The Learner's Mirror

D. Vogiatzis* [1], A. Tzanavari[1], S. Retalis[2], P. Avgeriou[3]
and A. Papasalouros[4]

[1] Department of Computer Science,
University of Cyprus
75 Kallipoleos St. P.O. Box 20537,CY-1678,
Nicosia, CYPRUS
{dimitrv, aimilia}@ucy.ac.cy

[2] Department of Technology Education and
Digital Systems, University of Piraeus
80 Karaoli & Dimitriou,185 34 Piraeus, GREECE
retal@unipi.gr

[3] Software Engineering Competence Center
(SE2C), University of Luxembourg
6, rue Richard Coudenhove-Kalergi L-1359 ,
Luxembourg-Kirchberg, LUXEMBOURG
paris.avgeriou@uni.lu

[4] National Technical University of
Athens,Department of Electrical and Computer
Engineering
Software Engineering Laboratory, 15780 Athens,
GREECE
andpapas@softlab.ntua.gr

* Contact Author

The Learner's Mirror

Abstract

Adaptive Hypermedia Educational Systems, AHES, represent an emerging technology that provides a unique advantage over traditional Web-based Educational Systems; that is the ability to adapt to the user's needs, goals, preferences etc. This system is increasingly becoming part of the mainstream education, yet there does not exist a disciplined way of designing them - most of the development is *ad-hoc*.

This paper aims to fill this void, which is the absence of disciplined design, by recording the expertise of existing Adaptive Hypermedia Educational Systems in the form of design patterns. In this paper we present a number of these patterns, as well as a more complete organization map that illustrates the entire pattern language based on the patterns' relations.

Introduction

An Adaptive Hypermedia Educational System (AHES) is a dynamic web-based application, which provides a tailored learning environment to its users, by adapting its key features, which are:

- the **content** - the educational material that the learner can explore in the application
- the **navigation** and **interaction** capabilities by which the user can explore the content and interact with it;
- the **activities** in which the user can be engaged and by which (s)he can modify the content and navigation structures (e.g., by marking some interesting material, by collecting material in personal "lessons") or the user representation (e.g., by answering some questions or tests);
- the **lay-out** - the concrete presentation on the screen of all the previous features.

This paper aims to initiate a pattern language in the domain of AHES for the user modelling component, which plays a pivotal role in the adaptation that takes place as part of an AHES. It is responsible for forming and maintaining an accurate "image" of the user, which at the same time has to be meaningful and useful to the system. This is subsequently used in the adaptation phase, where primarily content and presentation are tailored to the user's needs [Kavčič 1999]. The patterns presented here attempt to cover the entire user modelling process, both at design and run time.

More specifically, the patterns attempt to elaborate on the decisions that should be made for:

1. The user model definition, which shows what information a user model should have. Information about the user can be regarded as domain– independent and domain–dependent; or divided into a knowledge component (user's knowledge) and a preferences component (user's preferences); or as static (constant through the learning process) and dynamic (changes during the learning process).

2. The user model initialisation addresses the problem of deriving the user model at the beginning of the learning experience. Some applications start from an empty user model, but most of the times users fill in a short questionnaire with questions that refer to (usually part of) the user model, so that it is initialised. Subsequently, the model can be used as it is and modified as the learning experience progresses, or well–known methods from artificial intelligence and machine learning can be used for determining his/her user model characteristics, in full. Such methods include:

- Bayesian networks,
- rule learning,
- instance–based learning,
- learning of probabilities,
- logic–based,
- decision – theoretic,
- heuristic,
- other general techniques and principles (plan recognition),
- specifically developed computational or specifically developed qualitative rules and procedures (rules for selecting and evaluating examples, rules for choosing adaptation type, rules for choosing questions) [Jameson 1999].

3. The user model maintenance addresses the problem of maintaining an accurate user model. Inputs for this part of the user model are gathered directly from users, through:

- tests and practice (test results, user history of responses and problem solving behaviour),
- user's actions (browsing behaviour, number of nodes visited, visited concepts, time spent on page, total session time, selection of links, searching for further information, queries to the help system).

This information is constantly being collected during the learning process and is also used for updating the user model.

