

TOWARDS A PATTERN LANGUAGE FOR ADAPTIVE WEB-BASED EDUCATIONAL SYSTEMS

P. Avgeriou¹, D. Vogiatzis², A. Tzanavari², S. Retalis³

¹Software Engineering Competence Center, University of Luxembourg, 6, rue Richard Coudenhove-Kalergi L-1359 Luxembourg-Kirchberg, Luxembourg
paris.avgeriou@uni.lu

²University of Cyprus
Department of Computer Science
75 Kallipoleos Str., P.O. Box 20537,
1678 Nicosia, Cyprus
{dimitrv, aimilia}@cs.ucy.ac.cy

³University of Piraeus,
Department of Technology
Education and Digital Systems
80 Karaoli & Dimitriou Str.
185 34 Piraeus, Greece
retal@unipi.gr

Abstract

Adaptive Web-based Educational Systems 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. Adaptive Web-based Educational Systems are 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 Web-based Educational Systems in the form of design patterns. We present a categorization of the patterns according to an established paradigm in Adaptive Hypermedia and we provide three exemplary patterns.

Keywords

Design Patterns, Pattern Language, Adaptive Hypermedia Systems, User Modelling, e-Learning.

1. Introduction

An Adaptive Web-based Educational System (AWES) [1] is a dynamic web-based application, which provides a tailored learning environment to its users, by adapting both the presentation and the navigation through the content. Such a system is comprised of learning resources, as well as a set of tools that facilitate the process of studying, such as exams/questionnaires, glossaries, communication tools, etc. The learning content is dynamically generated based on some pedagogical rules that combine the (domain) model of the content with the model of the user. AWES are currently a 'hot' topic of research in the broader field of adaptive hypermedia applications [2],[3] and several AWES systems have been built during the past years [4], [5], [6], [7], [8], [9], [10], [11], [12], [13].

Several educational institutions are nowadays designing and developing their own AWES due to the fact that

AWES leverage the shortcomings of non-adaptive web-based educational systems, also known as Learning Management Systems or Virtual Learning Environments. In contrast to the latter, AWES can provide tailored, one-to-one tutoring, according to the specific characteristics of each individual learner rather than serve the same content massively to all the learners. Furthermore, some educational or research institutions tend to develop their own AWES in order to implement and test their own learning theories or instructional design methods.

However, the design and implementation of Adaptive Web-based Educational Systems (AWES) is a complex, if not overwhelming, task. This is due to the fact that it involves people from diverse backgrounds, such as software developers, web application experts, content developers, domain experts, instructional designers, user modelling experts and pedagogues, to name just a few. Moreover, these systems have presentational, behavioural, pedagogical and architectural aspects that need to be taken into account. To make matters worse, most AWES are designed and developed from scratch, without taking advantage of the experience from previously developed Adaptive Web-based Educational Systems, because the latter's design is not codified or documented. As a result, development teams are forced to 're-invent the wheel'.

Therefore, systematic and disciplined approaches must be devised in order to overcome the complexity and assortment of AWES and achieve overall product quality within specific time and budget limits. One such approach is the use of **design patterns** [14], so that these systems are not designed and implemented from scratch, but based on reusable design experience gained over several years of trial-and-error attempts. Therefore good design can be made explicit, and available to the whole community of designers, so that it becomes common practice. In this way, designers of new or existing AWES, especially inexperienced designers, can take advantage of previous design expertise and save precious time and resources.

The idea of design patterns began in the field of building architecture, when Christopher Alexander conceived the idea of capturing design guidelines in the form of design patterns [14]. According to him, "each pattern describes a problem, which occurs over and over again in our

environment, and then describes the core of the solution to that problem in such a way that you can use this solution a million times over". Patterns are not conceived in a 'big bang' but rather discovered or mined after numerous implementations of the same solution in a given problem, usually by different people. Alexander also proposed the notion of a **pattern language**, which is a collection of related patterns that captures the whole of the design process and can guide the designer through step-by-step design guidelines.

The 'Alexandrian' patterns found many followers in the computer science discipline, especially after the so-called 'GOF' book for object-oriented design [15]. Some of the fields that have adopted patterns are: software architecture [16],[17], hypermedia engineering [18], object-oriented analysis [19], business modelling [20], etc. Design patterns have also been developed for the field of Human Computer Interaction [21], including patterns for Interaction Design [22], [23], User Interfaces [24], Web Applications [25], Cooperative Interaction [26], Usability [27] and for teaching Human-Centred Design [28]. There is also a repository of patterns in the conventional learning and pedagogical discipline, mainly focused on teacher-based learning [29]. Recently the e-learning community has established a design pattern repository [30],[31], and it is this domain where the added value of this paper aims at. A relevant initiative that attempts to define reusable design techniques in user-modelling based adaptivity and adaptability can be found at [32].

