1

User Models for Adaptive Hypermedia and Adaptive Educational Systems

Peter Brusilovsky[1] and Eva Millán[2]

[1]School of Information Sciences
University of Pittsburgh, Pittsburgh PA 15260, USA
[2] ETSI Informática
University of Malaga
peterb@pitt.edu, eva@lcc.uma.es

Abstract. One distinctive feature of any adaptive system is the user model that represents essential information about each user. This chapter complements other chapters of this book in reviewing user models and user modeling approaches applied in adaptive Web systems. The presentation is structured along three dimensions: what is being modeled, how it is modeled, and how the models are maintained. After a broad overview of the nature of the information presented in these various user models, the chapter focuses on two groups of approaches to user model representation and maintenance: the overlay approach to user model representation and the uncertainty-based approach to user modeling.

1.1 Introduction

Adaptive hypermedia and other adaptive Web systems (AWS) belong to the class of *user-adaptive software systems* [174]. One distinctive feature of an adaptive system is a *user model*. The user model is a representation of information about an individual user that is essential for an adaptive system to provide *the adaptation effect*, i.e., to behave differently for different users. For example, when the user searches for relevant information, the system can adaptively select and prioritize the most relevant items (see Chapter 6 of this book [125]). When the user navigates from one item to another, the system can manipulate the links (e.g., hide, sort, annotate) to provide adaptive navigation support (see Chapter 8 of this book [21]). When the user reaches a particular page, the system can present the content adaptively (see Chapter 13 of this book [28]). To create and maintain an up-to-date user model, an adaptive system collects data for the user model from various sources that may include implicitly observing user interaction and explicitly requesting direct input from the user. This process is known as *user modeling*. User modeling and adaptation are two sides of the same coin. The amount and the nature of the information represented in the user model depend to a large extent on the kind of adaptation effect that the system has to deliver.

As mentioned in the introduction, Chapters 1 to 5 of this book are focused mostly on the modeling side of personalization, while the remaining chapters focus mostly on the adaptation side. Chapters 1 and 2 are specifically devoted to user models and user modeling. Beyond this, user modeling issues are discussed at different levels of detail in several other chapters. This chapter attempts to complement the remaining chapters in two ways. First, it provides an overview (a "big picture") of the user modeling side referring readers when necessary to additional information in chapters within this book. Second, it attempts to complement other chapters by presenting aspects that are either not covered in other chapters or covered insufficiently.

To envision the big picture, this chapter follows Sleeman [175] who suggested classifying user models by the nature and form of information contained in the model as well as the methods of working with it. Following his suggestions, we analyze user models along three layers: *what is being modeled* (nature), *how this information is represented* (structure) and *how different kinds of models are maintained* (user modeling approaches). In this book, the overview of user modeling along the first layer, the nature of the represented information, is provided in section 1.2 of this chapter. This section serves as a basis for understanding the user modeling problem as a whole (Fig. 1.1).

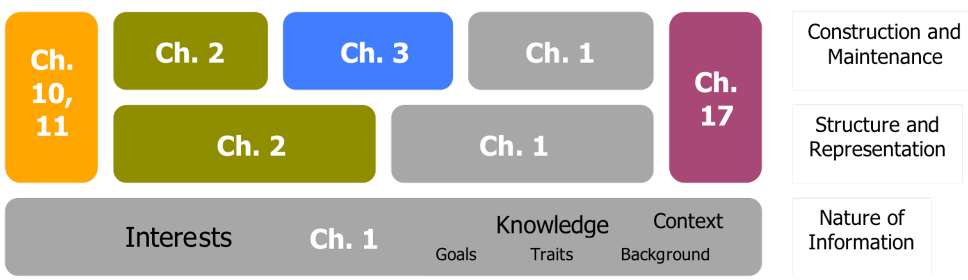


Fig. 1.1. Three layers for the analysis of user modeling approaches and their coverage in different chapters of this book. Horizontal dimension represents user features reflected in the models.

The review of the structure and the representation of information in user models (the second layer) is split between this chapter and Chapter 2 [72]. Together these chapters provide a detailed overview of the two most important and most elaborate types of user models, which were originally developed in the fields of information retrieval [107] and intelligent tutoring systems [158]. Information retrieval and filtering systems attempt to find documents that are most relevant to user interests and then to order them by perceived relevance. The user model that typically powers this kind of systems is known historically as a *user profile* and represents the user's *interests* in terms of keywords or concepts. Intelligent tutoring systems (ITS), strive to select educational activities and deliver individual feedback that is most relevant to the user's level of knowledge. The user model in ITS is known as a *student model* and represents mostly the user's *knowledge* of the subject in relation to expert-level domain knowledge. In this book, section 2.3 in Chapter 2 reviews three main types of user profiles for representing user interests, while section 1.3 (in this chapter) reviews the dominant overlay approach for modeling user knowledge.

Finally, the process of construction and maintenance of user models (the third layer) is discussed in several chapters. The lion's share of this presentation is provided in Chapters 1 and 2 - the main user modeling chapters. These chapters follow their foci on knowledge and interest modeling respectively: section 1.4 of Chapter 1 focuses on Bayesian Networks, the most important approach to knowledge modeling, while section 2.4 of Chapter 2 reviews major interest-modeling approaches. This presentation is complemented by several other chapters, which focus on specific user modeling approaches. Chapter 3 [135] reviews Data Mining approach to user profile constructions. Chapters 9 and 10 [152; 172] review content-based and collaborative user profiling as used for recommendation. Chapter 17 [109] elaborates on context modeling for mobile applications.

The design of this chapter is the result of a compromise between two goals: to provide core content that is determined by the chapter's share in presenting the "big picture" of user modeling while making it useful and meaningful on its own. To achieve the second goal, the chapter was focused on user modeling for Adaptive Hypermedia (AH) and Adaptive Educational Systems (AES). AH and AES are two groups of Web-based systems that extensively employ the user knowledge models covered by the core of this chapter. To better justify this title, the knowledge modeling core of sections 1.3 and 1.4 of this chapter were extended to cover overlay modeling and Bayesian user modeling, which goes "beyond knowledge". The nature of this chapter caused a different balance between the breadth and depth of coverage for the different sections. Section 1.2 attempts to provide a broad coverage while providing relatively few details about specific user models or modeling approaches. It is intended as a good overview of the topic that will be useful for anyone interested in adaptive Web systems. In contrast, sections 1.3 and 1.4 provide more details at the price of a broader coverage. While focusing mainly on the modeling of user knowledge, they also introduce approaches to the modeling of other kinds of information about the user.

1.2 What Is Being Modeled

According to the nature of the information that is being modeled in adaptive Web systems, we can distinguish models that represent features of the user as an individual from models that represent the current context of the user's work. The former are important to all adaptive Web systems while the latter are mostly the concern of mobile and ubiquitous adaptive systems, where context is essential. This section focuses on the five most popular and useful features found when viewing the user as an individual: the user's knowledge, interests, goals, background, and individual traits. We also discuss modeling the context of a user's work. At the end of this section we discuss stereotype-based user modeling that is an alternative to the more popular feature-based modeling.

1.2.1 Knowledge

The *user's knowledge* of the subject being taught or the domain represented in hyperspace appears to be the most important user feature, for existing AES and AHS. In AES, the knowledge is frequently the only user feature being modeled. In AHS, it is used by the majority of systems for both adaptive navigation support and adaptive presentation. The user's knowledge is a changeable feature. The user's knowledge can both increase (learning) or decrease (forgetting) from session to session and even within the same session. This means that an adaptive system relying on user knowledge has to recognize the changes in the user's knowledge state and update the user model accordingly.

The simplest form of a user knowledge model is the scalar model, which estimates the level of user domain knowledge by a single value on some scale – quantitative (for example, a number ranging from 0 to 5) or qualitative (for example, good, average, poor, none). Scalar models, especially qualitative, are quite similar to stereotype models. The difference is that scalar knowledge models focus exclusively on user knowledge and are typically produced by user self-evaluation or objective testing, not by a stereotype-based modeling mechanism. Despite their simplicity, scalar models can be used effectively to support simple adaptation techniques in AHS. A number of AHS's use scalar knowledge models to support adaptive presentation. These systems divide their users into two or three classes according to their knowledge level of the subject (i.e., expert, intermediate, and novice) and serve different versions of the whole page content [63] or page fragments [8; 14; 16] to users with different levels of knowledge.

A good example of adaptive presentation based on a scalar model is the MetaDoc system [14], MetaDoc represents user knowledge of UNIX (which was the domain of the system) as a qualitative scalar value (novice - beginner - intermediate – expert. The scalar model was used to generate an original adaptation based on *stretchtext*. Stretchtext is a special kind of hypertext where clicking on an anchor (hotword) simply "expands" it by inserting a fragment of content *after* or *instead of* the anchor. Another click collapses the expanded content back to the original hotword (a similar approach is used by the well-known Windows Explorer). Each page in MetaDoc is a stretchtext, which may contain many expandable hotwords. The idea of adaptive stretchtext presentation in MetaDoc is to present a requested page with all stretchtext fragments not relevant to the user being collapsed and all fragments relevant to the user being expanded. To achieve this result, an author must first classify expandable text fragments as either an additional explanation or a low-level detail. The user of MetaDoc with an expert level of knowledge of a concept will be presented with additional explanations hidden (collapsed) and low-level details expanded. On the other hand, the user with a beginner's level of knowledge will receive expanded additional explanations in all cases. After its presentation, a stretchtext page can be further adapted by the user who is free to expand and collapse text fragments. The study of MetaDoc demonstrated that this simple technology based on a scalar model can increase the speed and quality of user comprehension of the content [14]. More information about stretchtext and other adaptive presentation techniques can be found in Chapter 13 of this book [28].

The shortcoming of the scalar model is its low precision. User knowledge of any reasonably-sized domain can be quite different for different parts of the domain. For example, in word processing, a user may be an expert in using text annotation, but a novice in formula editing [66]. A scalar model effectively averages the user knowledge of the domain. For any advanced adaptation technique that has to take into account some aspect of user knowledge, the scalar model is not sufficient. For the above reason, AES's that focus on advancing user knowledge and many AHS's use various kinds of *structural models*. The structural models assume that the body of domain knowledge can be divided into certain independent fragments. These models attempt to represent user knowledge of different fragments independently. By the nature of represented knowledge, structural models can be independently classified along two different sub-dimensions, according to:

- the type of represented knowledge (declarative vs. procedural), and
- a comparison of the user's knowledge—represented in the model—to an expert's level of knowledge of the subject, referred to as domain model, expert model, or "ideal student" model.

The most popular form of a structural knowledge model is an *overlay model*. The purpose of the overlay model is to represent an individual user's knowledge as a subset of the *domain model*, which reflects the expert-level knowledge of the subject. For each fragment of domain knowledge, an overlay model stores some estimation of the user's knowledge level of this fragment. The pure overlay model, developed in the field of ITS over 30 years ago [194], assigns a Boolean value, yes or no, to each fragment, indicating whether the user knows or does not know this fragment. In this case, user knowledge is represented at each instant of time as an exact subset or "overlay" of expert knowledge. In its modern form, an overlay model represents the degree to which the user knows such a domain fragment. This can be a qualitative measure (good-average-poor), or a quantitative measure, such as the probability that the user knows the concept.

Since the overlay model represents the user's knowledge as a (weighted) subset of expert knowledge, the nature of the user knowledge reflected in the overlay model depends on the nature of the expert knowledge represented in a specific system. The majority of ITS focused on representing two types of domain knowledge: conceptual knowledge (facts and their relationships) and procedural knowledge (problem-solving skills). Conceptual knowledge is typically represented in the form of a network of concepts. Procedural knowledge is most frequently represented as a set of problem-solving rules. Several knowledge-representation approaches, such as ACT-R [2], or propositional representation [100] allowed one to combine these two types together. More recently, Stellan Ohlsson suggested focusing on a different kind of procedural knowledge – not the knowledge that allows the user to solve the problem, but the knowledge that allows him/her to evaluate the correctness of the solution [144]. This knowledge is typically represented as a set of constraints.

In turn, the nature of knowledge represented in a specific adaptive system is determined by the kind of personalized support it provides. A large class of ITS known as "tutors" focus on helping users solve educational problems and thus rely on the procedural knowledge of either problem solving or evaluation nature. Other types of ITS and IES focus on helping users to select the most relevant piece of educational content and thus rely on conceptual knowledge about the domain. The use of conceptual knowledge is shared by almost all non-educational AHS, which also focus on guiding the user to the most appropriate content. In all these cases, regardless of the nature of the represented expert knowledge, the overlay model can be successfully applied to model individual user knowledge, i.e., it can measure how well the user knows a concept, what is the probability that the user can apply a rule or which of the constraints or propositions are likely mastered.

Overlay models constitute a dramatic step forward from scalar models. Yet, in the field of ITS, overlay models have often been criticized for being "too simple." It has been argued that the state of user knowledge is never an exact subset of expert knowledge. The user may have misconceptions and her knowledge generally progresses to expert-level knowledge not by "filling the gaps," but through a complex process of generalization and refinement. To model user misconceptions, an overlay model was expanded into a *bug model,* representing both correct knowledge and misconceptions (known as buggy knowledge or "bugs"). Bug models were predominantly used to model user procedural problem solving knowledge. The most extensively studied form of bug model is called the *perturbation model*. This model assumes that several incorrect *perturbations* can exist for each element of domain knowledge. Incorrect user behavior may, from the viewpoint of this approach, be caused by the systematic application of one of the perturbations in place of the correct rule. The goal of a system with a bug model is not just to declare that a specific element of domain knowledge is incomplete or missing, but to identify, if possible, specific buggy knowledge that can be used to provide a higher quality adaptation. An even richer model that makes it possible to reflect the development (genesis) of user knowledge from the simple to the complex and from the specific to the general is known as a *genetic model* [74].

While both bug models and genetic models are certainly more powerful than the traditional overlay model, they are also much harder to develop. Research on these models has contributed to the development of the fields of cognitive modeling and ITS [194], but the practical use of these models has been quite limited. Genetic models have never been used in practical systems. Bug models have been used mostly in problem solving ITS created for simple domains, although several well-known systems created by Carnegie Mellon researchers have demonstrated that this approach could work for large-scale practical systems [1; 106]. In the area of Web-based systems the use of bug models is limited to a small subset of Web-based AES that are focused on adaptive problem solving support. Non-educational Web systems do not use bug models since they have no means (and no need) to diagnose misconceptions.

The typical pattern of bug model application in Web-based AES can be demonstrated by such systems as WITS [145], ILESA [116], and Web-PVT [192]. These systems, developed for three different educational domains, provide a personalized analysis of exercise solutions and some form of guidance through problem selection (WITS), problem generation (ILESA) or adaptive navigation support (Web-PVT). The systems use a combination of an overlay model and a bug model. Bug models allow the systems to recognize misconceptions in the user's problem-solving knowledge, distinguish it from random slips, such a typos and calculation errors, and provide a useful personalized explanation. However, the rest of the adaptive functionality (such as problem generation or adaptive navigation support) only considers the balance between correct and incorrect usage of knowledge, as reflected in the overlay model, ignoring the exact nature of the misconceptions represented in the bug model.

In contrast to the powerful but complicated and rarely used bug models, overlay knowledge models are extremely popular in both Web AES and AHS. Almost every Web AES and a majority of modern AHS are based on some form of overlay models. Multiple projects have demonstrated that an overlay model provides a good balance of simplicity and power. The ability to independently measure user knowledge within different elements of the domain provides a level of power that is sufficient to run the majority of the advanced adaptation techniques. Yet, overlay model are relatively easy to develop, especially for cases with less than 100 domain-knowledge elements. Due to the importance of overlay models for AES and AHS, this chapter provides a special section on the development of overlay models.

1.2.2 Interests

User interests always constituted the most important (and typically the only) part of the user profile in adaptive information retrieval and filtering systems that dealt with large volumes of information. It is also the focus of user models/profiles in Web recommender systems. In contrast, early AES and educational AHS paid no attention to user interests, instead focusing on learning goals when sequencing educational content. As for non-educational AHS, their nature and the small size of their hyperspaces created no demand for interest adaptation. This situation has changed dramatically over the last 10 years. User interests are now competing with user knowledge to become the most important user feature to be modeled in AHS. The change was caused by the rapid growth of the volume of information and the growing popularity of several new kinds of information-oriented AHS such as encyclopedias [84], hypertextual news systems [3], electronic stores [4], museum guides [142], and information kiosks [65] where access to information is mostly interest-driven. Following the pioneer attempts mentioned above, interest modeling was explored in a number of information-oriented AHS. More recently, the abundance of available content and the growing popularity of the interest-driven constructionist approach to education have encouraged more attempts to model user interests in educational AHS.

Starting from the pioneer systems, AHS focused on a new approach for modeling user interests. The predominant representation of user interests at that time was the weighed vector of keywords. This approach was used by nearly every adaptive information retrieval and filtering system and is still the most popular in these areas. More details about this approach can be found in Chapter 2 [72]. In contrast to this keyword-level approach, AHS adopted a concept-level approach to user interest modeling where user interests are represented as a weighed overlay of a concept-level domain model. The concept overlay approach to user interest modeling is very similar to the overlay knowledge modeling approach. Overlay user modeling and specific examples of its application to user interest modeling are presented in more detail in section 1.3 of this chapter.

Concept-level models of user interests are generally more powerful than keyword-level models. Concept-level models allow a more accurate representation of interests. Given a rich domain model, a concept overlay model can separately model different aspects of user interests. For example, a news personalization system can model user interests on distinct topics, based in a specific geographical location, and dealing with specific named entities [96]. An adaptive museum system can separately model interests in the designer, style, or origin of a jewelry item [143]. In addition, semantic links in the domain model allow different kinds of interest propagation to compensate for *sparsity*, a standard problem of large overlay models.