4. The user model implementation that contains definite design decisions for the implementation of the user model in an AHES. These are depicted in the lower part of figure 2 (below the dotted line). In particular the user model implementation pattern is divided into three patterns, corresponding to the three patterns of the upper level of figure 2 (above the dotted line). That is, there is *Definition Representation*, *Initialisation Implementation* and *Maintenance Implementation* that refer to user model definition, user model initialisation and user model maintenance respectively. The user model implementation solves the problem of choosing among the data types that can be used for representing the various elements the user model, methods for initialising, as well as maintaining the user model. In practice, the elements of user model can be represented as trees, lists, hash tables, etc. Concerning the methods, a combination of two or more is frequently applied, especially when a different method is used for initialising and a different for maintaining the user model. This assures more accurate modelling and allows better exploitation of gathered information. These methods include:

- Bayesian methods, machine learning methods
(rule learning, learning of probabilities, instance–based learning),
- logic–based methods (first order predicate calculus),

- specifically developed computational procedures (user’s expertise is calculated from their navigational actions or time spent on documents).

The organization of the user modelling patterns

This section presents an initial set of design patterns for the user model component of the Adaptive Hypermedia Educational Systems, AHES. The organization of these patterns can be formed based on their relation to one another. Figure 1 depicts a higher level organization of the design patterns proposed in this paper, whereas Figure 2 depicts the complete proposed pattern language, along with relationships between the individual patterns. The patterns shown in Figure 2, form a tree-like structure, to illustrate that the "children" patterns directly relate to the "parent", by focusing and elaborating (each one) on a particular aspect of the "parent's solution, and that the greater the pattern’s depth in the tree, the more specific the pattern.

This paper will deal with the patterns referred to in the high-level organization, as shown in Figure 1, except from the user model implementation. The main reason is that no firm results from evaluation studies exist, as yet that could help us in deriving design patterns that hold the distilled knowledge of implementers.