This paper aims to initiate a pattern language for the domain of AWES, based on a well-established reference model for adaptive hypermedia, the AHAM model [2]. In particular, it uses the various layers, components and interfaces of AHAM to categorize and organize the patterns. It also portrays an overview of the entire pattern language at three levels of abstraction:

- The seven categories of patterns according to AHAM
- The organization of the patterns in a specific category, the user model
- A sample exemplary pattern for each level of refinement of the user model category

The structure of the rest of this paper is as follows: the second section categorizes the pattern in the domain of AWES into seven thematic groups by utilizing the AHAM reference model. The third section presents an overview of the patterns in one category, namely the user model patterns, while the fourth section portrays three exemplary patterns for different levels of refinement in the user model category. Finally, the fifth section wraps up with conclusions and ideas for future work.

2. Design Patterns for AWES

The most widely accepted reference model for adaptive hypermedia applications is AHAM [2], a model based on the well-established Dexter reference model [33] for

hypermedia applications. AHAM extends the Dexter model and proposes that an adaptive hypermedia application is comprised of the following parts:

- **Within component layer**, which deals with the content and structure inside the hypermedia nodes.
- **Storage layer**, which is comprised of three parts:
 - The **user model** which defines the knowledge acquired by the learner and the links s/he has visited.
 - The **domain model**, which is a "database" with all the nodes and links of the application.
 - The **teaching model**, which is a set of pedagogical rules that combines the user model and the domain model to perform the actual adaptation.
- **Run-time layer**, which deals with the presentation of the hypermedia to the user and takes care of the dynamic aspects of the system.

AHAM also defines two interfaces between the above layers:

- An **anchoring interface** that connects the storage layer and in particular the domain model, with the within component layer
- A **presentation specification interface** between the storage layer and in particular the domain model with the run-time layer.

Fig. 1 depicts the AHAM reference model, using the object-oriented notation of the Unified Modelling Language [34].

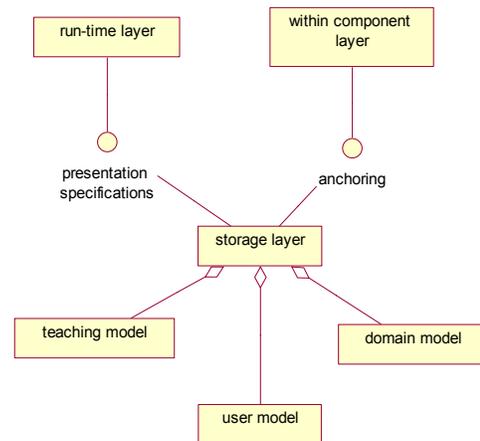


Figure 1. The AHAM Reference Model

According to the software engineering discipline [35] a reference model is a division of a system's functionality into parts, together with the flow of information between the pieces. The AHAM reference model accomplishes exactly that:

- It separates the functionality of an adaptive hypermedia application into three parts, namely the three layers, storage, within component and run-time.

- It defines the flow of information between the above parts through the two interfaces proposed by the model, anchoring and presentation specifications.

The AHAM reference model is adopted in this paper because it provides a clear and precise way to categorize the patterns in the domain of adaptive web-based educational applications. These categories define different thematic groups that solve similar problems, and they assist in managing the patterns, especially when their number increases to a large extent. Therefore, for the domain of AWES, we define seven categories of patterns:

- **Run-time layer patterns** that deal with the User Interface and dynamic aspects of AWES.
- **User model patterns** that concern the ways of creating and manipulating user models.
- **Domain model patterns** that concern the ways that the content is structured into nodes connected by links.
- **Teaching model patterns** that deal with the pedagogical rules that combine the user model and the domain model to perform the adaptation of the content.
- **Within component layer patterns** that concern the structuring of the content and the content per se within the nodes.
- **Anchoring patterns** that deal with the ways to link domain model elements with the within-component layer elements.
- **Presentation specification patterns** that concern the presentation of the domain model elements in the run-time layer.

The whole pattern language that is comprised of the aforementioned categories is not possible to be presented here due to its size. Instead the next section will provide an overview of the user model patterns, by briefly

outlining what the patterns are and what problems they solve.