Powerful, concept-level models of user interests in AHS have been enabled by the nature of AHS content that is traditionally manually indexed with domain model concepts. In closed corpus AHS, such as adaptive information kiosks or museum guides, the content was indexed at the time of system creation. In systems with expandable corpus, such as adaptive news systems, new content had to be indexed by a provider at the time of its insertion into the system [3; 96]. The need for manual indexing originally led to the establishment of two distinctive groups or systems that are able to adapt to user interests. Closed corpus systems (mostly AHS) used the concept-level interest model, paying the price of manual indexing. Open corpus systems (mostly information retrieval and filtering systems) used keyword-level models, but were able to work with an unrestricted corpus of document due to their ability to process documents automatically. This dichotomy is no longer clear-cut. A new generation of adaptive information access systems has attempted to combine concept-level interest modeling with automatic document processing. Many of these systems are based on automatic document categorization where each document is automatically assigned to a concept of an existing domain model (such as the Yahoo! directory or the ACM topic ontology). These approaches are presented in detail in Chapter 2 [72] of this book. More recent systems explore automatic multi-concept indexing, where a document can be automatically connected with several domain model concepts [43]. In addition, a new group of hybrid systems attempts to combine concept-level and keyword-level interest models in one system [54]. A more elaborate discussion on bridging the gap between closed corpus AHS and open corpus information retrieval and filtering systems is provided in Chapter 22 of this book [25].

1.2.3 Goals and Tasks

The user's goal or *task* represents the immediate purpose for a user's work within an adaptive system. Depending on the kind of system, it can be the goal of the work (in application systems), an immediate information need (in information access systems), or a learning goal (in educational systems). In all of these cases, the goal is an answer to the question "What does the user actually want to achieve?" The user's goal is the most changeable user feature: it almost always changes from session to session and can often change several times within one session of work. Early research on adaptation to the user's *work goal* was done in the area of adaptive interfaces and intelligent help systems [10; 66; 76]. Adaptation to the user's *learning goal* was explored by instructional planning and sequencing systems [18; 123; 124; 197]. Adaptation to the user's immediate *information need* was explored by adaptive information retrieval systems [17]. Among AHS systems goal modeling techniques were explored in the sub-areas of hypertext-based help systems [63; 78] and adaptive information access systems [88; 97; 121] where the ability to adapt to the current user goal is very important.

Goal modeling in modern AHS and AES mostly follows the approaches suggested in the pioneer research mentioned above. The user's current goal is usually modeled with a *goal catalog* approach, which is somewhat similar to overlay knowledge modeling. The core of this approach is a pre-defined catalogue of possible user goals or tasks that the system can recognize. Frequently this catalogue is simply a small set of independent goals, however, some systems use a more advanced catalog in the form of a goal or task hierarchy, which is inherited from earlier research on adaptive interfaces and instructional planning. In a goal hierarchy, relatively stable higher-level goals are progressively decomposed into subgoals down to the lowest level formed by short-term goals. Most typically, the system assumes that the user has exactly one goal (or one goal at each level of the hierarchy) at any moment of work with the system. The job of the user modeling component is to recognize this goal and to mark it as the current goal in the model. This fires the adaptation rules that refer to possible user goals specified in the catalogue. The adaptation rules can, for example, recommend some pages to the user [23], focus user attention on a subset of the hyperspace [69; 198], or adapt content of the selected page [88].

The goal/task recognition process is difficult and not precise in general. It is especially difficult in AH and other Web-based system where the flow of information from the user (bandwidth) required by user modeling components is thinner than in traditional desktop systems. Over the years, adaptive Web systems explored a number of approaches to fight this problem. To start with, a number of practical systems allow the user to specify the current goal. Typically the user has to select one of the pre-defined goals [69; 88; 151], although some systems are able to gradually learn how to adapt to a completely new goal introduced by the user [97; 121]. A different approach to fight the imprecise goal recognition is to model the user current goal as a probabilistic overlay of the goal catalogue where for each goal the system maintains the probability that this goal is the current goal of the user [63; 126]. Finally, several recent projects explored the use of data mining technologies to identify the current user task in an expected sequence of tasks and to provide personalized task-level support [87; 94]. More information about the use of data mining technologies for Web personalization ns provided is Chapter 3 of this book [135].

A popular example of goal adaptation is provided by the PUSH system [88]. This system has a small catalogue of user goals and adapts the presentation of each selected page to the current goal. Depending on the current goal, some parts of the page can be collapsed using the stretchtext approach explained in subsection 1.2.1 above. The system attempts to deduce the user goal by observing the user's actions and shows the assumed goal to the user in the spirit of a *glass-box adaptation*. The user can also change the current goal by selecting a more appropriate one from the catalog.

A different example of goal modeling is a performance support system ADAPTS [23]. ADAPTS deduces the current goal within a goal hierarchy by following the user's aircraft maintenance operations. Once the goal is recognized, the system generates a page with the list of technical manual fragments that are most relevant to the current goal and user level of knowledge.

1.2.4 Background

The *user's background* is a common name for a set of features related to the user's previous experience outside the *core domain* of a specific Web system (for example, the core domain of a city guide [36] is a specific city and its objects of interest; the core domain for a hospital information system [198] is a specific hospital, its objects and procedures). A range of backgrounds that have been used in adaptive Web systems includes the user's profession, job responsibilities, experience of work in related areas, and even specific view on the domain. For example, medical adaptive hypermedia systems can distinguish two or three categories of users according to their knowledge of medical terminology and adapt content presentation to the user category by selecting either medical terms or everyday language to present the same content [8; 178]. Alternatively these systems can distinguish users by their profession (student, nurse, doctor) which implies both the level of knowledge and responsibilities [53; 198]. More examples for this application area can be found in Chapter 15 of this book [34]. Another example is the categorizing of users by their language ability (i.e., native or non-native speakers), followed by choosing the appropriate version of the content for them [102]. Background information is used most frequently for content adaptation, although there are examples of the use of it within adaptive search [71; 121] and adaptive navigation support [198].

By its nature, user background is similar to the user's knowledge of the subject (i.e., it is also mostly a measure of knowledge beyond the core domain area). However, the representation and handling of user background in adaptive systems is different. Since detailed information about the background is not necessary, the common way to model user background is not an overlay, but a simple stereotype model. In addition, user background typically does not change during work with the system and is nearly impossible to deduce by simply watching the user work. As a result, user background is typically provided explicitly, either by the user herself or by some kind of a superior (a teacher in a college or an administrator at an institution).

1.2.5 Individual Traits

The *user's individual traits* is the aggregate name for user features that together define a user as an individual. Examples are personality traits (e.g., introvert/extravert), cognitive styles (holist/serialist), cognitive factors (e.g., working memory capacity) and learning styles. Similar to user background, individual traits are stable features of a user that either cannot be changed at all, or can be changed only over a long period of time. Unlike user background, however, individual traits are traditionally extracted not from a simple interview, but through specially-designed psychological tests. Many researchers agree on the importance of modeling individual traits and using them for adaptation. While different kinds of user traits are extensively discussed in psychological literature, current work on modeling and using individual traits for personalization focuses mostly on two groups of traits – *cognitive styles* and *leaning styles*. These groups are discussed below. More recently, researchers on adaptive systems started considering individual traits beyond cognitive and learning styles. For example, CUMAPH system [188] made a pioneer attempt to build a user profile from lover level cognitive abilities and apply for adaptively generating page content for the user. Another recent work considered the use of personality factors in the context of adaptive museum guides [77] (more information about adaptive museum guides is presented in Chapter 17 of this book [109]).

Cognitive Styles. By cognitive style, researchers typically mean an individually preferred and habitual approach to organizing and representing information [165]. Research on cognitive style has long attracted attention on researchers in Web personalization and related fields such as human information behavior. Professional literature distinguishes a number of dimensions in which the users cognitive styles may differ: field-dependent/independent, impulsive/reflective, conceptual/inferential, thematic/relational, analytic/global [35; 113]. Most popular among Adaptive Hypermedia researchers are Witkin's field-dependent/independent [205] and Pask's holist/serialist [150]. Since, by its nature, cognitive style influences humans' ability to access information, the work on adaptation to user cognitive style was focused more on the navigational side of AHS and AES. For example, several adaptive hypermedia systems [61; 133; 190; 191] distinguished field-dependent and field-independent users and provided different navigation organization, amount of user control, and navigation support tools for these groups.

A typical scenario for cognitive style adaptation is provided in AES-CS system [190; 191]. AES-CS attempted to adapt to both user knowledge and user cognitive style. A field-dependency test was used to classify users in two groups: field-dependent and field-independent. After that a range of system features were adapted to the identified cognitive style. Field-independent users received an access to the navigation menu to control their navigation. Field-dependent users were only able to proceed through the content sequentially, however they were provided with additional orientation support tools such as a concept map and a path indicator. Depending on their learning style, the users also received different instructions, feedback, and contextual organizers. Evaluated against a static version of content, the AES-CS (with both kinds of adaptation enabled) demonstrated a significant increase of user performance [191].

So far, the research on the use of cognitive style of adaptation is a mixed-success story. On one side, a number of studies confirmed that cognitive style affects both search and browsing behavior [35; 104]. On the other side, few success stories (like AES-CS) on using cognitive styles for adaptation were reported. A number of projects made a similar attempt to distinguish users by their cognitive style and match them with a version of a hypermedia system developed to support this style, but were not able to report any significant differences against a non-adaptive condition. It is interesting that attempts to mismatch user styles and system versions (i.e., to match users to version of the system developed for the alternative style) typically reported significant negative results [133]. Thus, cognitive style remains an important user feature to take into account, but reliable approaches of adaptation to user cognitive styles are yet to be found.

Learning Styles. Learning styles are typically defined as the way people prefer to learn. This group of individual traits is close to cognitive style, but more narrow in scope due to its focus on human learning. The application of learning styles to adaptation is limited to Web-based AES. After a few pioneer works that presented the idea of adapting Web-based AES to learning styles and suggested some ways to implement it [33], this direction of work quickly emerged into arguably the most popular kind of research on individual traits adaptation on the Web. A number of pre-Web approaches or *inventories* to classify and measure learning styles were applied and a number of studies demonstrated differences between users with different styles in Web context. Most of the work on learning style adaptation explored *content-level* adaptation attempting to match users with a specific learning style to content that should be the most appropriate for this style. This adaptation may take different forms, such as selecting the most style-relevant version of content for presentation, ordering content fragments by their relevance to user style or hiding style-irrelevant content. More recently, these popular approaches to learning style adaptation were integrated into several adaptation frameworks and authoring tools such as ACE [179], CAMELEON [111], AHA! [181], and APeLS [47]. For example, a system developed according to APeLS framework [47] matches the user model with content metadata in order to select learning objects that are most relevant to the user's learning style given certain alternatives provided in the pool of resources.

Despite all this work, there are no proven recipes for the application of learning styles in adaptation. It is still unclear which aspects of learning style are worth modeling, and what can be done differently for users with different styles. Dozens of experimental systems that consider different style inventories and suggest different ideas for adaptation were reported, but careful studies are rare and success stories are very few. On the contrary, a number of experimental studies aimed to evaluate the value of treating users with different learning styles differently concluded without finding any significant differences. As a whole, the situation is similar to the situation with cognitive styles: the area of adaptation to individual traits holds a lot of potential, but offer almost no practical suggestions. To progress in this area, we either need to learn more about the relationships between user traits and possible interface settings, or develop trait-agnostic techniques that treat user traits as a black box and attempt adapt to them using case-based and non-symbolic technologies [73].

1.2.6 Context of Work

Adaptation to the context of the user's work is a relatively new research direction within AHS. It was introduced by several pioneer Web-based systems and later expanded into the area of mobile adaptive systems. Early context-adaptive systems explored mostly platform adaptation issues. The growing interest to mobile and ubiquitous systems attracted researchers attention to other dimensions of the context such as user location, physical environment, social context, and affective state. Context modeling is conceptually different from modeling of other user features discussed above. Some information represented in the context models can hardly be considered information about the user in pure sense. However, context modeling and user modeling are tightly interconnected. Many user models include context features; similar techniques are used for context and user modeling; integrated frameworks are being developed for modeling both user context and user features [80; 210]. For all that reason, the overview of context modeling is included in this chapter.

User Platform. Since users of the same server-side Web application may use different equipment at different times, adaptation to the user's platform becomes an important issue. Adaptive hypermedia systems explored a range of techniques that might be used to adapt to such aspects of user platform (computing environment) as hardware, software, and network bandwidth. The largest stream of work focused on adaptation to the screen size by either converting pages designed for viewing on desktop Web browsers to mobile browsers or generating pages differently for desktop and mobile applications. Another stream focused on media presentation capabilities that are a combination of hardware, available software, and bandwidth. Over the last several years, the work on platform adaptation has grown from a pioneer research domain into practice and is now in the focus of W3C [202]. Most recent attempt to standardize the description and use of platform capabilities can be found in [105].

It is important to stress that platform-oriented context models are different from the knowledge, task, and goal models reviewed above. A context is typically described by a potentially long set of name-value pairs where names indicate parameters (i. e, screen width, type of pointing device, or presence of movie player) and values specify parameter values in the current context. This is, however, a *raw model* of the context. While adaptation rules can be written directly addressing the raw parameters, this solution is nowadays neither practical nor scaleable. As a result, we observe an emergence of context adaptation approach that is somewhat similar to stereotype modeling. I.e., a set of all possible combinations of name-value pairs is mapped to a smaller set of stereotypes that are, in turn, used by adaptation rules. For example, an adaptive system can distinguish two or more platforms *types* where each type is formed by a specific range of platform parameters. In this situation, platform recognition mechanism uses parameters of a specific platform to determine the current type. After that, simple adaptation rules check current type and perform different actions for different platform types. For example, if the user accesses the system from a handheld device, the system switches on conversion [112; 115; 206] or generation [15; 149] of presentation for a small screen. If a user platform cannot show color pictures or the bandwidth is low, the system converts the pictures to black-and-white or low resolution [166]. If the platform cannot show movies due to the absence of a movie player or low bandwidth, it can replace the movie with a picture or remove a link to the movie [93; 95]. More advanced technologies can generate considerably different interfaces for different platform types [11; 46; 62] and even use platform limitation to the benefits of user modeling [12]. For example, a Palm Pilot version of an adaptive news system [12] that is characterized by a small screen and low bandwidth requires the user to request pages of news story one by one thus sending each time an implicit relevance feedback to the system.

User Location. Mobile context-adaptive systems naturally focused on adaptation to user location. The modeling and the use of location is slightly different from other context elements. Most frequently the location is used not to fire adaptive presentation rules, but to determine a small subset of nearby objects of interests. This subset defines what should be presented or recommended to the user. This kind of adaptation was explored by early context-adaptive systems in several contexts such as museum guides [142], tourist guides [36], and marine information system [70]. Accordingly, user location is being modeled in a way that supports determining the nearby objects. Depending on the kind of location sensing it is typically a coordinate-based or zone-based representation. More information about location-adaptive systems can be found in Chapter 17 of this book [109]**.**

A Broader View of the Context. More recently, research on mobile and ubiquitous computing has considerably expanded the notion of context [173]. While there is no definite agreement about what should be included into the area of context, most of the work on context adaptation in mobile and ubiquitous computing focuses on a common core that includes *environment* and *human* dimensions [91; 173]. The environment dimension includes spatio-temporal aspect and physical conditions (light, temperature, acceleration, pressure, etc.). The human dimension includes personal context (user pulse, blood pressure, mood, cognitive load, etc.), social context [154], and user task. This may look confusing to the user modeling researchers who consider user tasks as a part of the user model, not context. To avoid confusion, it is important to remember that the research on context modeling is going on in two different research communities that consider context from two different points of view (Fig. 1.2). From the user-centered view employed in the user modeling field, the user task is not a part of the context, while the device itself is. From the device-centered view, which is dominant in the mobile and ubiquitous computing, a range of parameters characterizing the current state of the user are, indeed, a part of the device context. This view is presented in Chapter 17 of this book [109], which provides an extensive discussion of context modeling and reviews its use in several kinds of adaptive systems.

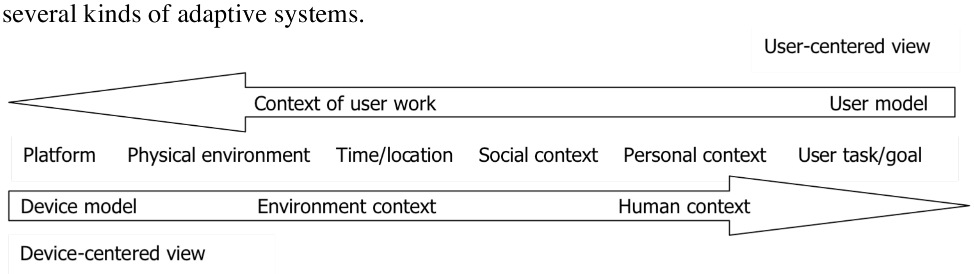


Fig. 1.2. The dimensions of context from user- and device-centered views

To establish a more objective border between user and context modeling, it is useful to observe that user modeling is focused mostly on the longer-term properties of the user that are distilled from observations, while context models attempt to represent the current features of the user and the environment, mostly read from context and physiological sensors [91; 173]. At the moment it is hard to classify the approaches to broader context modeling and adaptation in adaptive Web systems since these systems rarely attempt to model context beyond the platform and location. However, it seems that the emerging stereotype approach for platform adaptation can work within a range of other context parameters. Namely, a system may map a long list of name-value pairs supplied by sensors into a small set of meaningful pre-defined stereotype contexts. Once the current stereotype context is recognized, the system can use adaptation rules that address these stereotypes. For example, a context-aware application on a mobile device may use several parameters to recognize whether a user is rushing somewhere or not (two context stereotypes) and depending on the current context stereotype present information about the user's flight in different ways [137].

Affective State. An important context dimension that deserves to be mentioned separately is the user's affective state. Influenced by the idea of affective computing [156], the research on modeling using the affective state has become very popular in both user modeling and ubiquitous computing areas. A sequence of workshops devoted to "affective modeling" has been held in conjunction with the biannual User Modeling conference series since 1999. The proceedings of the last workshop in the series demonstrate the ranges of issues that are being explored now [30]. The methods used to model user affective state inherited approaches from both research areas mentioned above. While the methods applied by researchers with ubiquitous and pervasive computing backgrounds focus mostly on using various sensor input, researchers with user modeling background explored a range of approaches based on observing user-system interaction. Most important in the context of this chapter is the small stream of works on modeling the user's affective state by observing the user's Web log data. Despite the relatively low bandwidth of this source, a number of researchers have demonstrated that it can be used to detect user motivation, frustration, engagement, and disengagement [7; 9; 38; 193]. It is also interesting to observe that Bayesian networks emerged as the most popular technology to process both sensor input and the user action logs into user affective state [7; 42; 159; 193]. Section 1.4 of this chapter, which focuses on Bayesian approaches to user modeling, provides some further information on Bayesian affective modeling.

