 0.810	0.907	– 2.63	1.01	no
B1 - B2	2.580	0.946	0.68	4.48	yes

TABLE VIII
QUESTIONNAIRE ON THE SYSTEM

Questions about the system	A	B	C	Ave
Needed less than 10 min to start using the system (%)	96	81	85	87
Enjoyed learning with the system (1 to 5)	4.3	3.8	3.5	3.9
Would continue using the system (%)	96	92	65	85
User interface, navigation and system easy to use (%)	100	99	100	99
Grade the system usability (1 to 5)	4.4	4.0	4.2	4.2
Grade the usability of tests (1 to 5)	4.7	4.5	4.7	4.6
Grade the color link annotation usability (1 to 5)	4.4	–	–	4.4
Grade the adaptive link insertion usability (1 to 5)	3.5	3.6	–	3.6
Took advantage of inserted links to related units (%)	58	54	–	56

there are no significant differences in group pairs A1–B1 and A2–B2.

The results of the experiment show that there is a significant difference in the post-test scores between the users of adaptive features of the system (groups A1 and B1) and all the others (groups A2 and B2). There is no bigger difference between the groups with or without the color annotation of links (groups A1 and A2 compared to B1 and B2).

B. General Usability of the System

In the experiment, beside the measurements of the test results, we were also interested in general opinion of the users on the system and its usability. These data were collected using a questionnaire, where the users gave their subjective opinion on the system they had used. Table VIII summarizes the answers to the main questions about the system for the three testing versions (fully adaptive version A, adaptive version without color annotation B, and nonadaptive version C). The left-hand column summarizes the questions and the next columns show the answers, which are expressed either in the percentage of students that agreed with the question, or as an average grade (from 1 to 5) given by the students. Where the question is not applicable to the system version, the answer is omitted.

As we can see from the user interface answers, the system and its navigation seemed easy enough to use. This is proved by the fact that more than half of the users (55%) needed less than 5 min to get used to the system [29], and the majority (87%) needed less than 10 min. Most users enjoyed learning with the system (average score 3.9, but the score is lower for nonadaptive version C) and would continue to use it in the future (85%). The percentage of users that would use the system in the future is much higher for versions A and B (96% and 92%) than for version C (65%), indicating the users preferred adaptive versions of the system.

The system's general usability grade (4.2 on average) shows that the prototype is well designed. The majority of users were excited about the opportunity to simultaneously check the learning progress with tests (in average 4.6 points out of 5). The usability of color annotation scored well (4.4), although the post-test results did not show any particular influence on the quality of learning. Links to related units scored relatively low (only 3.6), but this can be the result of the fact that they were used only by half of the users (56%).

VI. CONCLUSION

Educational hypermedia combines traditional hypermedia learning through an exploration approach with dynamic adaptation to individual user needs of intelligent tutoring systems. The adaptation is provided through the user model, which reflects individual user's features, with their knowledge among the most important ones.

This paper has described a way for dealing with uncertainty of user knowledge in educational systems. It is based on fuzzy set theory: fuzzy sets of domain concepts are used for describing user knowledge and linguistic rules to update the model. The model initialization rests on pretest results. During the user's work with the system, the model updates according to the unit visits, test results and propagation of knowledge.

The proposed model was implemented in an educational system, which was tested on a group of students. The results showed a positive influence of the adaptive system on user learning. The users that took advantage of the adaptive features of the system achieved on average a 26% higher score in the final examination. This means that adaptation helps to improve learning results. The color annotation did not show any significant effects on learning results, although the users liked it and found it quite useful.

Beside the examination results, the users' answers to questionnaire about the system were analyzed. The educational system was very well accepted by the students, was intuitive and easy to use, and the students all enjoyed this alternative way of learning. Thus, the adaptive system and its adaptation techniques provide an efficient, easy-to-use tool that supports users in learning and leads to better learning results.