3. The map of user model patterns

This section presents an overview of the design patterns that belong to the user model patterns category, as described in the previous section. The organization of these patterns can be achieved according to their interdependencies [15]. In the most fundamental repositories of patterns such as [16], [15], graphs or maps are designed that show how the distinct patterns refer to each other and what the nature of their relationship is. Fig. 2 depicts the relationships between the proposed AWES user model design patterns. The semantics of the arrows between the patterns is that the pattern at the beginning of the arrows defines a general solution that in sequence entails more specific problems that are resolved by the patterns at the end of the arrows.

The user model component of an AWES is responsible for describing, representing, acquiring and maintaining the user model [36]. It is of paramount importance to define a complete and accurate user model so that the system can better adapt to the user's individual needs and characteristics. This is subsequently used in the adaptation phase, where primarily navigation and presentation are tailored to the user's needs. The patterns presented here attempt to cover the entire user modelling process. They originate from studying [36] a significant number of existing AWES, along with codifying expertise in user modelling. It is noted that the pattern names are in uppercase letters, so as to distinguish them inside the text. The USER MODEL COMPONENT is the umbrella pattern addressing the problem of how we design the user model component of an Adaptive Web-based Educational System (AWES) from scratch. It entails the following patterns:

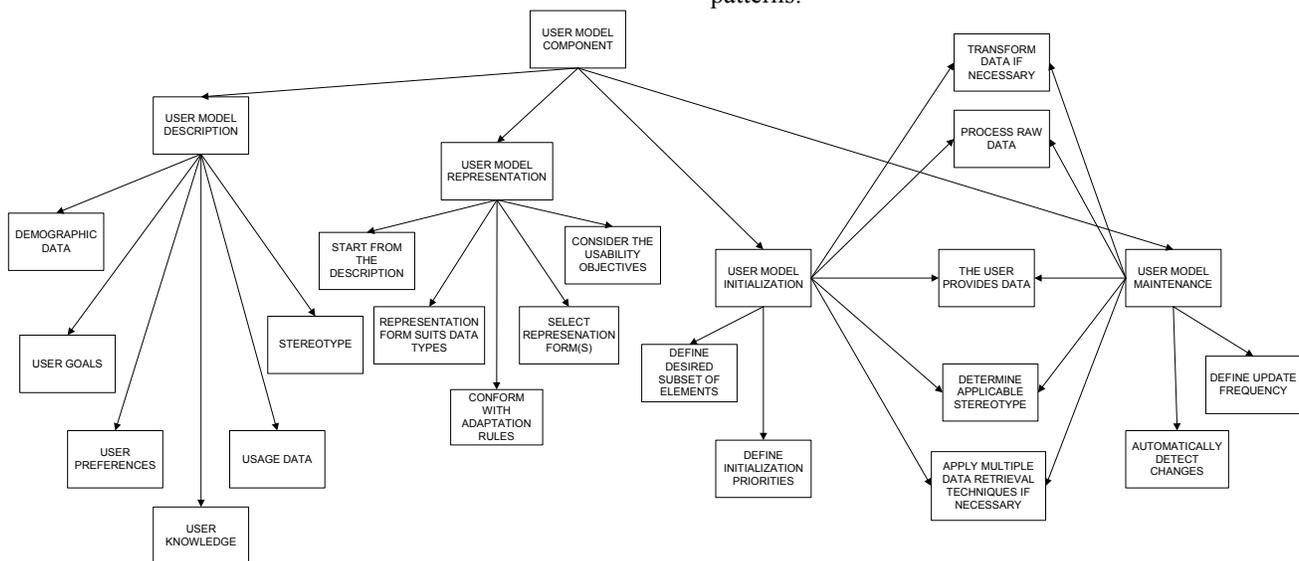


Figure 2. The patterns for user modelling in AWES

1. USER MODEL DESCRIPTION, shows what information should a user model that is to be used in an Adaptive Web-based Educational System (AWES) include, namely:
 - i. DEMOGRAPHIC DATA, which refers to the demographic information that has to be kept about the user.
 - ii. USER GOALS, which refers to the user's educational goals when using the AWES.
 - iii. USER PREFERENCES, which describes the user's preferences with respect to the various dimensions of the learning opportunity e.g. learning style or mode of assessment.
 - iv. USER KNOWLEDGE, which describes what should be considered as user knowledge and where this information may be found.
 - v. USAGE DATA, which portrays how usage data, can be exploited to learn things about the user.
 - vi. STEREOTYPE, which explains how stereotypes can be employed in a user model.
2. USER MODEL REPRESENTATION solves the problem of representing the user model. It is supported by the following patterns:
 - i. START FROM THE DESCRIPTION, which summarizes that the user model's description individual elements are the ones to be represented.
 - ii. REPRESENTATION FORM SUITS DATA TYPES, which explains why the data's characteristics can play an important role in the representation.
 - iii. CONFORM WITH ADAPTATION RULES, which gives evidence why the form of representation also depends on the adaptation rules format.
 - iv. CONSIDER THE USABILITY OBJECTIVES, which describes the reason why some of the AWES usability objectives may affect the decision regarding the representation form.
 - v. SELECT REPRESENTATION FORM(S), which demonstrates that one form of representation, may not be suitable to represent all the elements of the user model's description, but several forms may be selected.
3. USER MODEL INITIALIZATION addresses the problem of acquiring the user model initially. It is supported by the following patterns:
 - i. DEFINE DESIRED SUBSET OF ELEMENTS, which describes the subset of the user model elements that the designer considers sufficient to form the initial user model.
 - ii. DEFINE INITIALIZATION PRIORITIES, which analyzes the fact that some elements of the user model description, need to be initialized before some others.
 - iii. THE USER PROVIDES DATA, which describes the process of the user providing some elements of information directly.
 - iv. PROCESS RAW DATA, which describes the process of the system deriving some elements of

information based on the user's interaction with the AWES.

- v. TRANSFORM DATA IF NECESSARY, which explains the probability that the data derived after processing the raw data are not in the required representation form and as a result need to be transformed.
 - vi. APPLY MULTIPLE DATA RETRIEVAL TECHNIQUES IF NECESSARY, which gives evidence that different elements of the user model description, may not be suitable to be acquired with the same data retrieval techniques.
 - vii. DETERMINE APPLICABLE STEREOTYPE, which describes the process of determining the stereotype that applies to the particular user model description instance.
4. USER MODEL MAINTENANCE addresses the problem of maintaining an accurate user model.
 - i. AUTOMATICALLY DETECT CHANGES, which describes the process of the system monitoring the users' interaction to determine changes in their user model.
 - ii. DEFINE UPDATE FREQUENCY, which explains that the task of updating the user model should take place with a certain frequency that depends on a number of factors.
 - iii. THE USER PROVIDES DATA (same as above)
 - iv. PROCESS RAW DATA (same as above)
 - v. TRANSFORM DATA IF NECESSARY (same as above)
 - vi. APPLY MULTIPLE DATA RETRIEVAL TECHNIQUES IF NECESSARY (same as above).
 - vii. DETERMINE APPLICABLE STEREOTYPE (same as above)

4. A sample pattern from each level

A proposed pattern gets accepted by the corresponding pattern community only if there have been two or three examples of its use by someone other than the one who suggested the pattern [16]. Therefore we have searched for the implementation of the proposed patterns in existing AWES, such as AHA! [4], I-Help [10], ISIS-Tutor [7], ELM-ART II [8], BGP-MS (with KN-AHS) [11], Interbook [9], AST [13], INSPIRE [37], DCG [38], ACE [39] and ALE [12].

The next step was to describe them in a suitable format in a similar way to patterns of other domains. As eloquently stated in [40], it is more difficult to describe patterns than to actually find them. Almost all of the approaches that have proposed patterns in a subject field have also suggested a novel way of describing and cataloguing them. Our suggestion for a pattern description format is largely based on the format proposed in [41] and contains the following fields:

1. **Name** – a unique name to distinguish the pattern and uniquely refer to it.
2. **Problem** – a brief description of the design problem at hand.

3. **Context** – the situation in which the problem is solved by the solution.
4. **Forces** – the often-contradictory factors that need to be accounted for when choosing a solution to the problem.
5. **Solution** – a description of the solution proposed by this pattern that addresses the problem and motivation stated earlier.
6. **Related Patterns** – other patterns that are related to this one in some way, e.g. they are involved in the solution.
7. **Known uses** – examples of real AWES where this pattern has been applied.

A significant number of patterns of this language have been presented in [36]. For illustrative purposes, in this paper we include one pattern from each level as seen in Fig. 2, namely the USER MODEL COMPONENT, the USER MODEL DESCRIPTION and the USER GOALS patterns.