1.2.7 Feature-Based Modeling vs. Stereotype Modeling

Feature-based user modeling reviewed above is currently the dominant user modeling approach in adaptive Web systems. Feature-based models attempt to model specific features of individual users such as knowledge, interests goals, etc. During the user's work with the system, these features may change, so the goal of feature-based models is to track and represent an up-to-date state for modeled features. An alternative to the feature-based modeling is stereotype modeling.

Stereotype user modeling is one of the oldest approaches to user modeling. It was developed in the works of Elaine Rich [163; 164] over a quarter of century ago and elaborated in a number of user modeling projects. Stereotype user models attempt to cluster all possible users of an adaptive system into several groups, called stereotypes. All users belonging to the same stereotype are treated in the same way by the adaptation mechanisms. A user in a classical stereotype-based system is represented simply as her current stereotype (i.e., a group she currently belongs to). Naturally, each stereotype groups together users with specific mixture of features. However, stereotype modeling ignores the features and uses the stereotype as a whole. More exactly, the goal of stereotype modeling is to provide mapping from a specific combination of user features to one of the stereotypes. After that, only the user current stereotype is used for adaptation. Any changes in the user's features are responded to by simply re-assigning a user, if necessary, to a different stereotype. Elaine Rich discusses extensively when would be the right moment for this [164].

Stereotypes were extensively used in early adaptive systems between 1989 and 1994. A good overview of this generation of stereotype-based systems is provided in [101]. More recently, stereotype models were overshadowed by feature-based models, yet stereotypes were used in a number of adaptive Web systems [3; 5; 71; 127; 192]. In addition, as was mentioned above, the techniques developed for stereotype modeling and adaptation are used beyond classical stereotype modeling to manage low-granularity feature-based models.

A promising direction for the future application of stereotype models is their use in combination with feature-based models. One of the most popular combinations is the use of stereotypes to initialize an individual feature-based model [3; 5; 192]. This approach allows to avoid a typical "new user" problem in feature-based modeling where effective adaptation to new user is not possible since user modeling is started "from scratch." A good example of this combination is provided by SeAN system [3] for adaptive news presentation. Starting the work with a new user SeAN attempts to map the set of demographic features provided by the user (such as age, education, and type of job) into a set of predefined stereotypes. These stereotypes, in turn, are used to initialize the feature-based model of user knowledge and interests. Another promising direction for stereotype-based modeling is its use in combination with group models. Group models are becoming increasingly popular in Web personalization. More information about group modeling is provided in Chapter 20 of this book [92].

1.3 The Overlay Approach to User Modeling

This section provides more detailed information about an overlay approach to user modeling, the approach that is at the same time most important and most popular for AES and AHS. We start with overlay knowledge modeling, which was the original application of the overlay approach. After that we review the use of overlay approach for modeling user interests and discuss a generalization of the overlay approach for modeling other user features.

1.3.1 Overlay Modeling of User Knowledge

As we mentioned above, the idea of overlay knowledge modeling is to represent an individual user's knowledge as a subset of the domain model that resembles expert knowledge of the subject. Overlay models of user knowledge were introduced and developed in the field of ITS where overlay models were used mainly by systems with task sequencing, curriculum sequencing, and instructional planning functionalities. The popularity of this approach among early AES and AHS systems can be explained by their strong connection with ITS systems. In fact, a number of early AHS were developed in an attempt to extend an ITS system with hypertext functionality [8; 26; 75; 155]. The overlay knowledge models proved to be a good match for the core function of AHS: providing personalized access to information. As a result, within just a few years these models were accepted as de-facto standard by almost all educational and many non-educational adaptive hypermedia systems. This section provides more details on both components required for overlay knowledge modeling: the domain model and the overlay knowledge model.

The Domain Model. The heart of the overlay approach to knowledge modeling is a structured *domain model* that decomposes the body of knowledge about the domain into a set of *domain knowledge elements*. These elements can be named differently in different systems—concepts, knowledge items, topics, knowledge elements, learning objectives, learning outcomes, but in all the cases they denote elementary fragments of domain knowledge or information. In this paper we will be referring to these fragments as *concepts*. Though this name is slightly misleading[1] it is currently the most popular way to name domain knowledge elements. Depending on the domain, the application area, and the choice of the designer, concepts can represent larger or smaller pieces of domain knowledge: from a relatively large chunk of knowledge (i.e., a topic) down to elementary facts [143; 171], rules [167], or constraints [134].

The simplest form of domain model is formed by a set of independent (unrelated) concepts. This kind of model is called a *set model* or a *vector model* since the set of concepts has no internal structure [20]. Even this simple form of overlay modeling provides a powerful platform for maintaining a detailed picture of user knowledge that can support a fine grain adaptation in relatively large practical systems [22; 52]. The main problem of vector models is the lack of connection between concepts. A vector model can register user knowledge of a specific concept, but this does not help to model user knowledge of other concepts. As a result, when the number of concepts is large and the number of observations is not sufficient, the system is only able to predict user knowledge for a very small fraction of concepts, related to existing observations.

In a more advanced form of domain model concepts are connected to each other thus allowing some inter-concept inferencing. By its origin, it is possible to distinguish two main types of connected models. The first, relatively rare type is inspired by the ideas of instructional planning and used in a number of AES and educational AHS [32; 110; 117; 176]. The model is formed by a tree of educational objectives where larger objectives (starting with the whole course) are progressively decomposed into smaller objectives.

---

[1] The word "concepts" can cause someone to think that concepts can only represent fragments of conceptual knowledge. However, a concept is a general name that can denote a fragment of knowledge of any kind, including procedural knowledge.

The second type is both more general and more popular. In this type of domain model, concepts can be connected by different kinds of relationships, thus forming an arbitrarily complex network. This kind of model (known as a *network model*) was inspired by research on semantic networks (Fig. 1.3). Network domain models were used in many AHS and AES, including several development frameworks [6; 24; 50; 83; 148; 160; 169; 183; 189; 199].

Existing systems use network models of various complexity, with one or more types of links. In educational AHS and AES, the most popular links are prerequisite links between concepts, representing the fact that one of the related concepts must be learned before another. Prerequisite links can support several adaptation and user modeling techniques. In many AHS, prerequisite linkage is the only relationship given between concepts [81; 86; 141; 148; 157; 204]. More advanced educational systems as well as adaptive information systems favor classical semantic links such as "is-a" and "part-of" [23; 51; 183; 184; 189; 199].

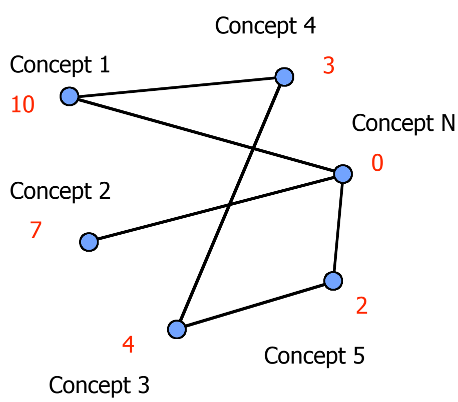


Fig. 1.3. A network domain model with a simple numeric overlay user model

All kinds of links between concepts are used to improve the precision of user modeling. When the user demonstrates a lack of knowledge, links can help to locate the most likely concepts that will remedy the situation. For example, if the user failed to answer a question that is based on the knowledge of several concepts, the concept that has fewer connections with the well-known concepts is the most likely source of problem. When the user demonstrates the presence of knowledge, links between concepts allow the propagation of the presence of knowledge beyond direct observation. For example, observing evidence of knowledge about a concept, the system can deduce the presence of knowledge for its prerequisite concepts. This approach is known as knowledge propagation. Different kinds of links may cause different types of knowledge propagation. An interesting approach for knowledge propagation is used in the AHA! system [48], which allows AHS authors to define new concept relationships types and their corresponding propagation semantics.

Recently, research on network domain models have been taken to the next level by several research teams that have attempted to use more formalized and more elaborated forms of network domain models inspired by the Semantic Web. The core idea here is to use formal domain *ontologies* [56; 58; 103; 120; 185] or domain *topic maps* [55; 146] in place of informal network domain models as a basis for overlay knowledge modeling. More information about the application of Semantic Web approaches in adaptive Web systems can be found in Chapter 23 of this book [57].

The Overlay Knowledge Model. The most important function of the domain model is to provide a framework for the representation of the user's domain knowledge using the overlay knowledge model. The key principle of the overlay model is that for each domain model concept, individual user knowledge model stores some data that is an estimation of the user knowledge level of this concept. The overlay model is powerful and flexible because it can measure independently the user's knowledge of different concepts. In the simplest (and oldest) form it is a binary value (known – not known) that enables the model to represent user's knowledge as an overlay of domain knowledge. This pure form of overlay model was used in several early AHS.

An extension of the pure overlay model is a *weighted overlay* model that can distinguish several levels of user's knowledge about each concept. This approach is used by the majority of modern AHS and AES. There are three popular forms of weighted overlay models: qualitative, simple numeric, and uncertainty-based that correspond to three approaches to user modeling, weight propagation, and adaptation. Qualitative models represent user knowledge of a concept as a qualitative value [22; 148] (for example, good-average-poor). These models are used mostly by systems with rule-based user modeling and adaptation components since qualitative values are very easy to update and use with rules. Simple numeric models use a quantitative value (for example, from 0 to 100) to represent the level of user knowledge [24; 51]. These models are explored by systems with simple algebraic approaches to knowledge modeling and propagation. The uncertainty-based models use different forms of uncertainty management such as Bayesian networks of fuzzy logics to model user knowledge. The user knowledge in these models are represented in the form dictated by the selected approach – most frequently, a probability that the user knows the concept [82; 177] or a probability distribution [3]. Due to its importance, the uncertainty-based approach is presented in detail in section 1.4 of this chapter.

One extension of the weighted overlay knowledge model is a *layered overlay model*. A layered model stores several values to represent user knowledge of each concept. Layered models were suggested to avoid mixing together the estimations of user knowledge obtained from different sources. For example, the system may choose to store separately the levels of user knowledge obtained by direct observation and by weight propagation [189; 203] or levels of user knowledge corresponding to different levels of concept mastery [27] such as Bloom's Taxonomy of Educational Objectives [13]. In these systems, different layers are maintained separately and mixed only in the process of adaptation decision making.

From the technical side, overlay knowledge models of individual users are typically stored as a set of name-value pairs where the names indicate domain model concepts and values represent the level of knowledge according to the representation formalism used in a system. Advanced overlay models such as layered models, fuzzy models, and models with probability distributions are stored as name-aspect-value triples. The details and the structure of the domain model is stored once in the system and is typically not replicated for individual users. As a result, the amount of information stored for each individual user is very small, even in systems with very large domain models.

1.3.2 Overlay Modeling of User Interests

Overlay approach to user interest modeling is very similar to overlay knowledge modeling. User interests are represented as an overlay of a concept-level model of the domain that the system covers. In general, domain models used for interest modeling and those used for knowledge modeling are slightly different in their structure and the size of concepts. However, this is caused mostly by the difference in system types: most of the domain models for knowledge modeling are developed for AES while most of the models for interest modeling are developed for adaptive information systems. This is rather a tradition than a necessity – a number of AES model user interests [85; 146] and many information systems model user knowledge [3; 143]. Conceptually domain models used for interest and knowledge modeling are compatible: most of the models build for knowledge modeling can be used for interest modeling and vice versa. A number of sophisticated AHS take an advantage of this compatibility suggesting modeling both user interests and user knowledge as two separate overlays over the same network of concepts [3; 143; 146; 171].

The variety of known domain models used for interest tracking can be classified into three groups that bear analogy to similar models used for knowledge tracking. First group is formed by systems that track user interests using a vector model: a set of unrelated concepts. In these systems, each information object (a document or a fact) is associated with one or more concepts. A demonstrated interest in an object is modeled by increasing interest in corresponding concept(s). For example interest in a craft object made in a specific style may increase the level of interest in this style [143]. Despite its simplicity, the vector model allows a reliable prediction of user interests for items that are related to concepts with previously registered interests. However, vector models do not allow the prediction of user interests for previously unexplored concepts, which becomes a serious problem as the set of concepts grows. Nowadays vector models are mostly replaced by two kinds of connected models – a taxonomy model and an ontology model. A *taxonomy* model is formed by a classification hierarchy of concepts that are frequently called topics, classes, or categories. Parents and children in this hierarchy are connected by topicsubtopic relationships. This kind of model is preferred by systems with expandable corpus such as Web directories or adaptive news systems [3; 128; 187] as well as similar open corpus systems reviewed in Chapter 2 [72] of this book. A new resource could be easily integrated into a hierarchy of concepts by attributing it to one of the leaf concepts (this classification can be done either manually by content provider or automatically). To increase the power of this model, it is possible to use more than one classification hierarchy to model different aspects of user interests. However, with some exclusion [96], this faceted classification is used only in closed corpus systems since it requires classifying each item along each of the taxonomies.

In an *ontology* model of interests the concepts form a rich network connected by different kinds of links. Most typical links are the same as in a network modeling of knowledge: is-a, part-of, and similarity. This kind of model is preferred by closed corpus adaptive information systems such as museum or tourist guides [171] or store catalogues [4]. The ontology model provides a richer representation of the world behind the information system and allows better interest tracking and propagation. Note that many adaptive systems that claim to use ontology for interest tracking are really using taxonomies that could be considered as a simple case of ontology. In contrast, some systems attempt to use very sophisticated ontologies such as WordNet to track user interests [118].

Both groups of domain models that represent links between concepts allow inter-concept interest propagation that is very important to fight sparsity and to increase the precision of interest tracking. In simple hierarchical domain models interest is typically propagated from child to parent concepts. For example, observed interest in *machine learning* causes increase of user interest in the parent concept *artificial intelligence* [128]. Richer domain models based on concept networks and "true" ontologies allow broader interest propagation. Typically these approaches ignore the nature of the links: if the system registers user interest in a specific concept, it assumes that related concepts in the domain model are also of some interest to the user [170; 187]. Some recent advanced approaches attempt to increase the precision of interest propagation using rules that take into account the structure and the type of connection in the domain model. For example, if a user is interested in a driver of a specific racing team, the user is probably interested in the team itself [43].

1.3.3 Generalized Overlay Models

User features beyond knowledge and interests are not formally modeled as overlays of domain models; however the approaches to modeling such features as goals, backgrounds and even whole stereotypes bear some reasonable similarity to the overlay approach. The framework for modeling these features is based on a space of possible characteristics (a set of possible goals, or a set of possible stereotypes) that is technically similar to a space of all concepts in structured domain models. To reveal this similarity, [19] suggested considering *generalized domain models* and *generalized overlay models*. A generalized domain model is a set of *aspects* where aspects can represent any characteristics of the user such as domain concepts, domain tasks and goals, and possible stereotypes. A generalized overlay user model is a set of pairs "aspect-value" in which the value in each pair can be "true" or "false" (indicating if the user has this characteristic) or some qualitative or quantitative value (measuring an extent to which user has this characteristic).

This generalization turned out to be very useful to find deep similarities between different AHS. Moreover, the ability to draw a similarity between approaches to model a specific user feature (such as background) and the very well explored overlay modeling approach could be instrumental for developing of more powerful and elaborated models. Below we provide some examples of using this similarity by re-considering approaches to goal and stereotype modeling.

Let's start with goal modeling. As reviewed in section 1.2.3, a system that is able to adapt to user goals typically uses a catalogue of user goals or tasks that the system can recognize. A traditional approach to goal modeling attempts to recognize the current goal in the catalogued set and mark it as current in the user model. Now let's consider the set of all goals to be a generalized domain model for goal modeling. As in traditional domain models, the set of goals can be unrelated or can form some connected structure – such as goal-subgoal hierarchy. In this context, the generalized overlay model of individual user can represent a probability that each specific goal in the generalized model is the current user goal. Now the traditional way of goal modeling by marking the current goal becomes simply a very primitive case of this generalized goal model (current goal has overlay value 1 and the others zero). Using this framework in full power allows building more elaborated goal models that can reflect uncertainty in modeling goals. In addition, recognizing the similarity with overlay knowledge modeling allows to re-use powerful uncertainty-based modeling and propagation techniques. Indeed, a few adaptive systems use some elements of this generalized vision to provide a more elaborated goal modeling. For example, some systems maintain a hierarchy of possible goals or tasks [198] and a few pioneer systems use probabilistic overlay modeling where each possible goal in the catalogue is represented by the system as a probability that it is the current user goal [63; 78; 126].

A similar generalization can be used for representing user stereotypes, backgrounds, or individual traits. For example, a stereotype user model is based on a collection of possible user stereotypes. A typical stereotype-based system attempts to recognize current stereotype, mark it as current in the user model and use for adaptation. An application of a generalized overlay approach allows us to treat a set of all possible stereotypes as a generalized domain model. A generalized overlay model now allows handling uncertainty of stereotype recognition as well as to represent more sophisticated cases of stereotype modeling such as the ability of the user to belong to several stereotypes at the same time. With this generalization, stereotype user models can be represented as a set of pairs "stereotype-value," where the value of each pair can either be "true" or "false" (which means that the user belongs or does not belong to the stereotype) or some probabilistic value (which represents the probability that the user belongs to the stereotype). A good case for maintaining a probabilistic overlay for modeling user stereotypes is provided by SeAN system [3] that uses a popular combination of stereotype and overlay modeling. SeAN attempts to map the set of demographic features provided by the user into a set of predefined stereotypes that are used to *seed* the overlay model of user knowledge and interests. Unlike typical stereotype modeling systems, SeAN is not trying to come up with the single, best stereotype for each user, but maintains a probabilistic overlay. This helps the system to achieve a more precise seeding of the overlay model.

The ability to view models of different user characteristics as generalized overlay models allows developers to better understand the similarity between different systems and to reuse more broadly the representations and user modeling approaches developed in the field. For example, as demonstrated in the next section of this chapter, elaborated approaches for knowledge modeling based on Bayesian networks could applied for modeling other user features such as individual traits.

On the practical side, the vision of generalized overlay models is currently supported by some universal authoring frameworks for developing AHS. The best example of such a universal framework is AHA! [49; 50]. AHA! allows the author of an AHS to introduce generalized concepts and relationships between the concepts. The concepts in AHA! can represent different user aspects such as knowledge, goals, interest, etc. For each concept, the system maintains an individual numeric overlay value. The semantics of the concepts, the connections, and the meaning of specific overlay values are encoded by the author into a set of user modeling rules and weight propagation rules. The universal nature of AHA! made it the most popular authoring tool in the field of adaptive hypermedia. So far, AHA! has already been used for modeling such user features as knowledge, interests, and even learning styles [181].