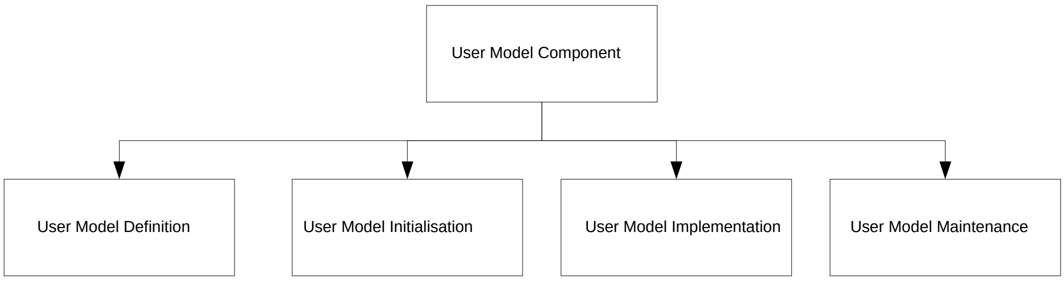

Fig. 1: The high-level patterns for user modelling in AHES

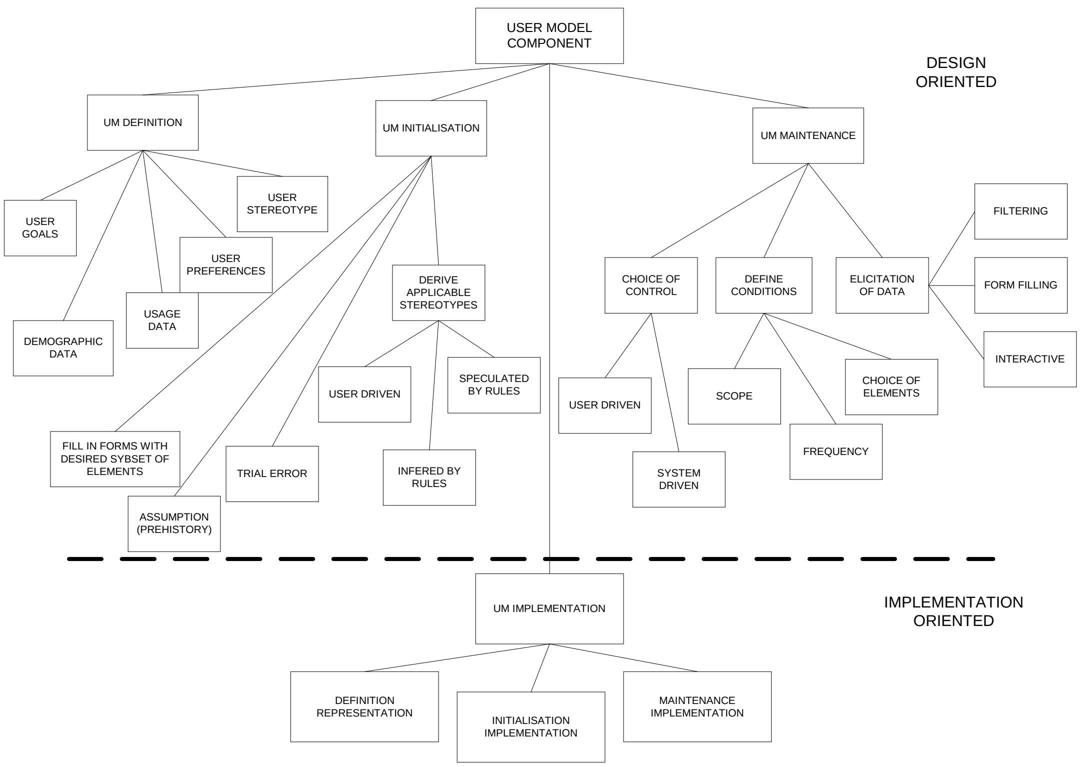

Fig. 2: The complete pattern language for user modelling in AHES

User Model Definition

Problem

In a traditional educational setting, an instructor is considered “a good one” when (s)he can get most out of her/his students individually, that is, taking into account different learning styles and needs. When the instructor’s role is to be played by an educational hypermedia system, then ignoring the learner’s individuality, limits the system’s ability to offer her/him an effective learning experience. Therefore, the system’s adaptation to individual learning-related characteristics is essential.

What information should an Adaptive Hypermedia Educational System keep for the user in order to offer him/her the best possible learning experience?

Analysis

A user model is essentially the “image” the system has about the user; the information it holds to describe him/her. An adaptive educational hypermedia system enriches its functionality by maintaining a user model and providing mechanisms to modify application features based on that. Modifications can be related to the organizational and presentational issues of the learning resources permission to continue or not, encouragement to read specific sections, undertake some tasks, move to a higher difficulty level etc., resulting in a personalized instruction. The closer the user model is to the user’s real characteristics and needs, the better the personalization. Therefore, the information kept in the user model has to be such that it describes the user/learner in the best way possible, but at the same time allows the model to be flexible in its manipulation.

Standardisation of the user model is an important issue, because through this we can greatly enhance the user model’s portability, as well as the interoperability of AHES that utilize such descriptions of learners. This will allow users to use several different AHES and “carry” their personal model with them, providing the systems with the same “image” of themselves, without that leading to compatibility problems. Attempts to standardize the user/learner model that should be taken into account are the IEEE Personal and Private Information, PAPI, [IEEE PAPI] and the IMS Learner Information Package, LIP [IMS LIP]. The PAPI standard reflects ideas from intelligent tutoring systems where the performance information is considered as the most important information about a learner and also considers interpersonal relationships [Vassileva et al. 2002], whereas the LIP standard is based on the classical notion of a CV and interpersonal relationships are not considered at all. Both get into too much detail and are thus hard to use. We need a user model that is smaller, more compact and more flexible.

Solution

A user in general is very complex to describe, meaning that theoretically, the information that would be needed to fully describe him/her (with great detail), would be too much for an application to handle, but also part of it would probably not be utilised. Consequently, a certain number of information items have to be carefully selected to form the user model. In an AHES setting, the items have to be directly related to the user as a learner – anything that would be considered useful to better adapt the learning experience to the learner's particular characteristics.

The IMS LIP and PAPI proposed standards for a learner model, include indeed several important attributes to describe the learner. However, one can observe that useful information that is missing from the first, can be found in the second or vice versa, or is missing from both. By closely looking at the two, we can identify the most useful elements and then enrich them with the necessary features to fill in the gaps.

As a result, a complete user model definition should generally be comprised of the following elements:

Demographic data, which are relevant to the particular AHES (e.g. as age, gender, etc.)

User goals, which are related to the long term and short term learning goals related to learning objectives of specific concepts to be learnt (e.g. "to complete course X")

User preferences with respect to the various dimensions of the learning opportunity (e.g. the mode of delivery, accessibility requirements, or assessment)

User knowledge, which includes the knowledge level about concepts to be learned and weaknesses and strengths on particular areas, sections or points of the concepts

Usage data, which include information like which pages were viewed, in what order, etc.

The **stereotype** that applies to the user, which essentially is the group of users s/he belongs to based on some predefined presuppositions in terms of knowledge level, learning and cognitive styles (e.g. the "Novice User", the "Expert User", the "acoustic user", the "activist user" stereotypes etc.).