4.1 The USER MODEL COMPONENT

Problem: How do we design the user model component of an Adaptive Web-based Educational System (AWES)?

Context: You are designing an Adaptive Web-based Educational System. You are at the stage of designing the USER MODEL COMPONENT.

Forces:

- Maintaining knowledge about the user's individual needs and characteristics is the most important factor in being able to provide tailored tutoring through AWES.
- Different users have different goals, plans, attributes, capabilities, knowledge and beliefs. The collection of these elements represents the user, in the sense that it describes their uniqueness in relation to the use of an AWES. This collection is referred to as the user model.
- The user's characteristics can be determined through their interaction with the system. However, once this information becomes available, it has to be stored in a format that will facilitate further processing.
- During the user's interaction with the system, the elements that form their model will most probably change. This change is to be detected and used to update their model so that the system has an accurate (to the degree possible) image of the user at all times. Maintaining good user models is fundamental for designing successful AWES that tailor instructional strategies, in terms of both content and style, and also provide feedback, hints, examples or extra problems accordingly.

Solution:

The task of designing a user modelling component can be divided into several smaller problems/questions: what

information the user model should include, how it should be extracted, how it should be represented, how it should be updated. Formally, this component should be comprised of the following elements:

- USER MODEL DESCRIPTION - what we know about the user that will be useful in further processing
- USER MODEL INITIALIZATION - what methods do we use to gather the information needed to form the user model initially
- USER MODEL REPRESENTATION - how do we represent the user model
- USER MODEL MAINTENANCE - how is an accurate user model maintained

Related patterns: USER MODEL DESCRIPTION, USER MODEL INITIALIZATION, USER MODEL REPRESENTATION, USER MODEL MAINTENANCE

Known uses: All the AWES examined have implemented this pattern, since it is the foundation stone for an AWES.

4.2 The USER MODEL DESCRIPTION

Problem: What information should a user model that is to be used in an Adaptive Web-based Educational System (AWES) include?

Context: You are designing an Adaptive Web-based Educational System and specifically the USER MODEL COMPONENT. You are at the stage of designing the USER MODEL DESCRIPTION.

Forces:

- A user model in an AWES is essentially the information the system holds about the user and is mainly related to the learning process.
- This information has to be such that the system can better adapt to the user's individual needs.
- When observing the interaction between a human personal tutor and a student, we can identify several adaptations that take place, some of which do so explicitly and some implicitly. The tutor will primarily organize the material based on the student's level, his preferred mode of delivery, the required course duration, etc. Subsequently and throughout the duration of the course he will monitor the student's progress and provide feedback, hints, examples or extra problems accordingly. The information based on which the tutor can take decisions like the ones mentioned, is information that also has to be available to an AWES for the same purpose: better adaptation to the user.
- It is often necessary to design the user model description in such a way that the description is portable to other AWES as well. For example if a user moves to a different institution s/he would need to be accompanied by the user model formed in the

previous AWES so that it can be imported into the new AWES.

Solution:

The information that has to be kept in the user model for the system to better adapt itself can be divided in a number of distinct elements. Specifically, a complete user model description should generally be comprised of the following elements:

- DEMOGRAPHIC DATA, which are relevant to the particular AWES (e.g. as age, gender, etc.)
- USER GOALS, which are related to the specific topic 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 or assessment)
- USER KNOWLEDGE, which includes topics covered and weaknesses and strengths on particular areas, sections or points of the topic to be learnt
- 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 (e.g. the “Novice Users” stereotypes, the “Expert Users” stereotype).

In order to achieve portability of the user model, its description should be compatible with an international accredited standard. Currently, the most suitable standard to describe the user model is the IMS Learner Information Package Specification [42]. This standard entails all six elements prescribed by this pattern.

Related patterns: USER MODEL COMPONENT, DEMOGRAPHIC DATA, USER GOALS, USER PREFERENCES, USER KNOWLEDGE, USAGE DATA, STEREOTYPE.

Known Uses: In INSPIRE [37], the description is comprised of data on the learner herself (general information, learning style), knowledge level on different topics and learning goals, performance on tests, number, type and order of resources s/he has accessed, etc. In ELM-ART [8] the description contains pages visited, test performance, known items, inferred as known items, user interaction (episodic knowledge). In DCG [38] it is limited to student knowledge, history and personal traits and preferences. In ACE [39] it involves interaction knowledge (used components), interests (unit clusters, hypothesis), knowledge (learned units) and preferences.