1.4 Uncertainty-based User Modeling for Adaptive Hypermedia and Adaptive Educational Systems

There is no doubt that in the case of user modeling, there is often the need to deal with information that is uncertain (we are not sure that the available information is absolutely true) and/or imprecise (the values handled are not completely defined). An example of a statement that we need to deal with would be: “the user failed this question, so *most probably* he/she doesn’t know concept C,” which is *uncertain* information. On the other hand, if we say, “the user has been reading about concept C for *quite* a long time” we are making an *imprecise* observation. Obviously, user modeling is a domain in which there are many different sources of uncertainty and/or imprecision, therefore numerically-approximate reasoning techniques are suitable for this purpose. The two more commonly used in this context are Bayesian Networks and Fuzzy Logic.

Bayesian Networks (BNs) were developed in the eighties by Judea Pearl [153] and since then there has been an increasing interest and enormous progress in the development of new techniques and algorithms, extensions to the model, and applications. BNs are a probabilistic model inspired by causality and provide a graphical model in which each node represents a variable and each link represents a causal influence relationship. Currently they are considered one of the best techniques available for diagnosis and classification problems. The Fuzzy Logic (FL) paradigm was originally proposed by Professor Lofti Zadeh in 1965 [207], and the basic idea is to allow membership functions (in Fuzzy Sets theory) or truth values (in Fuzzy Logic) to take values between 0 and 1, with 0 representing absolute falseness and 1 absolute truth. As in BNs, FL has been the object of intense research, both in the development of models and in its application to real problems, with special emphasis on control problems. Therefore, it is no surprise that in the field of adaptive systems researchers have used such models, since the development of an adaptive system involves diagnosis, classification, and control.

However, in the field of adaptive hypermedia there are few studies that report the use of approximate reasoning techniques (notable exceptions are [82] for BNs and [99] for FL). Probably, the reason is that researchers are devoting much more attention to other relevant aspects such as foundation and core techniques (data mining, meta-data, semantic web, intelligent web agents, web services, etc.) or practical aspects (architectures, privacy, security, usability, evaluation, etc). The situation might be comparable to what happened in other related fields ten years ago (for example, in the field of Intelligent Tutoring Systems only four of the papers presented at ITS'1996 reported the use of such techniques (all of them used BNs), while at ITS'2006, eight papers use BNs and one paper uses FL. Our prediction is that in the future the number of adaptive web-based systems that use BNs or FL will continue to grow.

Nowadays, most researchers who decide to use approximate reasoning techniques in their user models choose the Bayesian network paradigm. The reason for this is the suitability for diagnosis. Also, the fast development of this field over the last twenty years has resulted in an increasing range of techniques and tools for Bayesian reasoning. In this sense, tools like GeNIe (SMILE), HUGIN, or JAVABAYES allow for the easy definition of Bayesian models and seamless integration of learning and updating algorithms. Still, there are also a number of systems that use other approaches like Fuzzy Logic [37; 98; 108] and, more recently, neuro-fuzzy approaches [182]. The interested reader can see [90] for an excellent review of former systems and issues in the field of user modeling, and [99] for some discussion about fuzzy user modeling in the context of adaptive web-based applications.

At the time of this writing, most solid experience in Bayesian user modeling has been accumulated in the area of modeling users that interact with educational systems, also known as student modeling. Student modeling was always the most area for AES and AHS and it was also the area where new technologies were developed and explored before being applied for other types of user modeling. Therefore, the core part of this chapter is focused on Bayesian student models (BSMs), beginning with the modeling of user knowledge. We will then briefly discuss the use of BNs beyond modeling knowledge. Our goal is to explain in detail how to define a Bayesian student model (BSM), which student features can be modeled, and how the parameters of the model can be obtained. The use of BSM in adaptive web-based systems is still quite unusual. Because of this, we will also refer to more relevant models that have been deployed in stand-alone tutoring systems and learning environments. We do this to illustrate how such techniques might be employed in adaptive web-based educational systems following the success of stand-alone implementations.

1.4.1 Basics of BNs

A Bayesian Network can be formally defined as:
- A set V of propositional variables (X1, …, Xn) (nodes of the network)
- A set E of probabilistic relationships between the variables (arcs of the network)

Such that:
- The graph G = (V,E) is an acyclic directed graph.
- The conditional independence assumptions are satisfied, i.e., each node Xi is conditionally independent from the rest of the nodes (except from its descendants), given the state of its parents.

With respect to nomenclature, the words "node" and "variable" are used interchangeably in literature. The probabilistic relationship between variables is usually referred to as an "arc" or "link."

Though the relationship between the variables does not need to be causal in nature, the BN paradigm is very suitable for that case. Therefore, building a model usually involves putting the information available into a cause-effect schema. Regarding the probability distribution, it can be easily shown that if the conditional independence assumptions are satisfied, then the joint probability distribution is given as the product of the probability distribution of each node given its parents. Because of that, in the case of BN it is enough to provide the conditional probability distribution of each node given its parents (or prior probability distribution for root nodes) to make *any* kind of inference needed. In that sense, a powerful feature of BNs is that they allow for diagnosis (inferences about possible causes of an event) and prediction (future state/evolution of variables given evidence).

To illustrate the concepts presented above, let us consider the following example in the context of student modeling. In order to determine if a student knows a certain domain element K, we can use the result of a certain event E that provides evidence about it. This evidence-bearing event can be an answer to a test item, solution of a problem, teacher's opinion, the number of Web pages relevant to the element K that have been visited, etc. In that case, the nodes of the network are: node K (having knowledge about a domain element K), and node E (result of the evidential event E). In the simplest case, both variables are binary: K can be *known* or *not_known*, while the evidence provided by E can be *positive* or *negative*. To simplify the notation, we will denote the positive states (*known*, *positive*) by 1 and the negative states (*not_known*, *negative*) by 0. There is only one relationship: the state of variable K influences (in this case, causally) the result of the event E (Fig. 1.4).

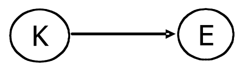


Fig. 1.4. The simplest BN for knowledge diagnosis

Let's concretize our example. A sample concept could be "Adding natural numbers" and its possible values could be *known* and *not_known*. A possible question could be "3+4" which has the possible states *right* and *wrong* (Figure 1.5).

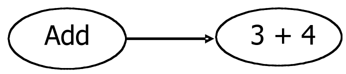


Fig. 1.5. A simple BN for diagnosing knowledge about adding natural numbers

Regarding the quantitative information (network parameters), we need to provide the conditional probability distribution of each node given its parents (for the nodes without parents, the a-priori probability distribution). So in the simplest general case, three probabilities need to be specified: P(K=1), P(E=1|K=1) and P(E=1|K=0). Let us use the following values for these parameters:

- P(K=1) = 0.2, means that, in our experience (for instance, if we are a math teacher), the probability that a random student knows how to add integers is 0.2.
- P(E=1|K=1) = 0.99, means that the probability of giving a correct answer to "3+4" is 0.99 for a student who knows how to add integers. Conversely, although the student knows how to add, there is still the probability of 0.01 of giving an incorrect answer, allowing us to model what is called a "slip" in related literature.
- P(E=1|K=0) = 0.02, means that the probability of giving a correct answer to "3+4" is 0.02, if the student does not know how to add fractions. Conversely, even if a student does not know how to add there is still a certain probability of guessing the correct answer.

In the related fields of medical diagnosis and statistics such parameters have a clear meaning. For example, in the medical diagnosis domain, if D represents an disease and T a test used to diagnose it, the causal relationship is D → E, and the parameters needed are: the a priori probability of the disease (which in medicine is called the *prevalence* of the disease), and the conditional probability distribution P(T/D), more concretely P(T=1/D=0), which is the rate of *false positives* (or *type II error* in statistics) and P(T=0/D=1), which is the rate of *false negatives* (or *type I error* in statistics) of test T. Both measures are commonly used as indicators of the quality of the test T or its suitability for the disease D.

Of course, the more complex the structure of the network, the higher the number of parameters needed. In general, if a node N has n parent nodes $P_1, \ldots, P_n$ and each of them has k states, the number of parameters needed for node N is k^n, i.e., the number of parameters needed grows exponentially with the number of parents.

Once the network has been defined, it can be used to make inferences about the domain. Using a BN model allows us to reason both in the diagnostic direction (what are the more probable causes given certain evidences) and in the prediction direction (what is the probability that a given configuration of variables states will happen, given a set of evidence?). To illustrate this, we will add some more nodes to our example (Fig. 1.6).

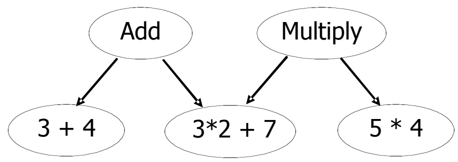


Fig. 1.6. Adding nodes to the basic BN

Let us imagine that our student gave a correct answer to "3+4." We can then reason in the diagnostic direction (i.e., compute the probability that he/she knows concept "Add") and also to predict a future result (i.e., compute the probability that he/she will be able to correctly solve "3*2+7"). This potential for doing both abductive and predictive reasoning is very useful in the case of student modeling. Not only does it allows us to accurately diagnose the student's current state of knowledge, but also to make an informed and adaptive recommendation for the most appropriate next activity (e.g. exercise, test) according to the evidence available about this particular student (previous activities with the system, favorite learning strategy, etc.). For example, for a student that likes to be challenged it might be better to select a "difficult" (lower probability of right answer) exercise while for a student that has low self-confidence it might be better to select an "easy" (higher probability of right answer) exercise.

As pointed out before, BN formalism implements certain independence assumptions amongst variables. Conditional independence modeled in a particular BN model should correspond to independence relationships existing in the real world. In our example, the conditional independence assumptions mean that the knowledge nodes "Add" and "Multiply" are independent a priori (which is reasonable, because if they are not the model should include a link between) and that each evidential node is conditionally independent of the rest of the evidential nodes given the values of its parents (which is also reasonable, because once the value of the corresponding parent nodes, i.e., knowledge nodes, is known, other evidential nodes do not provide information any more). For more information about what the conditional independence assumptions mean, see [140] or [153].

The construction of a student model based on BNs consists of two steps: (a) development of the *qualitative model*, that involves the definition of the structural model (nodes and arcs); and (b) development of the *quantitative model*, that involves the specification of the parameters needed (conditional and prior probability or distributions). There are two ways of obtaining the information needed in the above steps. The first way is to use domain expert's opinion to get the structure/parameters. The second is to infer them from data using BNs learning algorithms [79]. A combination of these two ways is also possible and often employed in practice. As far as learning is concerned, s*tructural learning* is a process of learning dependencies between the given variables (step a), while *parametric learning* is a process of learning the parameters describing the strength of the dependencies in the model (step b).

If a BN is built by an expert, it is necessary to make sure that the model includes all relevant aspects of the real world, i.e., the model has to account for the relevant variables and relationships between them, and the specified conditional probabilities have to represent the strength of these relationships. On the other hand, if a BN is to be learned from data, the relationships learned on the significance level fixed for the statistical independence tests performed during structural learning. If the significance level is low we might miss some weak relationships while if the significance level is high some hard-to-interpret relationships might be inferred or assumed. A possible solution is to have an expert to specify the structure (or fine-tune the learned structure) and then learn parameters from data.

The structure of the rest of section 1.4 is as follows: In subsection 1.4.2, we will discuss the development of the quantitative model under the assumption that the structure of a BN is to be built by an expert. To this end, we will discuss which aspects of the real world being modeled (in our case, a student) can be taken into account and go through different modeling options[2]. Subsection 1.4.3 gives some clues for defining qualitative modeling, while subsection 1.4.4 discusses the combination of expert knowledge and learning techniques in building BN student models. Section we present some conclusions.

1.4.2 Development of the Qualitative Model

As in any modeling process, the first important step is to specify aspects of the world being modeled which should be represented in the model, in order to achieve the defined goals. In the case of student modeling, the range of features to be modeled is wide and can be grouped into two main categories: knowledge (declarative, procedural, based on skills, competencies, etc.) and user features (learning styles, cognitive and meta-cognitive skills, emotional states, etc). The more user features are represented in the model the more complete it will be. However, the cost of creating and maintaining a more complete model should be carefully balanced against the usefulness of such a model. In the user modeling field, the goal is usually to have a personalized model of the user, in order to provide the basis for adaptation. So we must carefully choose the user features that will be useful to provide adaptation, because if we are going to consider k different variables and each variable can take n different values, then we will need to provide adaptive strategies for the k^n different profiles of students generated by the combination of such values.

To begin with, we will explain how to use BNs to model student's knowledge based on overlay models. Then we will present how to build knowledge models for other

[2] A good tutorial about modeling with BNs for troubleshooting problems is [114].

approaches such us constraint-based models and misconceptions models. Next, we will discuss how to model features beyond knowledge, such as meta-cognitive skills, personality traits, affective states, and attitudes and perceptions. Finally, we mention some examples where the qualitative model has been developed by experts and some other examples in which learning techniques have been used to infer it from available data. Along the discussion we will present examples of application, which will illustrate the main ideas and concepts.

Modeling Knowledge Based on Overlay Models with Bayesian Networks. As explained before, the first step consists in selecting the variables of interest for our model, which in this case are the knowledge variables and the sources of evidence that will be used to obtain relevant information about them. Each one of them will be modeled as a random variable, and represented as node in the BN. In the case of user modeling, typically the sources of evidence are the results of interaction with the system (e.g. answers to questions or exercises, time spent reading certain content, number of clicks, etc.). Open/scrutable user models allow that either the user or the expert can see and modify this information.

Once the relevant aspects of the real world being modeled have been selected, it is time to translate them to a mathematical model. In the case of BNs, this means that we need to structure this information in a causal relationship schema. To this end, a couple of important aspects related to building BN models should be clarified: first, that each node must represent a propositional value; and second, that the proper direction of the links needs to be established. We will discuss each of these aspects in more detail.

As aforementioned, each node must represent a *propositional* variable [140], which is defined to be as a variable that takes an *exhaustive* and *mutually exclusive* number of values. This aspect allows us to determine whether a given element of the real world should be modeled as a variable or as a state of a variable. Let us illustrate this using some examples.

Example 1: A medical diagnosis problem

A doctor is trying to diagnose a patient to determine the illness (or illnesses) that he/she is suffering from. Let us define a finite set of illnesses as $\{I_1, \ldots, I_n\}$. In this case there are two modeling options: (A) to consider a node I with values $I_1, \ldots, I_n$, and (B) consists in considering n binary nodes $I_1, \ldots, I_n$. In the case of medical diagnosis, the choice depends on whether or not more than one illness can be present at the same time. If the patient can have one and only one illness, the correct option is A. If the patient can have two or more illnesses at the same time, the correct option is B. It is important to point out that in option A all the possible illnesses must be considered within the set of possible values $\{I_1, \ldots, I_n\}$, and if this is not possible, one of the states should be labeled “other” to account for any illness not taken into account in the model (thus making the set of states exhaustive). In option B, a node "other" can be included, but it is not necessary.

Example 2: Classification problems

We have to classify a set of objects as belonging to certain categories C_1, …, C_n. Again, the decision of using a single node C with values C_1, …, C_n (option A) or n separate binary nodes C_1, …, C_n (option B). The decision depends on whether or not an object can belong to more than one category. So if we are classifying animals in mammals, amphibians, reptiles, etc. we should choose option A (with the additional value "others" if needed). However, if we are classifying persons according to their educational background into doctors, architects, lawyers, etc. we should use option B.

Example 3. The use of the two different options in user modeling

This example illustrates how the two different options presented above have been used in two real systems: KBS Hyperbook [82] and the English tutor I-Peter [161]. In both systems, users are being classified according to their knowledge level K into the categories novice (N), beginner (B), intermediate (I), and advanced (A). But while KBS Hyperbook models student's competence in each knowledge item i by using a node K_i that takes values {N, B, I, A}, I-Peter models student's competence in English by using four separate binary nodes N, B, I, A. In the second system, the underlying assumption is that a student can belong to more than one category simultaneously. A possible interpretation is that the I-Peter model might be intended to allow smooth transitions between categories. For instance, a possible interpretation of P(N=Yes)=0.2 and P(B=Yes)=0.6 is that the user is gradually leaving the category *novice* to get to the more advanced state *beginner* (quite like in a Fuzzy Logic model). But other combinations of probability values might be harder to interpret, and therefore estimating the parameters needed might be difficult in I-Peter.

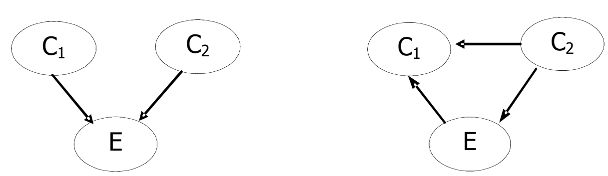


Fig. 1.7. A BN and the equivalent BN with the arc $C_1 \rightarrow$ E reversed

As already mentioned, a second important decision is the more appropriate direction for the arcs in our model, i.e., given two propositional variables X and Y, we need to decide if we will represent the relationship between them as $X \rightarrow Y$ or $X \leftarrow Y$. This is an especially important aspect, because in spite of the fact that the direction of any arc in a BN can always be reversed [89], BNs are very suitable for the case of causal relationships and therefore the model should stay as close to the notion of causality as possible. It is important to note that if the direction of an arc is reversed new arcs are introduced to the model. This is necessary for the new network to keep the same dependency structure. Fig. 1.7 shows the arc reversal result on a simple network:

Any of these two models would correctly represent the relationships between the two causes and the effect, but the second one (after an arc has been reversed) needs more parameters (which would also be harder to estimate and interpret). So using the proper direction of the arcs saves work and makes the definition of the model easier. To be able to choose the proper direction, we should think about which variables have causal influence in which others. For instance, if X is *type_of_animal* and Y represents a characteristic of an animal like *presence_of_feathers*, it seems clear that the type of animal has *causal influence* on the presence of feathers, and therefore the more adequate option is X → Y. However, it seems that humans find it easier to structure the information in terms of IF-THEN rules rather than in terms of causal relationships. Because of that, it is quite common to see the other option, perhaps influenced by the implicit construction in our mental model of the corresponding diagnostic rule (IF it has feathers, THEN it is an animal) instead of the causal relationship (it has feathers BECAUSE it is an animal).