Note that the above list is not restricting – it merely intends to provide the more generic elements with respect to the description of a learner. Designers are encouraged to include other specific elements that would fit their custom AHES.

Known uses

Interbook [Brusilovsky et al. 1998] and BGP-MS [Kobsa et al. 1994] mainly base the user model on the user knowledge, usage data, user goals and stereotypes. ALE [Specht et al. 2002] also maintains information about usage data, including evaluation results, as well as user knowledge and goals. ISIS-Tutor [Brusilovsky & Pesin 1994] incorporates user knowledge and usage data and in ELM-ART II [Weber et al. 1997] the topics learned by a user are represented as values (from a controlled vocabulary) that are assigned to the systems' units.

Information kept in user models used by the I-Help [Bull et al. 2001] system includes: knowledge, interests, cognitive style, interaction preferences and user actions. In addition, the notion of a group (the one the user belongs to), is employed extensively. The personal learner assistant developed within the ELENA project [Dolog & Nejdl 2003] is using the proposed blended approach which is represented with a RDF schema [Dolog et al. 2003].

User Model Initialisation

Problem

Before all interaction, the Adaptive Hypermedia Educational System initialises the user model.

What is the minimum amount of information needed to kick start the system? What kind of information and what amount is the user capable or willing to provide?

Analysis

Not all elements of the User Model definition have to be acquired in order for the user to start interacting with the AHES. There are two reasons for that. In the beginning of an interaction session, users do not like spending a lot of time providing information about them, answering long questionnaires for instance. Second, it is not necessary to have a complete model of the user; a partial model (with proper selection of a subset of UM elements) will be acceptable.

There are UM elements that can be acquired directly from the user and data that can be acquired through the AHES. For instance, demographic data can only be provided by the user. On the other hand, user knowledge can also be derived by the system e.g. via the prehistory of user's learning activities in other educational environments.

It is also important to initialise the user stereotype, because according to the various groups of users based on their stereotypes the learning tasks will be specified for each group separately.

There are two options for the UM, it will be identified with certainty, or it will be speculated. The first option is not usually the case in practical AHES systems.

Solution

The AHES designer should create fill-in forms with questions that refer to a desired subset of UM elements. The desired subset is required to form an initial view of the user model so as to kick start the AHES. There are a number of ways whereby the desired elements can be derived. Below we provide a list of plausible choices stemming from real AHES systems that a designer should take into account.

The desired UM elements could be obtained explicitly, by presenting to the user a questionnaire, which s/he has to fill in. Typically, the user provides data, such as demographic data, user preferences, and possibly other sorts of data that are compatible with the user model description specification.

Another option is to assume values for the desired UM elements from previous training sessions/learning activities of the user. For instance, a user having followed the prerequisites of the current course is considered to have enough knowledge to follow it.

Yet another option is to assume certain values with nothing to backup this choice apart from being plausible for the desired UM elements, and then to proceed with the interaction, expecting that the user model will be corrected during the running time of the AHES. This is essentially a trial and error approach.

Deriving the applicable stereotype requires that a minimum amount of knowledge and specifically a minimum number of user model definition elements is available. The derivation of the applicable stereotype can be performed in a number of ways.

The following list is to be considered as indicative rather than complete:

- **It can be user driven**

For instance the user specifies explicitly that (s)he belongs to the novices' stereotype

- **Inferred by rules**

Stereotypes are equipped with triggers, which activate them.
Rules tell which UM elements and with what values can activate a stereotype.

- **Speculated by rules**

If it is the case that there is absolutely no information which can suggest a certain stereotype, then the AHES designer should have some rules to allow selection of the stereotype. For instance, a rule of this kind might be: if user does not specify his/her knowledge level, then assume it is average.

Known uses

In INSPIRE [Grigoriadou et al. 2001] the user model is initialised, through a questionnaire filled in by the user at the beginning, or by explicitly selecting the category s/he fits in according to some general characteristics. ELM-ART II [Weber et al. 1997], requests from the users to declare knowledge units, which are already known to them. In DCG [Vassileva 1997], the user model, called student model, is initialized with a preliminary test. ACE [Specht et al. 2000] follows a somewhat mixed approach. The user model is initialised by explicit and implicit elicitation from the users. The former is performed, by the user, which specifies her/his learning strategy and stereotype; whereas the latter is done by a dynamically generated test.

User Model Maintenance

Problem

During the course of interaction, many things about the user are changed, e.g. assumed user knowledge, usage data etc. Thus, the user model must be adapted to the new realities. After all, the first letter in AHES stands for Adaptive.