4.3 USER GOALS

Problem: What information should be considered as user goals in a user model that is to be used in an Adaptive Web-based Educational System (AWES)?

Context: You have designed the USER MODEL DESCRIPTION as part of the USER MODEL COMPONENT of an AWES and have decided to include

USER GOALS. You are currently reflecting on what to consider as USER GOALS.

Forces:

- Being able to model the user/learner’s educational goal(s) can facilitate adaptation. An AWES (via its author) can “deliver” the same course differently to learners with different educational goals, by setting the appropriate conditions to meet those goals. For example a learner with a goal “to master subject X” will receive more in-depth tutoring than a learner with a goal “to familiarize themselves with subject X”.
- It is incorrect to assume that all users/learners aim at learning all of the material offered by an AWES.
- Educational goals can vary in scope. For example they may refer to the whole duration of the course, or to only a part of it.
- There are some types of user goals that can be determined by the users, but some others cannot. For example, the users can determine the initial educational goal before starting a course using the AWES, but probably cannot be in a position to set the best (short-term) goal(s) for themselves while the course is in progress.
- The user goals determined may be in a form that is not suitable to be directly included in a user model.

Solution:

Include specific user goals in the user model in order to facilitate adaptation, and capture the real intent of the learner with respect to the learning material. The information that has to be kept as user goals in order for the system to better adapt to its user, is divided in two categories:

- Long-term goals - educational goals that are valid for a longer period of time and require significant effort to be met.
- Short-term goals – educational goals that are valid for a shorter period of time and require relatively moderate effort to be met.

Long-term goals are usually determined by the users, whereas short-term goals by the AWES that plays the role of a tutor and is driven by the course author. In the second case, a goal-modelling component may be required: a component that will take relevant data (frequently USER KNOWLEDGE), process it and derive goals.

In order for particular user goals to be included in a user’s model, a pre-processing operation may be necessary to bring them to the required format that was defined by the USER MODEL REPRESENTATION.

Related patterns: USER MODEL DESCRIPTION, USER MODEL REPRESENTATION, USER KNOWLEDGE.

Known uses: Interbook [9] initially modelled an educational goal as a sequence of sets of concepts, while later on as a stack of sets, allowing the user to move a

selected goal to the top of the stack. BGP-MS [11] models the user's goals in multiple ways. Firstly, the developer of BGP-MS applications can specify groups of users that share common goals. In addition, s/he is able to specify the user goals that correspond to specific answers to a questionnaire, as well as to specify the user goals that correspond to specific user actions as they are observed by the system.

5. Conclusions and future work

We have introduced a pattern language for the design of Adaptive Web-based Educational Applications (AWES). The patterns have been categorized according to the AHAM paradigm, and one of these categories, that concerns the USER MODEL COMPONENT has been presented. Finally we demonstrated three of the patterns of the aforementioned category. It is noted that the hierarchical, tree-like categorization of patterns presented in the third section can be continued by decomposing the tree further and defining even lower-level patterns. For example, the USER GOALS pattern can be 'broken down' into smaller patterns that define types of user goals.

Our purpose was the introduction of a disciplined approach for designing AWES, which will be beneficial to inexperienced designers of AWES. The proposed pattern language structured as a hierarchy will help an AWES designer to have a broad view of the system, to know the components that should be incorporated, as well as their interdependencies. However, it should be understood that the pattern language is not a blue print design thus customization should be undertaken.

In the future, we aim to improve the pattern language by exposing it to the pattern community, as well as to incorporate it in a newly established relevant repository [31]. The final goal is to develop the rest of the patterns that are stated in the pattern language and have not been written yet.

References

[1] Brusilovsky, P. Adaptive and Intelligent Technologies for Web-based Education. *KI – Kunstliche Intelligenz*, 13(4), 1999, 19-25.
[2] De Bra, P., Houben, G. and Wu, H. AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. *Proc. ACM Conference on Hypertext and Hypermedia 99*.
[3] Brusilovsky, P. Adaptive educational systems on the world-wide-web: A review of available technologies. *Proc. of Workshop WWW-Based Tutoring - ITS 1998*.
[4] De Bra, P., Aerts, A., Smits, D. and Stash, N. AHA! Version 2.0: More Adaptive Flexibility for Authors. *Proc. AACE ELearn 2002*.
[5] De Bra, P. and Calvi, L. Creating adaptive hyperdocuments for and on the Web. *Proc. WebNet' 1997*, 149-155.