In the case of student modeling, the basic overlay knowledge model usually includes the following relationships: granularity relationships, relationships between knowledge and evidential nodes, and prerequisite relationships. Examples of Bayesian student models that use both modeling options (arcs in both causal and diagnostic directions) for each of such relationships can be easily found in literature and will be discussed in the following paragraphs. We continue using the generic notation introduced in section 1.4.1: K will represent a knowledge node, while E will represent any event that can provide information (evidence) about the student's knowledge K (e.g., a question, test, exercise, task, etc.).

For modeling the relationships between the evidential nodes E and the knowledge node K, the options are:

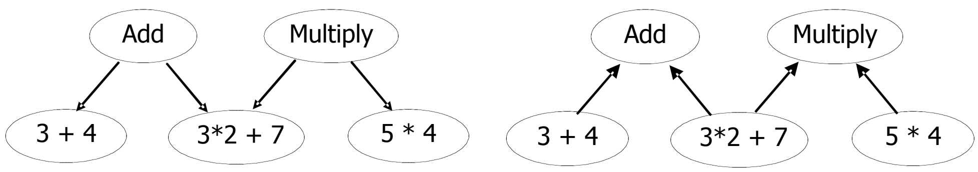
(o_1) causal direction, i.e., K → E; or

(o_2) diagnostic direction, E → K.

Option o_2 could be also interpreted as a causal relationship if the variable K is considered to be a measure of how well the student performed in a test. But usually the goal is to evaluate the knowledge of the student and not his/her performance. In fact, the performance can be seen as an effect of the knowledge, together with many other causes such us the emotional state of the student, the difficulty of the evidential task being performed, etc.

Let us see an example in the context of the adding/multiplying problem. The modeling options are as depicted in Fig. 1.8. If we use option (o_1), the nodes "Add" and "Multiply" represent the knowledge that the student has about those two operations. In option (o_2) the nodes "Add" and "Multiply" represent an artificial constructs that measures how well the student did in the test involving such operations.

Though this section is devoted only to qualitative model, thinking about the parameters needed in each option can help in taking a decision. In option (o_1) we need to estimate the prior probability distributions P(Add), P(Multiply), and the conditional probability distributions P((3+4)/Add), P((3*2+7)/Add, Multiply), and P((5*4)/Multiply). All of these parameters have an intuitive meaning: the priors represent the "prevalence" of the two basic operations in the population under study, i.e., how many of our target students know them, while the conditionals represent the probability of a correct answer given that the knowledge required has/has not been acquired. In option (o_2), we would need the priors P(3+4), P(3*2+7) and P(5*4), which can be meaningfully estimated, but also the conditionals P(Add/(3+4), (3*2+7)), P(Multiply/(3*2+7),(3+4)), which seem harder to interpret. For example, what is the probability that the student "performed well" in the test with respect to the operation "Add" given that his/her answer to 3+4 was incorrect and to 3*2+7 was correct?



(o_1) causal direction *(o_2) diagnostic direction*

Fig. 1.8. Options available for modeling relationships between knowledge and evidential nodes: an example

Both modeling options are present in the literature. In [196] evidence is provided by problems that the student needs to solve and knowledge nodes represent the physics rules that the student needs to apply. The conclusion reached in the paper is that if links are defined in the diagnostic direction it "means that the rules are conditionally independent given the evidence, which just isn't true (see [196] for an example). Therefore, the causal direction seems preferable in this context. The same conclusion is drawn in [130] that theoretically compares both options in terms of criteria such as knowledge representation, independence relationships, reasoning process, and knowledge engineering effort required. More recently, [119] performed an empirical study with the goal to compare the accuracy reached with each modeling option in the diagnosis of a group of real students, reporting a maximum accuracy of 0.508 if o_2 (the diagnostic direction) is used (no better than chance, in spite of their efforts of fine-tune the model using data) versus an average test set accuracy of 0.776 of o_1 (causal direction).

With respect to granularity relationships, options available are:

(o_3) the causal direction, i.e., $K_1 \rightarrow K$; or

(o_4) the diagnostic direction $K \rightarrow K_1$.

The first option (o_3) assumes that knowing a component indicates knowledge about the whole, while the second option (o_4) states that knowing the whole has influence in knowing each of its components.

In our example, we can define an aggregated knowledge element “Basic arithmetic,” which would be divided into “Add” and “Multiply.” These two modeling options are depicted in Fig. 1.9.

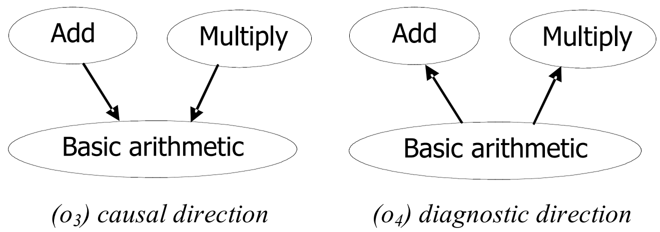


Fig. 1.9. Two options for modeling granularity relationships

The interpretation of both options follows. The implicit consideration under (o_3) is that if a student knows basic arithmetic, then he/she also knows how to add and how to multiply, while (o_4) assumes that a student has knowledge of basic arithmetic *because* he/she knows how to add and how to multiply.

In the context of granularity relationships, examples of BNs with links defined in the diagnostic direction (o_4) are: [82; 132; 139], while [119; 122; 130; 195; 208] use links defined in the causal direction (o_3). The first theoretical comparison between both options can be found in [90], which discusses some implications but does not explicitly recommend any of them. The causal direction (o_3) is supported by theoretical studies such us [130], which compare both options in terms of the same criteria described above, and empirical studies such as [39], which evaluate three different course hierarchies in the context of adaptive testing, using quality measures such as *test length* and *test coverage*.

Another interesting alternative for the modeling of relationships between knowledge and evidential nodes are dynamic models, which allows for changing the variables over time. This is a desirable characteristic in the context of student modeling since the knowledge and other relevant variables usually change continuously during a student's interaction with the system. In the *Dynamic Bayesian Networks* (DBNs) model [168], a separate BN is constructed for each time slice. This approach is also taken in [119; 122; 130; 209], probably inspired by Reye's work [162]. Reye presents a simple model based on DBN, which captures the dynamic nature of the student's knowledge. In this proposal, for each j = 1, ..., n, ..., the following nodes and relationships between them are defined (Fig. 1.10):

L_j = student's state of knowledge after the *j*'th interaction with the system

O_j = result of the *j*'th interaction.

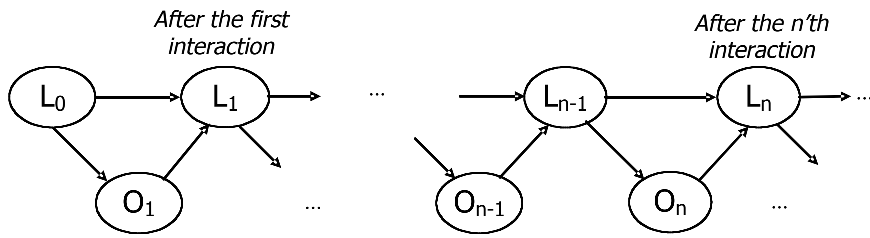


Fig. 1.10. Dynamic BN for student modeling. Adapted from [162]

We can see that at time n, the probability that the student is now in the learned state (L_n) depends on whether the student was already in the learned state at n-1 (L_{n-1}) and on the outcome of the last interaction with the system (O_n). Reye provides formulas for a two-phase updating of this basic model and proves that the knowledge tracing model used in the ACT cognitive tutors [44; 45] can be viewed as a particular case of this more general model (by adding constraints to the general model until the equivalence is shown). Please note that in spite of the fact that this description seems to refer to only one knowledge node, the implementations of this model will contain as many knowledge nodes as needed, resulting in more complex networks. A later study [131] makes the empirical comparison of a static BN with a DBN using simulated students, showing that both models produce a very similar performance in terms of test length and accuracy, although the static model seems to perform slightly better.

In summary, it seems that evidence available (both theoretical and empirical) encourages the use of the causal direction. The arc should then go from variables that cannot be observed directly and need to be estimated (knowledge in the case of student modeling, illness or disease in the case of medical diagnosis, faulty components in the case of trouble-shooting systems) to variables that can be observed (answers to problems, symptoms, or problems, respectively).

With respect to prerequisite relationships, it seems clear that if A is a prerequisite of B, knowing A has *causal influence* on knowing B, so the direction of the link should be A→ B. However, the main difficulty of introducing this new kind of relationship into our model is that, as reported in [31], the meaning of the relationships between the nodes becomes somehow unclear and the specification of the parameters becomes more difficult. To illustrate this, let us consider the following example: in a basic arithmetic course, students are taught how to add and multiply natural numbers (***N***) and fractions (***Q***). A basic overlay BN student model for this course (with links in the causal direction) is given in Fig. 1.11 where prerequisite relationships are represented as a dotted line.

In this example, one of the parameters needed is P(Multiply/Mult_N, Mult_Q, Add). But the fact that different types of relationships are mixed in the conditional distribution makes this probability difficult to estimate. Just as an example, we would need to provide the probability of knowing how to multiply, given that "the student knows how to multiply natural numbers and fractions, but does not now how to add," which is quite improbable. As suggested in [31], a possible solution is to disregard that kind of relationship in the model (thus making a simplification of reality, in which prerequisite relationships do exist). Another possible solution is to make a different simplifying assumption: instead of not including prerequisites, consider that both relationships operate at different levels, i.e., to build a multi-layered student model as proposed by [200]. In [31], Carmona et al present an empirical study comparing both alternatives. The study supports a conclusion previously established in [59]: "not considering valid prerequisites relationships does not lead to a wrong assessment of a student's knowledge state, but it renders the assessment less efficient in the sense that more answers than necessary have to be collected." So, if the test length is not an important issue, eliminating prerequisites will not affect the accuracy of our assessment, but if length is important, a possible option would be to use a multilayered model which can improve the performance (in the sense that less questions will be needed to reach the same accuracy for our estimation of the student's knowledge).

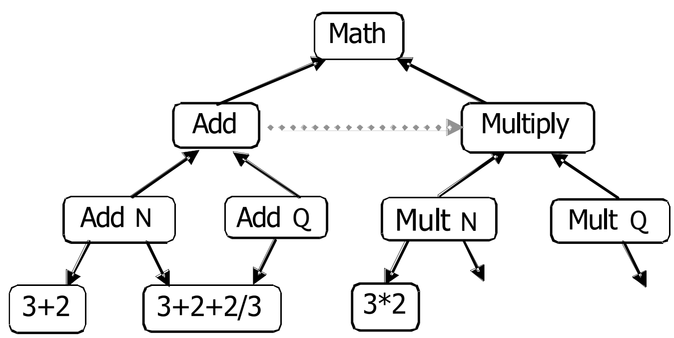


Fig. 1.11. Introducing prerequisites in the BN

Another option for considering prerequisite relationships is presented in [82] and is based on categorizing the knowledge nodes into levels. The first level corresponds to simple concepts, which do not require prerequisites. The successive levels require that knowledge about some of the concepts in the previous levels be understood. In this application, prerequisite relationships are only allowed between the root nodes of each level (recall that this system uses the diagnostic direction (o_4) for the granularity relationships, so the root nodes are the compound concepts). In this way, the problem of confusing different kinds of relationships in the model can be avoided.

Modeling Knowledge Beyond Overlay Models. There are also some proposals to model student knowledge with BNs that are not based on overlay models. Alternative approaches include constraint-based modeling and models based on misconceptions. A good example of constraint-based modeling is the student model of CAPIT [122], a constraint-based tutor for English capitalization and punctuation. In this model each node L_i represents the outcome of the student's last attempt with respect to the violation or not of a certain constraint I. That attempt can take values S (satisfied), V (violated), VFB (violated with feedback) or NR (not relevant). When a new problem is presented, node N_i represents the predicted outcome of this new attempt with respect to the violation of constraint i. The authors decided to include relationships between the constraints, which caused the model to grow more complex but was more realistic and allowed for a better prediction of student performance. The model also includes decision nodes and their utilities, i.e., transforming the BN in an Influence Diagram or Decision Network (DN). This allows for the selection of the next tutorial action by maximizing the expected utility. A similar approach based on DN has also been presented in [138].

As for bug models, a good example is the student model for a tutoring system on decimal numbers presented in [180]. In this case there are several types of nodes: *evidential nodes*, which can be either test items of a decimal comparison test (nodes TI) or variables that account for student behavior during the interaction with the system (nodes B) and *student-type nodes*, which serve the purpose of classification of students according to the type of misconception they have. This classification happens at two levels of granularity: coarse and fine. The basic structure of the network is as represented in Figure 1.12.

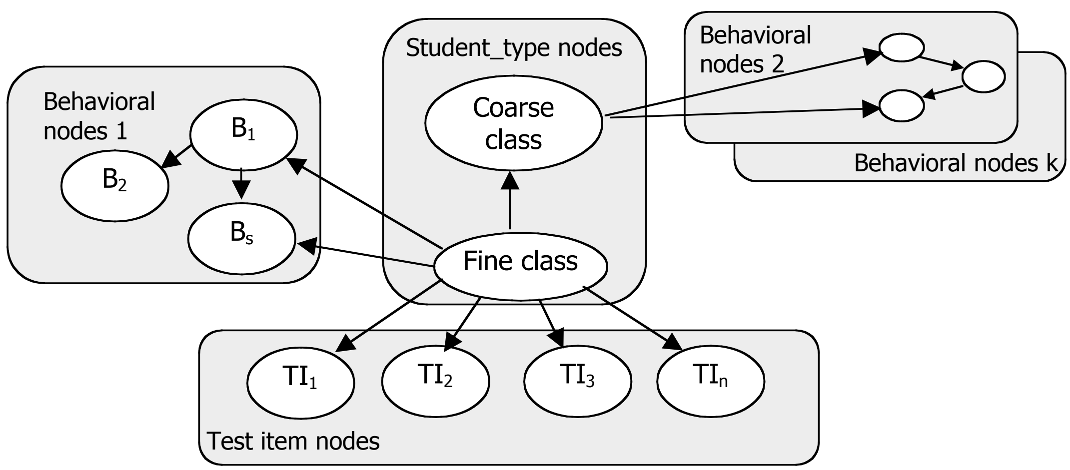


Fig. 1.12. A Bayesian student model based on misconceptions

This model features several groups of behavioral nodes. Those groups can be of different sizes and can have different internal structures (i.e., they can be connected in different ways). They model different instructional episodes. Please note that there is only one group of test item nodes and student type nodes. As it can be seen, in this student model the links follow the causal direction. It is important to note that the performance of this student model has been evaluated by comparing the results of manual expert diagnosis with the automatic system's diagnosis. The comparison took into account the results of more than 2,000 students who took the DCT test and showed an 80-90% agreement rate (the 10% variation was due to varying the values of certain parameters).

Beyond Modeling Knowledge. The basic knowledge models presented above can be enhanced by including information that allows for *model-tracing* (in addition to knowledge-tracing). This can be achieved by adding other variables like student goals, plans, etc. An example of the second kind of model is the Bayesian student model for the physics domain in the tutoring system Andes [40], which divides knowledge nodes into proposition nodes (fact and goal nodes), rule application nodes, and strategy nodes (which allow for modeling the different methods used to solve a problem, and therefore for model tracing). A similar approach was later adopted in [186] for the domain of medical diagnosis. The architecture of the Bayesian student model of the Andes system is depicted in Fig. 1.13.

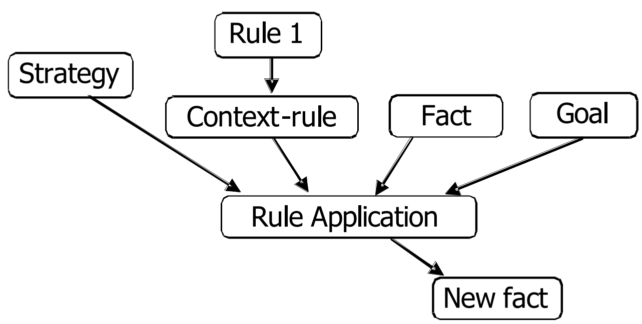


Fig. 1.13. General BN model for facts, rules, strategies, and goals

For example, in the physics domain, if the student knows a certain rule (such as F=m*a) he/she will be able to use it in any context. Then, if he/she also knows certain facts (for example, m=50kg and a=4m/s^2), has a certain goal (compute the force) and a certain strategy (way of solving the problem), then the rule will be applied to the problem and a new fact will be inferred (in this case, that F=200N). In the medical diagnosis example, the idea of "rule" is replaced by "medical concept" that can be applied to derive a hypothesis (equivalent to facts in the physical domain). In both domains (physics and medicine), the application, goal and apply nodes are used in the same way. Strategies are modeled by the fact that students can enumerate the causal hypothesis structure in any order. That's why they are not explicitly introduced as nodes in the model. One difference between these two domains is that causal relationships exist between hypotheses in the medical application but do not exist between facts in the physics tutor.

Recently, there has been increasing interest in widening the range of user characteristics that can be measured and used in adaptive systems. In this sense, BNs have been used to model the student beyond cognitive features (knowledge, goals) towards a richer variety of features, like:

- Meta-cognitive skills, for example, self-explanation [41] and exploration [29]. To support the evaluation of meta-cognitive skills the BN student model includes nodes to represent student's tendency to self-explain, to explore, and also nodes to model student's actions that are indicative of relevant applications of such abilities. Another recent example of modeling a meta-cognitive skill is [136], which presents a computational framework to support learning by using analogical problem solving. To this end, new nodes have been introduced in the Andes Bayesian student model: copy nodes (with values *correct*, *incorrect*, *no_copy*) which represent if the student copied the solution from the solved example; similarity nodes (with values *trivial*, *non_trivial*, *none*) that measure the degree of similarity of fact nodes between the example solution and the student's solution; analogy nodes (that represent the student's tendency to use the minimum or maximum analogies, i.e., to try to solve the problem on their own and consult the example only when reaching an impasse or to copy as much as possible), and an EBLC node, which allows modeling of the user's tendency to learn by the Explanation Based Learning of Correctness.
- Personality traits. An example is [68], that presented a BN to model learning styles within a web-based education system. To this end, they consider three dimensions of Felder's framework [64], namely perception (sensory/intuitive), processing (active/reflective), and understanding (sequential/global) as unobservable nodes, and several evidential nodes, such as the use of mail, forum, chats, number of examples visited, and exam results. In this case they are using the diagnostic instead of the causal direction for links between variables.
- Affective states. An example is the affective user model in the educational game Prime Climb, a game designed to help children learn about number factorization while being coached by an intelligent agent [42]. This affective user model has nodes such as: goal nodes (that model the objective during game playing, e.g., learn math, have fun, beat partner), action nodes (for both the player and the action), goal satisfaction nodes (to model the degree of satisfaction that each action causes), and emotional nodes that allow for the modeling of six of the 22 emotions described in the OCC theory of emotions [147], namely, joy/distress (states of the node "emotion for the game"), pride/shame (states of the node "emotion for self") and admiration/reproach (states of the node "emotion for agent"). In this case, links are established in the causal direction.
- Attitudes, perceptions. In [7], log-data is used to infer the student's hidden attitudes towards learning, learning gains, and perception of the system. To this end, the student model contains unobservable variables that measure if the students liked the system, learned, seriously tried to learn, wanted to finish quickly, wanted to challenge himself/herself, was concerned with getting help, had a fear of doing the wrong action, etc. Observable variables were: average of hints asked per problem, time between

attempts, average seconds per problem, etc. In that model also, links are pointed in the causal direction.