How should the system capture those changes so as to maintain a good user model?

Analysis

The assumption that the user model will remain the same as when it was acquired originally is in most cases incorrect. As in tutoring between a human tutor and a student, where the student constantly demonstrates changes, the user of an AHES also changes and as a result his/her model has to reflect this. During the course of interaction, leverage of user knowledge develops and the usage data builds up. Since the adaptation is to a large extent based on user knowledge and usage data, changes should definitely be recorded and be related to a "cause / result".

In fact, information such as "demographic data" does not change with a high frequency. There is also, information like "topics covered", part of user knowledge effect if the system is to function effectively that changes continuously.

It is also important for users to be in control, to a degree acceptable to the AHES, of their model for several reasons. They need to be able to modify information in their model if they feel that it is inaccurate or incorrect. Also, being in control, builds up their trust in the system.

Solution

The maintenance of an accurate User Model, UM, can be user driven or system driven. In the former case it is the user who provides explicit information about changes in his/her UM. In the latter case, the AHES derives information by closely watching the user.

The AHES designer should define the conditions that govern the maintenance of the user model. In particular the designer should define the scope of the maintenance changes. The scope defines the reason for updates. The reason is then quantified in terms of choice of elements to undergo change. For instance, if the scope says that only a minimal update of the user knowledge is going to occur, then choice of elements is bound to user knowledge only. On the other hand a wider choice for the scope would allow updates of user knowledge and user preferences e.g. to read theory and having links to examples. Finally, the frequency of UM description elements updates should be defined.

The UM maintenance module elicits data to update the UM description. Elicitation of data could take many forms; next we provide some characteristic examples. The user is presented with a form to fill in, whereby the update UM is derived. UM description update can also be interactive, when the AHES opens for instance a pop-up form requesting the user to explicitly answer a question. Finally, another option for UM update is through filtering the stream of data that are produced through user interaction. A typical example is the browsing strategy which can be reduced to a small number of primitives, like ‘ringiness’ (a route that returns to the start node), ‘spikiness’ (a route with a return path retracing the original path), ‘loopiness’ (a ring that contains no other rings), ‘pathiness’ (a route that does not visit any node twice) [Canter et al., 1985]. The raw data constitute the actual path that the user has followed, but they must be filtered, processed or summarized to be translated to the predefined number of primitives.

Known uses

In [Grigoriadou et al. 2001] there is an Interaction Monitoring Module, which collects information and updates the learner model accordingly. The system allows the users to intervene, expressing their perspective. A similar approach is followed in [Weber et al. 1997], where the update of the user model is driven by the system. It is also possible to inspect and to edit the user model. Yet another similar approach is followed in [Vassileva 1997]. Student model changes are performed according to student progress. Students can also explicitly modify their Personal Traits and Preferences. In [Specht et al. 2000] there is a diagnostic module for automated updates of the user module. Learners can also modify their model anytime.

Conclusions

In this paper we proposed a disciplined approach for designing AHES which takes advantage of the notion of design patterns that capture the expertise, know-how and tacit knowledge of the developers of such systems. This approach can be beneficial to inexperienced designers of AHES. We have introduced a conceptual schema and some patterns for the user modelling component of the AHES. The proposed set of patterns structured as a hierarchy will help an AHES designer to have a broad view of this component, the problems that he/she has to solve as well as their interdependencies. However, we have not tackled the difficult problem of user model implementation, which means that we have not proposed solid solutions for the developers. As mentioned [Borchers 2001], for the design of interactive systems, and, in our case educational systems, instructional designers, human computer interaction specialist, software engineers, and subject domain experts, all in a project team, should express their expertise in the form of a pattern language.

The future direction of our work is to propose software design patterns for AHES which will be considered a useful language for communication among software developers and a practical vehicle for introducing less experienced developers into the field. However, we will continue towards creating patterns for the design of AHES, since we agree with Borchers that there is "a good chance to push the concept of participatory design forward by introducing patterns". This is what we are trying with a European funded project called ELEN (http://www.tisip.no/ELEN). We are involving user representatives, instructional designers, e-learning specialists and domain experts in the design process to evaluate prototypes of e-learning systems and participate in design discussions.

Acknowledgement

This work was supported by the "ELEN: A Network of e-learning centers" project which is partly sponsored by the European Commission under Socrates Minerva programme (ref nums: 101421-CP-1-2002-1-CY-MINERVA-M).