[6] De Bra, P. and Calvi, L. Towards a Generic Adaptive Hypermedia System. *Proc. Workshop on Adaptive Hypertext and Hypermedia 1998*, 5-11.
[7] Brusilovsky, P. and Pesin, L. ISIS-Tutor: An adaptive hypertext learning environment. *Proc. JCKBSE 1994*, 83-87.
[8] Brusilovsky, P., Schwarz, E. and Weber, G. ELM-ART: An intelligent tutoring system on World Wide Web. *Proc. ITS 1996, LNCS 1086*, 261-269.
[9] Brusilovsky, P., Eklund, J. and Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. *Computer Networks and ISDN Systems*, 30(1-7), 1998, 291-300.
[10] Bull, S., Greer, J., McCalla, G., Kettel, L. and Bowes, J. User Modelling in I-Help: What, Why, When and How. *Proc. UM 2001. LNAI 2109*, 117-126.
[11] Kobsa, A., Müller, D. and Nill, A. KN-AHS: An adaptive hypertext client of the user modelling system BGP-MS. *Proc. UM 1994*, 31-36.
[12] Specht, M., Kravcik, M., Klemke, R., Pesin, L. and Hüttenhain, R. Adaptive Learning Environment for Teaching and Learning in WINDS. *Proc. 2002 Conference on Adaptive Hypermedia and Adaptive Web Based Systems*.
[13] Specht, M., Weber, G., Heitmeyer, S. and Schöch, V. AST: Adaptive WWW-Courseware for Statistics. *Proc UM 1997*.
[14] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel, S. *A Pattern Language* (Oxford University Press, 1977).
[15] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley, 1994).
[16] Buschmann F., Meunier, R., Rohnert, H., Sommerlad P. and Stal, M. *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns* (John Wiley & Sons, 1996).
[17] Schmidt, D., Stal, M., Rohnert, H. and Buschmann, F. *Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects* (John Wiley & Sons, 2000).
[18] Rossi, G., Garrido, A. and Carvalho, S. Design Patterns for Object-Oriented Hypermedia Applications. In Vlissides, J., Coplien, J., Kerth, N. (eds.) *Pattern Languages of Program Design, 2* (Addison-Wesley, 1996) 177-191.
[19] Fowler, M. *Analysis Patterns: Reusable Object Models* (Addison-Wesley, 1996).
[20] Eriksson, H. and Penker, M. *Business Modelling with UML-Business Patterns at work* (John Wiley & Sons, 2000).
[21] HCI design patterns web site <http://www.hcipatterns.org/>
[22] Borchers, J. *A Pattern Approach to Interaction Design* (John Wiley, 2001).
[23] Erickson, T. Towards a Pattern Language for Interaction Design. In Luff, P., Hindmarsh, J. and Heath, C. (eds) *Workplace Studies: Recovering Work Practice and Informing Systems Design* (Cambridge University

Press, 2000).

[24] Tidwell, J. Common Ground: A Pattern Language for Human-Computer Interface Design. http://www.mit.edu/~jtidwell/common_ground.html

[25] Weiss, M. Patterns for Web Applications. *Proc. PLOP 2003*

[26] Martin, D., Rouncefield, M. and Sommerville, I. Applying patterns of cooperative interaction to work (re)design: e-government and planning. *Proc. CHI 2002*, 235-242.

[27] Graham, I. *A Pattern Language for Web Usability* (Pearson Education, 2002).

[28] Seffah, A. Learning the ropes: human-centered design skills and patterns for software engineers' education. *Interactions*, 10(5), 2003, 36-45.

[29] Pedagogical Patterns Project site, <http://www.pedagogicalpatterns.org/>.

[30] Avgeriou, P., Papasalouros, A., Retalis, S. and Skordalakis, E., Towards a Pattern Language for Learning Management Systems, *Educational Technology and Society Journal*, 6(2), 2003, 11-24.

[31] E-LEN project. <http://www.tisip.no/E-LEN/>

[32] ADAPT project web site <http://www.wis.win.tue.nl/~acristea/HTML/Minerva/index.html>

[33] F. Halasz and M. Schwartz. The Dexter Reference Model. *Communications of the ACM*, 37(2), 1994, 30-39.

[34] Object Management Group (The), Unified Modeling Language Specification, Version 1.5, March 2003.

[35] Bass, L., Clements, P. and Kazman, R. *Software Architecture in Practice* (Addison-Wesley, 1998).