Going beyond modeling the student's knowledge can certainly provide a much better adaptation and therefore better performance in terms of learning gains. However, as explained in this section, the relationship between the cost of building, maintaining and effectively using such enriched student models must be evaluated on an individual basis, to decide whether each one is worth the gained improvement in performance of the system.

Building the Qualitative Model versus Using Learning Techniques. In the field of student modeling, many researchers have chosen to create structural models, with or without the help of domain experts. Good examples are Andes [40], HYDRIVE [132], Adele [67] and DT-Tutor [138]. However, the relationships between the variables of the model can also be learned from studying the data. To do so, the variables of interest, X_1, X_2, …, X_n for the domain being modeled, must first be identified. From that list of variables and from a dataset composed of the sets of their values (x_1, …, x_n), a structural learning algorithm infers the relationships between the variables, according to a desired confidence level. A key point of this approach is to choose a good value for the confidence level. If too low a value is selected, low data-evidence relationships will be inferred. If the value is too high, some important relationships can be missed. A possible solution is having domain experts fine-tune the inferred model.

There are few examples in which the structure has been discovered from studying the data, but the interested reader can find an excellent one in [122], which describe the techniques used and the strengths and weaknesses of this approach.

1.4.3 Development of the Quantitative Model

Once the qualitative part of the model has been defined or learned, the next step is to define the parameters, which in this case are the prior probability distributions of the root nodes and the conditional probability distributions for the rest of the nodes. This task has commonly been cited as one of the main difficulties when building a BN model. The options available are: a) knowledge engineering, i.e., having experts specify the probabilities; b) using pre-existent models to specify part of the probability distributions needed (canonical models, theoretical models, etc.); and c) learning the parameters from available data (cases). A combination of these alternatives can also be used.

Examples of systems in which the parameters have been estimated by experts include those mentioned in subsection 1.4.2 as examples of systems in which experts provided also the qualitative model. However, there are other systems where the structure has been specified or constrained by experts and the parameters are partially or totally derived from data [201], or adjusted to theoretical models, for example, models inspired by Item Response Theory [130] and [90]. Some approaches to simplify parameter specification in the context of student modeling have also been proposed [129]. Finally, an example in which both the structure and parameters have been discovered from data is [122].

Also, some modeling tricks can help to relieve the burden of specifying the numbers. An example of such a trick is grouping the parent nodes into related causal categories, which reduces the number of parameters needed. For a more comprehensive literature guide to tricks and techniques, see [60].

1.4.4 Building Student Models by Combining the Domain Expert's Knowledge and Learning Methods

More recently, researchers have begun to use mixed approaches, in which parts of the model are defined by experts while others are learned from data. Some examples are:

- In the student model for the medical domain by Suebnukarn et al [186], the structure has been specified by experts and then the parameters are mined from the data obtained in transcripts of problem-based learning sessions.
- Stacey et al construct their Bayesian student model based on misconceptions through a combination of elicitation from the domain experts and automated methods [180].
- When developing the knowledge student model for Prime Climb, [119] compares the performance of two different structures (one developed following teacher's suggestions, the other one inspired by causality) and for both of them the parameters are revealed in the data.
- Arroyo and Woolf define the structure of the network, taking into account the knowledge gained during a correlation analysis between variables, and then look for conditional probabilities in the data to evaluate the accuracy of the model by cross validation [7].
- In [29] and [42], Conati and colleagues use an iterative design process: the initial model is developed using the researcher's intuition or the expert's opinions, and then data from real users is collected and used to refine both the parameters and the structure, either by adding new nodes (such as "general exploration" or nodes that describe attitudes towards the intelligent agent, such as "wanting help") or by modifying the direction of the links (for example, between knowledge and exploration nodes or game events and goals).

1.5 Conclusions

A common feature of various adaptive Web systems is the application of user models (also known as profiles) to adapt the systems' behavior to individual users. User models represent the information about users that is essential to support the adaptation functionality of the systems. Adaptive Web systems have investigated a range of approaches to user modeling, exploring how to organize the storage for user information, how to populate it with user data, and how to maintain the current state of the user. The majority of modern adaptive Web systems use feature-based approach to represent and model information about the users. The competing stereotype-based approach, once popular in the pre-Web area of adaptive interfaces, has lost dominance but is still applied, especially in combination with the feature-based approaches.

The most popular features modeled and used by adaptive Web systems are user knowledge, interests, goals, background, individual traits, and context of work. Each individual adaptive system typically uses a subset of this list, as determined by the class of adaptive systems it belongs to and the adaptation needs of this class (Fig. 1.14). Web-based adaptive educational systems (AES) rely mostly on user knowledge and learning goals capitalizing on the modeling and representation techniques established in the field of Intelligent Tutoring Systems (ITS). Adaptive information systems and Web recommenders focus on modeling the user's interests and extend modeling approaches originally developed for adaptive information retrieval systems. Meanwhile, adaptive hypermedia systems attempt to represent and employ an even wider range of user features. In addition to user knowledge and interests, these systems frequently model user goals (following approaches developed in the field of adaptive interfaces), individual traits, and the context of user's work.

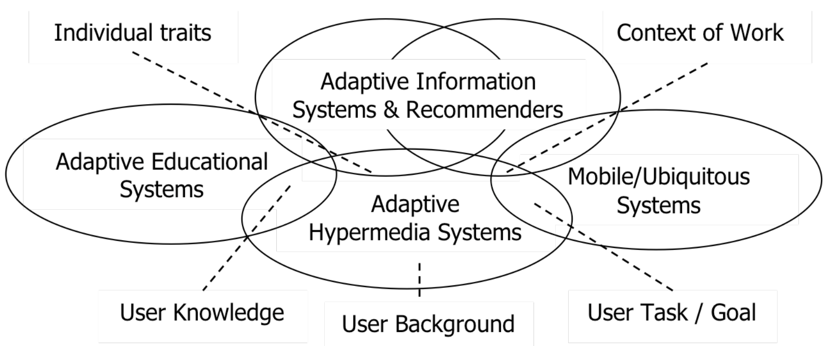


Fig. 1.14. User features typically modeled by different classes of adaptive Web systems

Overlay user modeling is currently the leading user modeling approach in AES and AHS. This approach was originally developed in the field of ITS to model user knowledge as an overlay of a concept-level domain model. Currently, the overlay approach has grown to include the modeling of knowledge, interests, goals, and some other features. In the area of adaptive information systems the concept-overlay approach to modeling user interests is now competing with the more traditional keyword-level profiling. This is one example of the convergence that has begun to blur the boundaries between different classes of adaptive Web systems.

One of the main advantages of using a formal method for student modeling is its robustness. Once this model behaves in a stable and theoretically-correct fashion, the evaluation of a system can be focused on other components (such as quality of the learning material, learning strategies used, or adaptation capabilities). The most commonly used formal reasoning technique for student modeling is currently the BN paradigm. Applications of other approaches (like Fuzzy Logic) are less frequent,. A very attractive potential use of BNs in the context of adaptive web-based applications would be to employ learning algorithms that would further shape on-the-fly improvements within the model itself. The future adaptation learning algorithm would then be able to process each user's interactive behavior information and simultaneously update the structure of the model. We hope to see this potential use developed in the very near future.

References

1. Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive tutors: Lessons learned. The Journal of the Learning Sciences 4, 2 (1995) 167-207
2. Anderson, J.R., Lebiere, C.: Atomic Components of Thought. Lawrence Erlbaum Associates, Hillsdale, NJ (1998)
3. Ardissono, L., Console, L., Torre, I.: An adaptive system for the personalised access to news. AI Communications 14 (2001) 129-147
4. Ardissono, L., Goy, A.: Tailoring the interaction with users in Web stores. User Modeling and User Adapted Interaction 10, 4 (2000) 251-303
5. Ardissono, L., Goy, A., Meo, R., Petrone, G., Console, L., Lesmo, L., Simone, C., Torasso, P.: A Configurable System for the Construction of Adaptive Virtual Stores. World Wide Web 2, 3 (1999) 143-159
6. Aroyo, L., Dicheva, D.A.: Concept-based approach to support learning in a Web-based support environment. In: Moore, J.D., Redfield, C.L., Johnson, W.L. (eds.) Proc. of AI-ED'2001. IOS Press (2001) 1-12
7. Arroyo, I., Woolf, B.P.: Inferring learning and attitudes from a Bayesian Network of log file data. In: Looi, C.-K., McCalla, G., Bredeweg, B., Breuker, J. (eds.) Proc. of 12th International Conference on Artificial Intelligence in Education, AIED'2005. IOS Press (2005) 33-40
8. Beaumont, I.: User modeling in the interactive anatomy tutoring system ANATOM-TUTOR. User Modeling and User-Adapted Interaction 4, 1 (1994) 21-45
9. Beck, J.: Engagement tracing: using response times to model student disengagement. In: Looi, C.-K., McCalla, G., Bredeweg, B., Breuker, J. (eds.) Proc. of 12th International Conference on Artificial Intelligence in Education, AIED'2005. IOS Press (2005) 88-95
10. Benyon, D., Murray, D.: Experience with adaptive interfaces. The Computer Journal 31, 5 (1988) 465-473
11. Berti, S., Mori, G., Paternò, F., Santoro, C.: An environment for designing and developing multi-platform interactive applications. In: Ardissono, L., Goy, A. (eds.) Proc. of HCITALY'2003. University of Turin (2003) 7-16
12. Billsus, D., Pazzani, M.J.: A learning agent for wireless news access. In: Lieberman, H. (ed.) Proc. of 2000 International Conference on Intelligent User Interfaces. ACM Press (2000) 94-97
13. Bloom, B.S.: Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain. David McKay Co Inc., New York (1956)

14.Boyle, C., Encarnacion, A.O.: MetaDoc: an adaptive hypertext reading system. User Modeling and User-Adapted Interaction 4, 1 (1994) 1-19
15.Brady, A., Conlan, O., Wade, V.: Dynamic Composition and Personalization of PDA-based eLearning – Personalized mLearning. In: Nall, J., Robson, R. (eds.) Proc. of World Conference on E-Learning, E-Learn 2004. AACE (2004) 234-242
16.Brailsford, T.J., Stewart, C.D., Zakaria, M.R., Moore, A.: Autonavigation, links, and narrative in an adaptive Web-based integrated learning environment. In: Proc. of The 11th International World Wide Web Conference. (2002)
17.Brajnik, G., Guida, G., Tasso, C.: User modeling in intelligent information retrieval. Information Processing and Management 23, 4 (1987) 305-320
18.Brusilovsky, P.: A framework for intelligent knowledge sequencing and task sequencing. In: Frasson, C., Gauthier, G., McCalla, G. (eds.) Proc. of Second International Conference on Intelligent Tutoring Systems, ITS'92. Springer-Verlag (1992) 499-506
19.Brusilovsky, P.: Adaptive hypermedia, an attempt to analyze and generalize. In: Brusilovsky, P., Kommers, P., Streitz, N. (eds.): Multimedia, Hypermedia, and Virtual Reality. Lecture Notes in Computer Science, Vol. 1077. Springer-Verlag, Berlin (1996) 288-304
20.Brusilovsky, P.: Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools. In: Murray, T., Blessing, S., Ainsworth, S. (eds.): Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software. Dordrecht, Kluwer (2003) 377-409
21.Brusilovsky, P.: Adaptive navigation support. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
22.Brusilovsky, P., Anderson, J.: ACT-R electronic bookshelf: An adaptive system for learning cognitive psychology on the Web. In: Maurer, H., Olson, R.G. (eds.) Proc. of WebNet'98, World Conference of the WWW, Internet, and Intranet. AACE (1998) 92-97
23.Brusilovsky, P., Cooper, D.W.: Domain, Task, and User Models for an Adaptive Hypermedia Performance Support System. In: Gil, Y., Leake, D.B. (eds.) Proc. of 2002 International Conference on Intelligent User Interfaces. ACM Press (2002) 23-30
24.Brusilovsky, P., Eklund, J., Schwarz, E.: Web-based education for all: A tool for developing adaptive courseware. In: Ashman, H., Thistewaite, P. (eds.) Proc. of Seventh International World Wide Web Conference. Vol. 30. Elsevier Science B. V. (1998) 291-300
25.Brusilovsky, P., Henze, N.: Open corpus adaptive educational hypermedia. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
26.Brusilovsky, P., Schwarz, E., Weber, G.: ELM-ART: An intelligent tutoring system on World Wide Web. In: Frasson, C., Gauthier, G., Lesgold, A. (eds.) Proc. of Third International Conference on Intelligent Tutoring Systems, ITS-96. Lecture Notes in Computer Science, Vol. 1086. Springer Verlag (1996) 261-269
27.Brusilovsky, P., Sosnovsky, S., Yudelson, M.: Ontology-based framework for user model interoperability in distributed learning environments. In: Richards, G. (ed.) Proc. of World Conference on E-Learning, E-Learn 2005. AACE (2005) 2851-2855
28.Bunt, A., Carenini, G., Conati, C.: Adaptive content presentation for the Web. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume

29. Bunt, A., Conati, C.: Probabilistic Student Modelling to Improve Exploratory Behaviour. User Modeling and User-Adapted Interaction 13, 3 (2003) 269-309
30. Carberry, S., de Rosis, F. (eds.): Proceedings of Workshop on Adapting the Interaction Style to Affective Factors hold in conjunction with User Modeling 2005, July 25, 2005. Edinburgh, UK (2005) available online at http://www.di.uniba.it/intint/UM05/WS-UM05.html
31. Carmona, C., Millán, E., Perez de la Cruz, J.-L., Trella, M., Conejo, R.: Introducing Prerequisite Relations in a Multi-layered Bayesian Student Model. In: Proc. of 10th International Conference UM'2005. Vol. 3538. Springer-Verlag (2005) 347-356
32. Carro, R.M., Pulido, E., Rodríguez, P.: Dynamic generation of adaptive Internet-based courses. Journal of Network and Computer Applications 22, 4 (1999) 249-257
33. Carver, C.A., Howard, R.A., Lavelle, E.: Enhancing student learning by incorporating student learning styles into adaptive hypermedia. In: Proc. of ED-MEDIA'96 - World Conference on Educational Multimedia and Hypermedia. AACE (1996) 118-123
34. Cawsey, A., Grasso, F., Paris, C.: Adaptive information for consumers of healthcare. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
35. Chen, S.Y., Macredie, R.D.: Cognitive styles and hypermedia navigation: Development of a learning model. Journal of the American Society for Information Science and Technology 53, 1 (2002) 3-15
36. Cheverst, K., Davies, N., Mitchell, K., Smith, P.: Providing tailored (context-aware) information to city visitors. In: Brusilovsky, P., Stock, O., Strapparava, C. (eds.) Proc. of Adaptive Hypermedia and Adaptive Web-based Systems. AH'2000. Lecture Notes in Computer Science, Vol. 1892. Springer-Verlag (2000) 73-85
37. Chin, D., Kobsa, A., Wahlster, W.: Modelling what the User Knows in UC. In: Loveland, D.W. (ed.) User Models in Dialog Systems. Symbolic Computation Series, Springer Verlag, Berlin (1989) 74-107
38. Cocea, M., Weibelzahl, S.: Can log files analysis estimate learners' level of motivation? In: Herder, E., Heckmann, D. (eds.) Proc. of 14th Workshop on Adaptivity and User Modeling in Interactive Systems, ABIS 2006. University of Hildesheim (2006) 32-35
39. Collins, J.A., Greer, J.E., Huang, S.H.: Adaptive Assessment Using Granularity Hierarchies and Bayesian Nets. In: Frasson, C., Gauthier, G., Lesgold, A. (eds.) Proc. of 3rd International Conference on Intelligent Tutoring Systems, ITS'96. Lecture Notes in Computer Science, Vol. 1086. Springer-Verlag (1996) 569-577
40. Conati, C., Gertner, A., VanLehn, K.: Using Bayesian Networks to Manage Uncertainty in Student Modeling. User Modeling and User-Adapted Interaction 12, 4 (2002) 371-417
41. Conati, C., Larkin, J., VanLehn, K.: A Computer Framework to Support Self-explanation. In: du Bolay, B., Mizoguchi, R. (eds.) Proc. of 8th World Conference on Artificial Intelligence in Education AIED'97. Knowledge and Media in Learning Systems, IOS Press (1997) 279-286
42. Conati, C., Maclaren, H.: Data-Driven Refinement of a Probabilistic Model of User Affect. In: Proc. of 10th International Conference UM'05. Lecture Notes in Computer Science, Vol. 3538. Springer-Verlag (2005) 40-49
43. Conlan, O., O'Keeffe, I., Tallon, S.: Combining adaptive hypermedia techniques and ontology reasoning to produce dynamic personalized news services. In: Wade, V., Ashman, H., Smyth, B. (eds.) Proc. of 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2006). Lecture Notes in Computer Science, Vol. 4018. Springer Verlag (2006) 81-90