REFERENCES

[1] R. Ubell, "Engineers turn to e-learning," *IEEE Spectrum*, vol. 37, pp. 59–63, Oct. 2000.
[2] P. Brusilovsky and M. T. Maybury, "From adaptive hypermedia to adaptive web," *Commun. ACM (Special Issue on the Adaptive Web)*, vol. 45, no. 5, pp. 31–33, 2002.
[3] P. De Bra, "Adaptive educational hypermedia on the web," *Commun. ACM (Special Issue on the Adaptive Web)*, vol. 45, no. 5, pp. 60–61, 2002.
[4] R. M. Gagné, *Principles of Instructional Design*, 3rd ed. New York: Holt, Rinehart and Winston, 1988.
[5] P. Brusilovsky, "Adaptive hypermedia: an attempt to analyze and generalize," in *Multimedia, Hypermedia, and Virtual Reality*. Berlin, Germany: Springer-Verlag, 1996, pp. 288–304.
[6] ——, "Adaptive hypermedia," *User Modeling User Adapted Interaction (Tenth Year Anniversary Issue)*, vol. 11, no. 1/2, pp. 87–110, 2001.

[7] ——, "Methods and techniques of adaptive hypermedia," in *Adaptive Hypertext and Hypermedia*. Norwell, MA: Kluwer, 1998, pp. 1–43.
[8] J. Eklund and P. Brusilovsky, "Individualising interaction in Web-based instructional systems in higher education," in *Proc. AUC Academic Conf.,*. Melbourne, Australia, 1998, pp. 27–30.
[9] E. Wenger, *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. San Mateo, CA: Morgan Kaufmann, 1987.
[10] A. Jameson. User-adaptive systems: an integrative overview. presented at Proc. 7th Int. Conf. User Modeling. [Online] Available: http://w5.cs.uni-sb.de/~jameson/um99-tutorial-index.html.
[11] E. M. Valldeperas, "Bayasian systems for student modeling," Ph.D. dissertation, Univ. Málaga, Málaga, Spain, 2000.
[12] E. H. Shortliffe, *Computer-Based Medical Consultations: MYCIN*. New York: Elsevier/North Holland, 1976.
[13] R. O. Duda, J. Gaschnig, and P. E. Hart, "Model design in the Prospector consultant system for mineral exploration," in *Expert Systems in the Micro Electronic Age*. Edinburgh, U.K.: Edinburgh Univ. Press, 1979, pp. 153–167.
[14] G. Weber and P. Brusilovsky, "ELM-ART: an adaptive versatile system for web-based instruction," *Int. J. Artif. Intell. Educ.*, vol. 12, pp. 351–384, 2001.
[15] D. N. Chin, "KNOME: modeling what the user knows in UC," in *User Models in Dialog Systems*. New York: Springer-Verlag, 1989, pp. 74–107.
[16] "Proc. AAAI Spring Symp. Representing Mental States and Mechanisms," AAAI Tech. Rep., SRI Int., Stanford, CA, 1995.
[17] S. Katz, A. Lesgold, G. Eggan, and M. Gordin, "Modeling the student in Sherlock II," *J. Artif. Intell. Educ.*, vol. 3, pp. 495–518, 1992.
[18] N. Capuano, M. Marsella, and S. Salerno, "ABITS: an agent based intelligent tutoring system for distance learning," in *Proc. Int. Workshop on Adaptive and Intelligent Web-Based Educational Systems Held in Conjunction With ITS 2000 Montreal, Canada*. Osnabrück, Germany, 2000, pp. 17–28.
[19] D. Mullier, "The application of neural network and fuzzy logic techniques to educational hypermedia," Ph.D. dissertation, School of Comput., Leeds Metropolitan Univ., Leeds, U.K., 1999.
[20] J. Martin and K. VanLehn, "Student assessment using Bayesian nets," *Int. J. Human Comput. Stud.*, vol. 42, no. 6, pp. 575–591, 1995.
[21] C. Conati and K. VanLehn, "POLA: a student modeling framework for probabilistic on-line assessment of problem solving performance," in *Proc. 5th Int. Conf. User Modeling*, Kailua-Kona, HI, 1996, pp. 75–82.