[36] Tzanavari, A., Avgeriou, P. and Vogiatzis, D. Patterns for Adaptive Web-based Educational Systems, *Technical Report TR-2003-4*, Department of Computer Science, University of Cyprus, December 2003.

[37] Papanikolaou K.A., Grigoriadou M., Kornilakis H., and Magoulas G.D. INSPIRE: an INtelligent System for Personalized Instruction in a Remote Environment. *Proc. Third Workshop on Adaptive Hypermedia, LNCS 2266*, 2002, 215-225.

[38] Vassileva J. Dynamic Courseware Generation, *Communication and Information Technologies*, 5(2), 1997, 87-102.

[39] Specht, M. and Oppermann R. ACE - Adaptive Courseware Environment. *The New Review of Hypermedia and Multimedia*, 4 (1), 1998, 141-161.

[40] Garrido, A., Rossi, G. and Schwabe, D. Pattern Systems for Hypermedia. *Proc. PLOP 1997*.

[41] Mezaros, G. and Doble, J. A Pattern Language for Pattern Writing. *Proc. PLOP 1998*, 529-574.

[42] IMS Learner Information Package Specification, <http://www.imsglobal.org/profiles/>

Acknowledgement

This work was supported by the "E-LEN: 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).

About the authors



Dr. Paris Avgeriou is an ERCIM (www.ercim.org) Research Fellow at the Software Engineering Competence Center, University of Luxembourg and the Fraunhofer Gesellschaft Institute, in Darmstadt, Germany. He received a diploma (M.Sc.) in Electrical and Computer

Engineering (1999), as well as a Ph.D. in Software Engineering (2003) from the National Technical University of Athens (NTUA), Greece. He has worked as a Visiting Lecturer at the Department of Computer Science (Jan. 03 - Jan. 04), University of Cyprus, and as a research and teaching assistant at NTUA (Sep. 99 – Dec. 02). He is a member of editorial boards for journals and conference program committees. He is a founding member of the World-Wide Institute of Software Architects and the president of its Greek chapter. His research interests concern the areas of software engineering with emphasis on software architecture, patterns, quality assurance, web engineering and mobile clients. His research is mainly applied in the domain of e-learning.



Dr. Dimitrios Vogiatzis is a Visiting Lecturer at the Department of Computer Science, University of Cyprus. He holds a degree in Computer Science from the Department of Computer Science, National & Capodistrian University of Athens, Greece; an M.Sc. degree in Information Technology Knowledge Based Systems from

the Department of Artificial Intelligence, University of Edinburgh, Scotland; and a Ph.D. diploma from the Department of Electrical and Computer Engineering, National Technical University of Athens, Greece. His research interests lie in the field of Computational Intelligence, in particular in rule extraction from neural networks, and in applications of Computational Intelligence techniques in bioinformatics.



Dr. Aimilia Tzanavari is a Visiting Lecturer at the Department of Computer Science, University of Cyprus. She holds the degrees of M.Sc. in Advanced Computing and Ph.D., both from the University of Bristol, United Kingdom. Prior to that she received a B.Sc. in Information Engineering from the Department of Informatics,

Technological Educational Institute of Athens, Greece.

She has worked at the National Centre of Public Administration in Athens, Greece (1997) and at the University of Bristol, UK as a lab instructor for various courses (1998-2001) and as a research assistant (1998-1999) in the AI Group. During the academic year 2001-2002 she was a Visiting Assistant Professor at Miami University, OH (USA). Aimilia has been a Visiting Lecturer at the Department of Computer Science of the University of Cyprus, CYPRUS, since September 2002. During December '02 and January '03 she was a BT (British Telecom) fellow. Her research interests include human computer interaction and user modelling.



Dr. *Symeon Retalis* is Assistant Professor at the Department of Technology Education & Digital Systems, University of Piraeus. He holds an M.Sc. degree in Information Technology - Knowledge Based Systems from the Department of Artificial Intelligence, University of Edinburgh, Scotland, and a Ph.D.

diploma from the Department of Electrical and Computer Engineering, National Technical University of Athens, Greece. His research interests lie on the development of web-based learning systems, design of adaptive hypermedia systems, web engineering, and human computer interaction. He has participated in various European R & D projects such as ELEN, MENU, UNIVERSAL, etc. He serves in the editorial board of international journals such as Computers in Human Behavior, IEEE Journal of Educational Technology and Society, ACM Computing Reviews, Journal of Information Technology Education. He participates to the ACM Web Engineering special interest group, to the CEN/ISSS learning technologies workshop. His publication list contains more than 70 items.