44.Corbett, A., Anderson, J.: Student Modelling and Mastery Learning in a Computer-based Programming Tutor. In: Frasson, C., Gauthier, G., McCalla, G.I. (eds.) Proc. of 2nd International Conference on Intelligent Tutoring Systems, ITS'92. Lecture Notes in Computer Science, Vol. 608. Springer-Verlag (1992) 413-420
45.Corbett, A., Anderson, J.R., O'Brien, A.T.: Student Modeling in the ACT Programming Tutor. In: Cognitively Diagnostic Assessment. Erlbaum, Hillsdale, NJ (1995) 19-41
46.Cotter, P., Smyth, B.: WAP-ing the Web: Content personalization for WAP-enabled devices. In: Brusilovsky, P., Stock, O., Strapparava, C. (eds.) Proc. of Adaptive Hypermedia and Adaptive Web-based systens. Lecture Notes in Computer Science, Vol. 1892. Springer-Verlag (2000) 98-108
47.Dagger, D., Conlan, O., Wade, V.P.: An architecture for candidacy in adaptive eLearning systems to facilitate the reuse of learning Resources. In: Rossett, A. (ed.) Proc. of World Conference on E-Learning, E-Learn 2003. AACE (2003) 49-56
48.De Bra, P., Aerts, A., Rousseau, B.: Concept Relationship Types for AHA! 2.0. In: Driscoll, M., Reeves, T.C. (eds.) Proc. of World Conference on E-Learning, E-Learn 2002. AACE (2002) 1386-1389
49.De Bra, P., Aerts, A., Smits, D., Stash, N.: AHA! Version 2.0: More Adaptation Flexibility for Authors. In: Driscoll, M., Reeves, T.C. (eds.) Proc. of World Conference on E-Learning, E-Learn 2002. AACE (2002) 240-246
50.De Bra, P., Calvi, L.: AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia 4 (1998) 115-139
51.De Bra, P., Ruiter, J.-P.: AHA! Adaptive hypermedia for all. In: Fowler, W., Hasebrook, J. (eds.) Proc. of WebNet'2001, World Conference of the WWW and Internet. AACE (2001) 262-268
52.De Bra, P.M.E.: Teaching Hypertext and Hypermedia through the Web. Journal of Universal Computer Science 2, 12 (1996) 797-804
53.de Rosis, F., De Carolis, B., Pizzutilo, S.: User tailored hypermedia explanations. In: Brusilovsky, P., Beaumont, I. (eds.) Proc. of Workshop Adaptive Hypertext and Hypermedia at Fourth International Conference on User Modeling. (1994) http://wwwis.win.tue.nl/ah94/deRosis.html
54.Díaz, A., Gervás, P.: Personalisation in news delivery systems: Item summarization and multi-tier item selection using relevance feedback. Web Intelligence and Agent Systems 3, 3 (2005) 135-154
55.Dichev, C., Dicheva, D., Aroyo, L.: Using Topic Maps for Web-based Education. Advanced Technology for Learning 1, 1 (2004) 1-7
56.Dolog, P., Henze, N., Nejdl, W., Sintek, M.: Personalization in distributed e-learning environments. In: Proc. of The Thirteenth International World Wide Web Conference, WWW 2004 (Alternate track papers and posters). ACM Press (2004) 161-169
57.Dolog, P., Nejdl, W.: Semantic Web Technologies for the Adaptive Web. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
58.Dolog, P., Schäfer, M.: Learner Modeling on the Semantic Web. In: Proc. of PerSWeb'05, Workshop on Personalization on the Semantic Web at 10th International User Modeling Conference. (2005) http://www.win.tue.nl/persweb/Camera-ready/6-Dolog-full.pdf
59.Dowling, C.E., Hockemeyer, C., Ludwig, A.H.: Adaptive Asessment and Training Using the Neighbourhood of Knowledge States. In: Frasson, C., Gauthier, G., Lesgold, A. (eds.) Proc. of

3rd International Conference on Intelligent Tutoring Systems, ITS'96. Lecture Notes in Computer Science, Vol. 1086. Springer-Verlag (1996) 578-585
60. Druzdzel, M., van der Gaag, L.C.: Building Probabilistic Networks: Where Do the Numbers Come From? - Guest editors' introduction. IEEE Transactions on Knowledge and Data Engineering 12, 4 (2000) 481-486
61. Dufresne, A., Turcotte, S.: Cognitive style and its implications for navigation strategies. In: du Boulay, B., Mizoguchi, R. (eds.) Proc. of AI-ED'97, 8th World Conference on Artificial Intelligence in Education. IOS (1997) 287-293
62. Eisenstein, J., Vanderdonckt, J., Puerta, A.: Applying model-based techniques to the development of UIs for mobile computers. In: Proc. of 6th International Conference on Intelligent User Interfaces. ACM Press (2001) 69-76
63. Encarnação, L.M.: Multi-level user support through adaptive hypermedia: A highly application-independent help component. In: Moore, J., Edmonds, E., Puerta, A. (eds.) Proc. of 1997 International Conference on Intelligent User Interfaces. ACM (1997) 187-194
64. Felder, R.: Learning and teaching styles. Journal of Engineering Education 78, 7 (1988) 674-681
65. Fink, J., Kobsa, A., Nill, A.: Adaptable and adaptive information provision for all users, including disabled and elderly people. The New Review of Hypermedia and Multimedia 4 (1998) 163-188
66. Fischer, G.: User modeling in human-computer interaction. User Modeling and User Adapted Interaction 11, 1-2 (2001) 65-86
67. Ganeshan, R., Johnson, W., Shaw, E., Wood, B.P.: Tutoring Diagnostic Problem Solving. In: Proc. of 7th International Conference on Intelligent Tutoring Systems, ITS'2004. Vol. 1839. Springer-Verlag (2000) 33-42
68. García, F., Amandi, A., Schiaffinoa, S., Campoa, M.: Evaluating Bayesian networks' precision for detecting students' learning styles. Computers & Education (2006) In press
69. Garlatti, S., Iksal, S.: Context filtering and spacial filtering in an adaptive information system. In: Brusilovsky, P., Stock, O., Strapparava, C. (eds.) Proc. of Adaptive Hypermedia and Adaptive Web-based systens. Lecture Notes in Computer Science, Vol. 1892. Springer-Verlag (2000) 315-318
70. Garlatti, S., Iksal, S., Kervella, P.: Adaptive on-line information system by means of a task model and spatial views. In: Brusilovsky, P., Bra, P.D. (eds.) Proc. of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web. (1999) 59-66, also available at http://wwwis.win.tue.nl/asum99/garlatti/garlatti.html
71. Gates, K.F., Lawhead, P.B., Wilkins, D.E.: Toward an adaptive WWW: a case study in customized hypermedia. New Review of Multimedia and Hypermedia 4 (1998) 89-113
72. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User profiles for personalized information access. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
73. Gilbert, J.E., Han, C.Y.: Arthur: Adapting Instruction to Accommodate Learning Style. In: Bra, P.D., Leggett, J. (eds.) Proc. of WebNet'99, World Conference of the WWW and Internet. AACE (1999) 433-438
74. Goldstein, I.P.: The genetic graph: a representation for the evolution of procedural knowledge. In: Sleeman, D.H., Brown, J.S. (eds.): Intelligent tutoring systems. Academic press, London (1982) 51-77

75.Gonschorek, M., Herzog, C.: Using hypertext for an adaptive helpsystem in an intelligent tutoring system. In: Greer, J. (ed.) Proc. of AI-ED'95, 7th World Conference on Artificial Intelligence in Education. AACE (1995) 274-281
76.Goodman, B.A., Litman, D.J.: On the interaction between plan recognition and intelligent interfaces. User Modeling and User-Adapted Interaction 2, 1 (1992) 83-115
77.Goren-Bar, D., Graziola, I., Pianesi, F., Zancanaro, M.: The influence of personality factors on visitor attitudes towards adaptivity dimensions for mobile museum guides. User Modeling and User Adapted Interaction 16, 1 (2005) 31-62
78.Grunst, G.: Adaptive hypermedia for support systems. In: Schneider-Hufschmidt, M., Kühme, T., Malinowski, U. (eds.): Adaptive user interfaces: Principles and practice. North-Holland, Amsterdam (1993) 269-283
79.Heckerman, D.: A Tutorial on Learning with Bayesian Networks, Technical Report No. MSR-TR-95-06, Microsoft Research Advanced Technology Division (1995)
80.Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: Gumo - The General User Model Ontology. In: Ardissono, L., Brna, P., Mitrovic, A. (eds.) Proc. of 10th International User Modeling Conference. Lecture Notes in Artificial Intelligence, Vol. 3538. Springer Verlag (2005) 428-432
81.Henze, N., Naceur, K., Nejdl, W., Wolpers, M.: Adaptive hyperbooks for constructivist teaching. Künstliche Intelligenz, 4 (1999) 26-31
82.Henze, N., Nejdl, W.: Student modeling for KBS Hyperbook system using Bayesian networks, Technical report, University of Hannover (1999) available online at http://www.kbs.uni-hannover.de/paper/99/adaptivity.html
83.Henze, N., Nejdl, W.: Adaptation in open corpus hypermedia. International Journal of Artificial Intelligence in Education 12, 4 (2001) 325-350
84.Hirashima, T., Hachiya, K., Kashihara, A., Toyoda, J.i.: Information filtering using user's context on browsing in hypertext. User Modeling and User Adapted Interaction 7, 4 (1997) 239-256
85.Hirashima, T., Matsuda, N., Nomoto, T., Toyoda, J.i.: Context-sensitive filtering for browing in hypertext. In: Proc. of International Conference on Intelligent User Interfaces, IUI'98. ACM Press (1998) 21-28
86.Hockemeyer, C., Held, T., Albert, D.: RATH - A relational adaptive tutoring hypertext WWW-environment based on knowledge space theory. In: Alvegård, C. (ed.) Proc. of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering. (1998) 417-423
87.Hollink, V., Someren, M.v., Hage, S.t.: Discovering stages in web navigation. In: Ardissono, L., Brna, P., Mitrovic, A. (eds.) Proc. of 10th International User Modeling Conference. Lecture Notes in Artificial Intelligence, Vol. 3538. Springer Verlag (2005) 473-482
88.Höök, K., Karlgren, J., Wærn, A., Dahlbäck, N., Jansson, C.G., Karlgren, K., Lemaire, B.: A glass box approach to adaptive hypermedia. User Modeling and User-Adapted Interaction 6, 2-3 (1996) 157-184
89.Horvitz, E.J., Breese, J.S., Henrion, M.: Decision Theory in Expert Systems and Artificial Intelligence. International Journal of Approximate Reasoning 2 (1988) 247-302
90.Jameson, A.: Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues. User Modeling and User-Adapted Interaction 5 (1996) 193-251
91.Jameson, A.: Modeling both the context and the user. Personal Technologies 5, 1 (2001) 29-33
92.Jameson, A., Smyth, B.: Recommendation to groups In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume

93. Jantke, K.P., Memmel, M., Rostanin, O., Rudolf, B.: Media and service integration for professional e-learning. In: Nall, J., Robson, R. (eds.) Proc. of World Conference on E-Learning, E-Learn 2004. AACE (2004) 725-732
94. Jin, X., Zhou, Y., Mobasher, B.: Task-Oriented Web User Modeling for Recommendation. In: Ardissono, L., Brna, P., Mitrovic, A. (eds.) Proc. of 10th International User Modeling Conference. Lecture Notes in Artificial Intelligence, Vol. 3538. Springer Verlag (2005) 109-118
95. Joerding, T.: A temporary user modeling approach for adaptive shopping on the Web. In: Brusilovsky, P., De Bra, P. (eds.) Proc. of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web. (1999) 75-79, also available at http://wwwis.win.tue.nl/asum99/joerding/joerding.html
96. Jokela, S., Turnpeinen, M., Kurki, T., Savia, E., Sulonen, R.: The Role of Structured Content in a Personalised News Service. In: Proc. of 34th Hawaii International Conference on System Sciences. (2001) 1-10
97. Kaplan, C., Fenwick, J., Chen, J.: Adaptive hypertext navigation based on user goals and context. User Modeling and User-Adapted Interaction 3, 3 (1993) 193-220
98. Katz, S., Lesgold, A., Eggan, G., Gordin, M.: Modelling the student in SHERLOCK II. In: Greer, J.E., McCalla, G. (eds.): Student Modelling: The Key to Individualized Knowledge-Based Instruction. Series F: Computer and Systems Sciences. NATO ASI Series, Springer Verlag, Berlin Heidelberg (1994) 99-125
99. Kavcic, A.: Fuzzy user modeling for adaptation in educational hypermedia. IEEE Transactions on Systems, Man, and Cybernetics 34, 4 (2004) 439-449
100. Kawai, K., Mizoguchi, R., Kakusho, O., Toyoda, J.: A framework for ICAI systems based on inductive inference and logic programming. New Generation Computing 5 (1987) 115-129
101. Kay, J.: Lies, damned lies and stereotypes: pragmatic approximations of users. In: Kobsa, A., Litman, D. (eds.) Proc. of Fourth International Conference on User Modeling. MITRE (1994) 175-184
102. Kay, J., Kummerfeld, R.J.: An individualised course for the C programming language. In: Proc. of Second International WWW Conference. (1994) http://www.cs.usyd.edu.au/~bob/kay-kummerfeld.html
103. Kay, J., Lum, A.: Ontologies for Scrutable Learner Modeling in Adaptive E-Learning. In: Aroyo, L., Tasso, C. (eds.) Proc. of Workshop on Application of Semantic Web Technologies for Adaptive Educational Hypermedia at the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004). Technische University Eindhoven (2004) 292-301
104. Kim, K.S., Allen, B.: Cognitive and Task Influences on Web Searching Behavior. Journal of the American Society for Information Science and Technology 53, 2 (2002) 109-119
105. Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Johan Hjelm, Butler, M.H., Tran, L.: Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation 15 January 2004. (2004) http://www.w3.org/TR/CCPP-struct-vocab/
106. Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A.: Intelligent tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education 8 (1997) 30-43
107. Korfhage, R.R.: Information storage and retrieval. Wiley Computer Publishing, N.Y. (1997)
108. Kosba, E., Dimitrova, V., Boyle, R.: Using Fuzzy Techniques to Model Students in Web-Based Learning Environments. International Journal of Artificial Intelligence Tools 13, 2 (2004) 279-297

109. Krüger, A., Baus, J., Heckmann, D., Kruppa, M., Wasinger, R.: Adaptive mobile guides. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
110. Kumar, A.N.: A Scalable Solution for Adaptive Problem Sequencing and its Evaluation. In: Wade, V., Ashman, H., Smyth, B. (eds.) Proc. of 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2006). Lecture Notes in Computer Science, Vol. 4018. Springer Verlag (2006) 161-171
111. Laroussi, M., Benahmed, M.: Providing an adaptive learning through the Web case of CAMELEON: Computer Aided MEdium for LEarning on Networks. In: Alvegård, C. (ed.) Proc. of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering. (1998) 411-416
112. Lee, K.B., Grice, R.A.: An Adaptive Viewing Application for the Web on Personal Digital Assistants. In: Proc. of ACM SIGDOC'03. IEEE (2003) 125-132
113. Liu, Y., Ginther, D.: Cognitive styles and distance education. Online Journal of Distance Learning Administration 2, 3 (1999) http://www.westga.edu/~distance/liu23.html
114. Locke, J.: Microsoft Bayesian Networks. Basics of Knowledge Engineering. Microsoft Support Technology (1999) http://freelock.com/files/KE.pdf
115. Lopez, J.F., Szekely, P.: Web page adaptation for universal access. In: Stephanidis, C. (ed.) Proc. of 1st International Conference on Universal Access in Human-Computer Interaction. Lawrence Erlbaum Associates (2001) 690-694
116. López, J.M., Millán, E., Pérez-de-la-Cruz, J.-L., Triguero, F.: ILESA: a Web-based Intelligent Learning Environment for the Simplex Algorithm. In: Alvegård, C. (ed.) Proc. of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering. (1998) 399-406
117. Lundgren-Cayrol, K., Paquette, G., Miara, A., Bergeron, F., Rivard, J., Rosca, I.: Explor@ Advisory Agent: Tracing the Student's Trail. In: Fowler, W., Hasebrook, J. (eds.) Proc. of WebNet'2001, World Conference of the WWW and Internet. AACE (2001) 802-808
118. Magnini, B., Strapparava, C.: Improving user modeling with content-based techniques. In: Bauer, M., Gmytrasiewicz, P.J., Vassileva, J. (eds.) Proc. of 8th International Conference on User Modeling, UM 2001. Lecture Notes on Artificial Intelligence, Vol. 2109. Springer-Verlag (2001) 74-83
119. Manske, M., Conati, C.: Modelling Learning in an Educational Game. In: Proc. of 12th World Conference of Artificial Intelligence and Education AIED'05. IOS Press (2005) 411-419
120. Masthoff, J.: Towards an authoring coach for adaptive Web-based instruction. In: De Bra, P., Brusilovsky, P., Conejo, R. (eds.) Proc. of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002). Lecture Notes in Computer Science, Vol. 2347. (2002) 415-418
121. Mathé, N., Chen, J.: User-centered indexing for adaptive information access. User Modeling and User-Adapted Interaction 6, 2-3 (1996) 225-261
122. Mayo, M., Mitrovic, A.: Optimising ITS behaviour with Bayesian networks and decision theory. International Journal of Artificial Intelligence in Education 12 (2001) 124-153
123. McArthur, D., Stasz, C., Hotta, J., Peter, O., Burdorf, C.: Skill-oriented task sequencing in an intelligent tutor for basic algebra. Instructional Science 17, 4 (1988) 281-307
124. McCalla, G., Bunt, R.B., Harms, J.J.: The design of the SCENT automated advisor. Computational Intelligence 2, 2 (1986) 76-91