References

[Alexander et al. 1977] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel, S. *A Pattern Language* (Oxford University Press, 1977).

[Borchers 2001] Jan Borchers. A Pattern Approach to Interaction Design, John Wiley & Sons, ISBN: 0471498289, 2001

[Brusilovsky & Pesin 1994] Brusilovsky, P. and Pesin, L. ISIS-Tutor: An adaptive hypertext learning environment. JCKBSE'94, Japanese CIS Symposium on knowledge-based software engineering, Pereslavl-Zalesski, Russia, pp. 83-87, 1994.

[Brusilovsky et al. 1998] Brusilovsky, P., Eklund, J. and Schwarz, E. (1998) Web-based education for all: A tool for developing adaptive courseware. *Computer Networks and ISDN Systems* Vol. 30, Issue 1-7, pp. 291-300.

[Brusilovsky 2001] Peter Brusilovsky. Adaptive hypermedia. User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 87-110

[Bull et al. 2001] Bull, S., Greer, J., McCalla, G., Kettel, L. and Bowes, J. (2001) User Modelling in I-Help: What, Why, When and How. In: *Proc. of User Modeling 2001. LNAI 2109*, pp. 117-126.

[Canter et al., 1985] Canter, David, Rivers, Rod, and Storrs, Graham. Characterizing user navigation through complex structures. In Behavior and Information Technology, 4(2), 93-102.

[Dolog et al., 2003] Peter Dolog, Rita Gavriloaie, Wolfgang Nejdl, and Jan Brase. Integrating adaptive hypermedia techniques and open rdf-based environments. In Proc. of 12th International World Wide Web Conference, Budapest, Hungary, May 2003.

[Dolog & Nejdl 2003] Peter Dolog and Wolfgang Nejdl. Personalisation in Elena: How to cope with personalisation in distributed eLearning Networks. In Proceedings of the Conference on Worldwide Coherent Workforce, Satisfied Users - New Services For Scientific Information, Oldenburg, Germany, September 2003.

[Grigoriadou et al. 2001] Grigoriadou, M., Papanikolaou, K., Kornilakis, H. and Magoulas, G. (2001). INSPIRE: An Intelligent System for Personalized Instruction in a Remote Environment. In: Proc. of the 3rd Workshop on Adaptive Hypertext and Hypermedia, in User Modeling 2001, LNCS, Springer

[IEEE PAPI 2002] IEEE. IEEE P1484.2/D7, 2000-11-28. draft standard for learning technology. public and private information (papi) for learners (papi learner). Available at: http://edutool.com/papi/. Accessed on May 6, 2004.

[IMS LIP 2002] IMS. IMS learner information package specification. Available at: http://www.imsproject.org/profiles/index.cfm. Accessed on: May 5, 2004.

[Jameson 1999]. A. Jameson, "User–Adaptive Systems: An Integrative Overview", Tutorial presented at UM99, 7th International Conference on User Modeling, Banff, Canada, 1999.

[Kavčič 1999]. Alenka Kavčič, "Adaptive Hypermedia Learning System on the Web", EUROMEDIA'99, München, Germany, April 1999.

[Kobsa 2001] Alfred Kobsa. Generic user modeling systems. User Modeling and User-Adapted Interaction, 11(49):49–63, 2001.

[Kobsa et al. 1994] Kobsa, A., Miiller, D. and Nill, A. (1994) KN-AHS: An adaptive hypertext client of the user modeling system BGP-MS. In: *Proc. of User Modeling 1994*, pp. 31-36.

[Specht et al. 2000] Specht, M., Oppermann, R. (2000) ACE: Adaptive Courseware Environment, Lecture Notes in Computer Science, Vol. 1892

[Specht et al 2002] Specht, M., Kravcik, M., Klemke, R., Pesin, L. and Hüttenhain, R. (2002) Adaptive Learning Environment for Teaching and Learning in WINDS. In: *Proc. 2002 Conference on Adaptive Hypermedia and Adaptive Web Based Systems*.

[Vassileva. 1997] Vassileva J. Dynamic Courseware Generation, Communication and Information Technologies, Vol. 5, Issue 2, pp. 87-102

[Vassileva et al, 2002] Julita Vassileva, Gordon McCalla, and Jim Greer. Multi-agent multi-user modelling in I-Help. User Modeling and User-Adapted Interaction, 2002.

[Weber et al. 1997] Weber, G. and Specht M. (1997) User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems. In: Proc. of User Modeling 1997.