[22] A. S. Gertner, C. Conati, and K. VanLehn, "Procedural help in Andes: generating hints using Bayesian network student model," in *Proc. 15th Nat. Conf. Artificial Intelligence*. MIT Press: Cambridge, MA, 1998, pp. 106–111.
[23] R. J. Mislevy and D. H. Gitomer, "The role of probability-based inference in an intelligent tutoring system," *User Modeling User-Adapted Interaction*, vol. 5, no. 3/4, pp. 253–282, 1995.
[24] L. Kettel, J. Thomson, and J. Greer, "Generating individualized hypermedia applications," in *Proc. Int. Workshop on Adaptive and Intelligent Web-Based Educational Systems Held in Conjunction With ITS 2000 Montreal, Canada*. Osnabrück, Germany, 2000, pp. 37–49.
[25] N. Henze, "Adaptive Hyperbooks: Adaptation for project-based learning resources," Ph.D. dissertation, Fac. Math. Comput. Sci., Univ. Hannover, Hannover, Germany, 2000.
[26] M. A. Bauer, "A Dempster-Shafer approach to modeling user preferences for plan recognition," *User Model. User-Adapted Interact.*, vol. 5, no. 3/4, pp. 317–348, 1995.
[27] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[28] L. A. Zadeh, "Fuzzy sets," *Inf. Control.*, vol. 8, pp. 338–353, 1965.
[29] A. Kavčič, "Adaptation in web-based educational hypermedia with regard to the uncertainty of user knowledge," Ph.D. dissertation, Faculty Comput. Inform. Sci., Univ. Ljubljana, Ljubljana, Slovenia, 2001.
[30] Sun Microsystems, Inc.. The Source for Developers, A Sun Developer Network Site. J2EE, Java Servlet Technology. [Online]http://java.sun.com/products/servlet/
[31] P. A. M. Kommers, S. Grabinger, and J. C. Dunlap, *Hypermedia Learning Environments: Instructional Design and Integration*. Mahwah, NJ: Lawrence Erlbaum, 1996.
[32] E. Herder, "Utility-based evaluation of adaptive systems," in *Proc. 2nd Workshop Empirical Evaluation of Adaptive Systems at the 9th Int. Conf. User Modeling*, Johnstown, PA, 2003, pp. 25–30.
[33] M. Marolt, M. Privošnik, and A. Kavčič, *Introduction to Java*. Ljubljana, Slovenia: Faculty Comput. Inform. Sci., Univ. Ljubljana, 2000.
[34] M. Campione, K. Walrath, and A. Huml, *The Java Tutorial: A Short Course on the Basics*, 3rd ed. Reading, MA: Addison-Wesley, 2000.
[35] M. D. Mann, "Using the adaptive navigation support technique of link hiding in an educational hypermedia system: an experimental study," Ph.D. dissertation, Fac. Graduate College, Oklahoma State Univ., Stillwater, 1999.
[36] A. Mitrovic and K. Hausler, "Porting SQL-tutor to the web," in *Proc. Int. Workshop Adaptive and Intelligent Web-Based Educational Systems Held in Conjunction With ITS 2000 Montreal, Canada*. Osnabrück, Germany, 2000, pp. 50–60.
[37] W. L. Hays, *Statistics*, 5th ed. Orlando, FL: Harcourt Brace Jovanovich, 1994.

Alenka Kavčič (M'98) received the Ph.D. degree in computer science from the University of Ljubljana, Ljubljana, Slovenia, in 2001.

She is an Assistant in the Faculty of Computer and Information Science, University of Ljubljana. In addition to her pedagogical responsibilities, she is involved in a number of research projects related to multimedia and internet technologies, computer-based education and learning, human–computer interaction, grid computing, and virtual and augmented reality. Her main research interest is in the field of hypermedia and computer based learning, especially adaptive hypermedia systems and user modeling.

Dr. Kavčič is a member of the Association for Computing Machinery (ACM), Slovenia.