125. Micarelli, A., Gasparetti, F., Sciarrone, F., Gauch, S.: Personalized search on the World Wide Web. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
126. Micarelli, A., Sciarrone, F.: A case-based system for adaptive hypermedia navigation. In: Smith, I., Faltings, B. (eds.): Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin (1996) 266-279
127. Micarelli, A., Sciarrone, F.: Anatomy and empirical evaluation of an adaptive Web-based information filtering system. User Modeling and User Adapted Interaction 14, 159-200 (2004)
128. Middleton, S.E., Shadbolt, N.R., De Roure, D.C.: Ontological User Profiling in Recommender Systems. ACM Transactions on Information Systems 22, 1 (2004) 54-88
129. Millán, E., Agosta, J.M., Perez de la Cruz, J.-L.: Bayesian Student Modelling and the Problem of Parameter Specification. British Journal of Educational Technology 32, 2 (2001) 171-181
130. Millán, E., Perez de la Cruz, J.-L.: A Bayesian Diagnostic Algorithm for Student Modeling. User Modeling and User-Adapted Interaction 12 (2002) 281-330
131. Millán, E., Perez de la Cruz, J.-L., García, F.: Dynamic versus Static Student Models Based on Bayesian Networks: An Empirical Study. In: Proc. of 7th International Conference KES'2003. Lecture Notes in Computer Science, Vol. 2774. Springer-Verlag (2003) 1337-1344
132. Mislevy, R., Gitomer, D.H.: The Role of Probability-Based Inference in an Intelligent Tutoring System. User Modeling and User-Adapted Interaction 5, 3-4 (1996) 253-282
133. Mitchell, T., Chen, S.Y., Macredie, R.: Adapting Hypermedia to cognitive styles: Is it necessary? In: Proc. of Workshop on Individual Differences in Adaptive Hypermedia at the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems. (2004) http://www.dcs.bbk.ac.uk/~gmagoulas/AH2004_Workshop/Proceedings.htm
134. Mitrovic, A.: An Intellignet SQL Tutor on the Web. International Journal of Artificial Intelligence in Education 13, 2-4 (2003) 173-197
135. Mobasher, B.: Data mining for Web personalization. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
136. Muldner, K., Conati, C.: Using Similarity to Infer Meta-Cognitive Behaviours During Analogical Problem Solving. In: Proc. of 10th International Conference UM'05. Vol. 3538. Springer-Verlag (2005) 134-143
137. Müller, C., Großmann-Hutter, B., Jameson, A., Rummer, R., Wittig, F.: Recognizing time pressure and cognitive load on the basis of speech: An Experimental study. In: Bauer, M., Gmytrasiewicz, P.J., Vassileva, J. (eds.) Proc. of 8th International Conference on User Modeling, UM 2001. Lecture Notes on Artificial Intelligence, Vol. 2109. Springer-Verlag (2001) 24-33
138. Murray, R.C., VanLehn, K., Mostow, J.: Looking ahead to select tutorial actions: A decision-theoretic approach. International Journal of Artificial Intelligence in Education 14, 3-4 (2004) 235-279
139. Murray, W.: An Easily Implemented, Linear-time Algorithm for Bayesian Student Modeling in Multi-level Trees. In: Lajoie, S., Vivet, M. (eds.) Proc. of 9th World Conference of Artificial Intelligence and Education AIED'99. IOS Press (1999) 413-420
140. Neapolitan, R.: Probabilistic Reasoning in Expert Systems: Theory and Algorithms. John Wiley & Sons, New York (1990)

141. Neumann, G., Zirvas, J.: SKILL - A scallable internet-based teaching and learning system. In: Maurer, H., Olson, R.G. (eds.) Proc. of WebNet'98, World Conference of the WWW, Internet, and Intranet. AACE (1998) 688-693
142. Not, E., Petrelli, D., Sarini, M., Stock, O., Strapparava, C., Zancanaro, M.: Hypernavigation in the physical space: adapting presentation to the user and to the situational context. New Review of Multimedia and Hypermedia 4 (1998) 33-45
143. Oberlander, J., O'Donell, M., Mellish, C., Knott, A.: Conversation in the museum: experiments in dynamic hypermedia with the intelligent labeling explorer. The New Review of Multimedia and Hypermedia 4 (1998) 11-32
144. Ohlsson, S.: Constraint-based student modeling. Journal of Artificial Intelligence in Education 3, 4 (1992) 429-447
145. Okazaki, Y., Watanabe, K., Kondo, H.: An Implementation of the WWW Based ITS for Guiding Differential Calculations. In: Brusilovsky, P., Nakabayashi, K., Ritter, S. (eds.) Proc. of Workshop "Intelligent Educational Systems on the World Wide Web" at 8th World Conference on Artificial Intelligence in Education. (1997) 18-25, also available at http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Okazaki/Okazaki.html
146. Ong, E., Tay, A.-H., Ong, C.-K., Chan, S.-K.: Personalising Information Assets in Collaborative Learning Environments. In: Looi, C.-K., McCalla, G., Bredeweg, B., Breuker, J. (eds.) Proc. of 12th International Conference on Artificial Intelligence in Education, AIED'2005. IOS Press (2005) 523-530
147. Ortony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press, (1988)
148. Papanikolaou, K.A., Grigoriadou, M., Kornilakis, H., Magoulas, G.D.: Personalising the interaction in a Web-based Educational Hypermedia System: the case of INSPIRE. User Modeling and User Adapted Interaction 13, 3 (2003) 213-267
149. Paris, C., Wan, S., Wilkinson, R., Wu, M.: Generating personal travel guides - and who wants them? In: Bauer, M., Gmytrasiewicz, P.J., Vassileva, J. (eds.) Proc. of 8th International Conference on User Modeling, UM 2001. Lecture Notes on Artificial Intelligence, Vol. 2109. Springer-Verlag (2001) 251-253
150. Pask, G.: A fresh look at cognition and the individual. International Journal on the Man-Machine Studies 4 (1972) 211-216
151. Paternò, F., Paganelli, L.: Intelligent analysis of user interactions with Web applications. In: Gil, Y., Leake, D.B. (eds.) Proc. of 2002 International Conference on Intelligent User Interfaces. ACM Press (2002) 111-118
152. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
153. Pearl, J.: Probabilistic Reasoning in Expert Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc, San Francisco (1988)
154. Pentland, A.: Socially Aware Computation and Communication. Computer 38, 3 (2005) 33 - 40
155. Pérez, T., Gutiérrez, J., Lopistéguy, P.: An adaptive hypermedia system. In: Greer, J. (ed.) Proc. of AI-ED'95, 7th World Conference on Artificial Intelligence in Education. AACE (1995) 351-358
156. Picard, R.W.: Affective Computing. MIT Press, Cambridge, MA (1997)

157. Pilar da Silva, D., Durm, R.V., Duval, E., Olivié, H.: Concepts and documents for adaptive educational hypermedia: a model and a prototype. In: Brusilovsky, P., De Bra, P. (eds.) Proc. of Second Adaptive Hypertext and Hypermedia Workshop at the Ninth ACM International Hypertext Conference Hypertext'98. Eindhoven University of Technology (1998) 35-43, also available as http://wwwis.win.tue.nl/ah98/Pilar/Pilar.html
158. Polson, M.C., Richardson, J.J. (eds.): Foundations of intelligent tutoring systems. Lawrence Erlbaum Associates, Hillsdale (1988)
159. Prendinger, H., Mori, J., Ishizuka, M.: Recognizing, modeling, and responding to user affective states. In: Ardissono, L., Brna, P., Mitrovic, A. (eds.) Proc. of 10th International User Modeling Conference. Lecture Notes in Artificial Intelligence, Vol. 3538. Springer Verlag (2005) 60-69
160. Prentzas, J., Hatzilygeroudis, I., Garofalakis, J.: A Web-based intelligent tutoring systems using hybrid rules as its representation basis. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) Proc. of 6th International Conference on Intelligent Tutoring Systems (ITS'2002). Lecture Notes in Computer Science, Vol. 2363. Springer-Verlag (2002) 119-128
161. Read, T., Bárcena, E., Barros, B., Verdejo, F.: I-PETER: Modelling Personalised Diagnosis and Material Selection for an Online English Course. In: Proc. of 8th Ibero-American Conference IBERAMIA'2002. Lecture Notes in Computer Science, Vol. 2527. Springer-Verlag (2002) 734-744
162. Reye, J.: Two-phase Updating of Student Models Based on Dynamic Belief Networks. In: Proc. of 4th International Conference on Intelligent Tutoring Systems, ITS'98. Lecture Notes in Computer Science, Vol. 1452. Springer-Verlag (1998) 6-15
163. Rich, E.: Building and Exploiting User Models. In: Proc. of Sixth International Joint Conference on Artificial Intelligence. (1979) 720-722
164. Rich, E.A.: Stereotypes and user modeling. In: Kobsa, A., Wahlster, W. (eds.): User models in dialog systems. Vol. 18. Springer-Verlag, Berlin (1989) 35-51
165. Riding, R., Rayner, S.: Cognitive Styles and Learning Strategies: Understanding Style Differences in Learning and Behavior. David Fulton Publisher, London (1998)
166. Rist, T.: A perspective on intelligent information interfaces for mobile users. In: Smith, M., Salvendy, G., Harris, D., Koubek, R.J. (eds.) Proc. of 9th International Conference on Human-Computer Interaction, HCI International'2001. Vol. 1. Lawrence Erlbaum Associates (2001) 154-158
167. Ritter, S.: PAT Online: A Model-tracing tutor on the World-wide Web. In: Brusilovsky, P., Nakabayashi, K., Ritter, S. (eds.) Proc. of Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education. ISIR (1997) 11-17, also available at http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Ritter/Ritter.html
168. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, (1995)
169. Sanrach, C., Grandbastien, M.: ECSAIWeb: A Web-based authoring system to create adaptive learning systems. In: Brusilovsky, P., Stock, O., Strapparava, C. (eds.) Proc. of Adaptive Hypermedia and Adaptive Web-based Systems, AH2000. Lecture Notes in Computer Science, Vol. 1892. Springer-Verlag (2000) 214-226
170. Santos Jr., E., Nguyen, H., Zhao, Q., Wang, H.: User modeling for intent prediction in information analysis. In: Proc. of 47th Annual Meeting for the Human Factors and Ergonomics Society (HFES-03). (2003) 1034-1038
171. Sarini, M., Strapparava, C.: Building a User Model for a Museum Exploration and Information-Providing Adaptive System. In: Brusilovsky, P., De Bra, P. (eds.) Proc. of Second

Adaptive Hypertext and Hypermedia Workshop at the Ninth ACM International Hypertext Conference Hypertext'98. (1998) 63-68, also available at http://wwwis.win.tue.nl/ah98/Sarini/Sarini.html
172. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
173. Schmidt, A., Beigl, M., Gellersen, H.-W.: There is more to context than location. Computers and Graphics 23, 6 (1999) 893-901
174. Schneider-Hufschmidt, M., Kühme, T., Malinowski, U. (eds.): Adaptive user interfaces: Principles and practice. Human Factors in Information Technology, North-Holland, Amsterdam (1993)
175. Sleeman, D.H.: UMFE: a user modeling front end system. International Journal on the Man-Machine Studies 23 (1985) 71-88
176. Sosnovsky, S., Brusilovsky, P.: Layered Evaluation of Topic-Based Adaptation to Student Knowledge. In: Proc. of Fourth Workshop on the Evaluation of Adaptive Systems at 10th International User Modeling Conference, UM 2005. (2005) 47-56
177. Specht, M., Klemke, R.: ALE - Adaptive Learning Environment. In: Fowler, W., Hasebrook, J. (eds.) Proc. of WebNet'2001, World Conference of the WWW and Internet. AACE (2001) 1155-1160
178. Specht, M., Kobsa, A.: Interaction of domain expertise and interface design in adaptive educational hypermedia. In: Brusilovsky, P., De Bra, P. (eds.) Proc. of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web. (1999) 89-93
179. Specht, M., Oppermann, R.: ACE - Adaptive Courseware Environment. The New Review of Hypermedia and Multimedia 4 (1998) 141-161
180. Stacey, K.P., Sonenberg, L., Nicholson, A., Boneh, T., Steinle, V.: A Teaching Model Exploiting Cognitive Conflict Driven by a Bayesian Network. In: Brusilovsky, P., Corbett, A., Rosis, F.d. (eds.) Proc. of 9th International User Modeling Conference. Lecture Notes in Computer Science, Vol. 2702. Springer-Verlag (2003) 352-362
181. Stash, N., Cristea, A., De Bra, P.: Authoring of learning styles in adaptive hypermedia: Problems and solutions. In: Proc. of The 13th International World Wide Web Conference (Alternate track papers and posters). ACM Press (2004) 114-123
182. Stathacopoulou, S., Magoulas, G., Grigoriadou, M., Samarakou, M.: Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis. Information Sciences 170, 2-4 (2005) 273-307
183. Steinacker, A., Faatz, A., Seeberg, C., Rimac, I., Hörmann, S., Saddik, A.E., Steinmetz, R.: MediBook: Combining semantic networks with metadata for learning resources to build a Web based learning system. In: Proc. of ED-MEDIA'2001 - World Conference on Educational Multimedia, Hypermedia and Telecommunications. AACE (2001) 1790-1795
184. Steinacker, A., Seeberg, C., Rechenberger, K., Fischer, S., Steinmetz, R.: Dynamically generated tables of contents as guided tours in adaptive hypermedia systems. In: Kommers, P., Richards, G. (eds.) Proc. of ED-MEDIA/ED-TELECOM'99 - 11th World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications. AACE (1999) 640-645
185. Stojanovic, L., Staab, S., Studer, R.: eLearning based on the Semantic Web. In: Fowler, W., Hasebrook, J. (eds.) Proc. of WebNet'2001, World Conference of the WWW and Internet. AACE (2001) 1774-1783

186. Suebnukarn, S., Haddawy, P.: Modeling Individual and Collaborative Problem Solving in Medical Problem-Based Learning. In: Ardissono, L., Brna, P., Mitrovic, A. (eds.) Proc. of 10th International User Modeling Conference, UM'2005. Lecture Notes in Artificial Intelligence, Vol. 3538. Springer-Verlag (2005) 377-386
187. Tanudjaja, F., Mui, L.: Persona: A contextualized and personalized Web search. In: Proc. of 35th Hawaii International Conference on System Sciences. IEEE (2002) 1232-1240
188. Tarpin-Bernard, F., Habieb-Mammar, H.: Modeling elementary cognitive abilities for adaptive hypermedia presentation. User Modeling and User Adapted Interaction 15, 5 (2005) 459-495
189. Trella, M., Carmona, C., Conejo, R.: MEDEA: an Open Service-Based Learning Platform for Developing Intelligent Educational Systems for the Web. In: Proc. of Workshop on Adaptive Systems for Web-based Education at 12th International Conference on Artificial Intelligence in Education, AIED'2005. IOS Press (2005) 27-34
190. Triantafillou, E., Pomportis, A., Demetriadis, S.: The design and the formative evaluation of an adaptive educational system based on cognitive styles. Computers and Education (2003) 87-103
191. Triantafillou, E., Pomportis, A., Demetriadis, S., Georgiadou, E.: The value of adaptivity based on cognitive style: an empirical study. British Journal of Educational Technology 35, 1 (2004) 95–106
192. Tsiriga, V., Virvou, M.: Modelling the Student to Individualise Tutoring in a Web-Based ICALL. International Journal of Continuing Engineering Education and Lifelong Learning 13, 3-4 (2003) 350-365
193. Ueno, M.: Intelligent LMS with an agent that learns from log data. In: Richards, G. (ed.) Proc. of World Conference on E-Learning, E-Learn 2005. AACE (2005) 2068-2074
194. VanLehn, K.: Student models. In: Polson, M.C., Richardson, J.J. (eds.): Foundations of intelligent tutoring systems. Lawrence Erlbaum Associates, Hillsdale (1988) 55-78
195. VanLehn, K., Martin, J.: Evaluation of an assessment system based on Bayesian Student Modeling. International Journal of Artificial Intelligence in Education 8, 2 (1998) 179-221
196. VanLehn, K., Niu, Z., Siler, S., Gertner, A.S.: Student Modeling from Conventional Test Data: A Bayesian Approach Without Priors. In: Goettl, B., Redfield, C.L., Halff, H.M., Shute, V.J. (eds.) Proc. of 4th International Conference on Intelligent Tutoring Systems, ITS'98. Lecture Notes in Computer Science, Vol. 1452. Springer-Verlag (1998) 434-443
197. Vassileva, J.: An architecture and methodology for creating a domain-independent, plan-based intelligent tutoring system. Educational and Training Technology International 27, 4 (1990) 386-397
198. Vassileva, J.: A task-centered approach for user modeling in a hypermedia office documentation system. User Modeling and User-Adapted Interaction 6, 2-3 (1996) 185-224
199. Vassileva, J.: DCG + GTE: Dynamic Courseware Generation with Teaching Expertise. Instructional Science 26, 3/4 (1998) 317-332
200. Vassileva, J., McCalla, G., Greer, J.: Multi-Agent Multi-User Modeling in I-Help. User Modeling and User-Adapted Interaction 12 (2003) 179-210
201. Vomlel, J.: Bayesian Networks in Educational Testing. International Journal of Uncertainty, Fuzziness and Knowledge-Based System 12 (2004) 83-100
202. W3C: Device Independence: Access to a Unified Web from Any Device in Any Context by Anyone. World Wide Web Consortium (2006) http://www.w3.org/2001/di/
203. Weber, G., Brusilovsky, P.: ELM-ART: An adaptive versatile system for Web-based instruction. International Journal of Artificial Intelligence in Education 12, 4 (2001) 351-384

204. Weber, G., Kuhl, H.-C., Weibelzahl, S.: Developing adaptive internet based courses with the authoring system NetCoach. In: Bra, P.D., Brusilovsky, P., Kobsa, A. (eds.) Proc. of Third workshop on Adaptive Hypertext and Hypermedia. (2001) 35-48, also available at http://wwwis.win.tue.nl/ah2001/papers/GWeber-UM01.pdf
205. Witkin, H.A., Moore, C.A., Goodenough, D.R., Cox, P.W.: Field-dependent and field-independent cognitive styles and their educational implications. Review of Educational Research 47, 1 (1977) 1-64
206. Yin, X., Lee, W.S., Tan, Z.: Personalization of Web Content for Wireless Mobile Device. In: Proc. of Wireless Communications and Networking Conference. IEEE (2004) 2569-2574
207. Zadeh, L.: Fuzzy sets. Information and Control 8 (1965) 338-353
208. Zapata-Rivera, D., Greer, J.: SModel Server: Student Modelling in Distributed Multi-Agent Tutoring Systems. In: Moore, J.D. (ed.) Proc. of 9th World Conference of Artificial Intelligence and Education, AIED'99. IOS Press (2001) 446-455
209. Zapata-Rivera, D., Greer, J.: Inspectable Bayesian student modelling servers in multi-agent tutoring systems. International Journal of Human-Computer Studies 61 (2004) 535-563
210. Zimmermann, A., Specht, M., Lorenz, A.: Personalization and Context Management. User Modeling and User-Adapted Interaction 15, 3-4 (2005) 275-302