Designing Patterns for Adaptive or Adaptable Educational Hypermedia: a Taxonomy

Alexandra Cristea[(1)] and Franca Garzotto[(2)]

(1) Department of Computer Science, Eindhoven University of Technology
P.O. Box 513, Eindhoven, The Netherlands
a.i.cristea@tue.nl

(2) HOC - Hypermedia Open Center
Department of Electronics and Information, Politecnico di Milano
Via Ponzio 34/5 - 20133 Milano, Italy
garzotto@elet.polimi.it

Abstract: The design of an adaptive or adaptable educational hypermedia (AEH) is a complex process. Beside the design activities of "conventional" educational hypermedia, it involves a number of other tasks - the design of the *learner model*, of the user model *detection mechanisms*, of the *adaptation capabilities* (to dynamically customize the application according to the learner's model), of the *user control* over customization. Patterns can significantly help mastering the complexity of AEH design. As a preliminary steps towards the definition of a pattern language for AEH, we have identified and structured the typical problems a designer may face during each AEH design task. From the problem analysis and classification we have defined a taxonomy for AEH patterns, which can be used as a conceptual framework for discovering patterns and for structuring a pattern language in the AEH field. The paper reports our approach, discusses our taxonomy, and presents examples of AEH design patterns.

Introduction

According to architect Christopher Alexander (Alexander et al., 1977), a design 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, without ever doing it the same way twice". Although this definition came from architecture, the concept of pattern is now exploited in many other fields. In instructional design, for example, *pedagogical patterns* (www.pedagogicalpatterns.org) try to capture expert knowledge of the practice of teaching and learning in a compact form that can be easily communicated to those who need that knowledge. In computer science, design patterns originated from the need of systematically recording the experience of object oriented systems designers, in order to make it *reusable* and *sharable* in the software engineering community (Gamma 1996). In computer science, patterns have later evolved to capture the design practice in specific classes of software systems, e.g., hypermedia (Bolchini et al, 1999), or specific application domains, e.g., e-commerce (Rossi 1999), and are generally acknowledged as means to improve the quality of any software product while reducing the development cost.

In this paper, we focus on design patters for *adaptive/adaptable educational hypermedia (AEH)*. Adaptive/Adaptable hypermedia (Brusilovsky, 2001) are characterized by the capability of dynamically customizing some application features (content, link structure, lay-out properties) according to the characteristics of the current user, or *user model*[1]. AEH are adaptive/adaptable hypermedia built for educational uses: their multimedia hyper-contents and customization capabilities are designed to provide learning experiences that are dynamically tuned according to the

[1] Adaptive hypermedia differ from adaptable hypermedia in the degree of user control on the user model. In adaptive hypermedia, the system dynamically infers the user model from the user behavior. In adaptable hypermedia, the user is explicitly involved in the definition and update of his/her user model. Many applications are "hybrid", i.e., offer a combination of adaptivity and adaptability. In this paper, AEH will denote either an adaptive or an adaptable hypermedia, or a "hybrid" system.

learners' characteristics.

The development of AEH is a complex process not only from an implementation but above all from a design perspective. The designer must understand which are the proper learner's characteristics to be represented in the system and how the system can detect them; he must model how the learner's characteristics map on the content, link structures, lay-out of the application; he must decide the level of control offered to the user and how the user can feel aware of the application behavior. Thus AEH design sums up the complexity of designing usable hypermedia structures with the complexity of designing adaptive capabilities that can offer an effective educational experience.

Although it is obvious that *design reuse* has the potential for significantly reducing this complexity, the concept of design patterns for AEH is almost unexplored. In this paper we present the initial results of our research in this field, carried on within the EC funded project ADAPT "Adaptivity and adaptability in ODL based on ICT" (Socrates Program). We discuss the *taxonomy* we have defined for AEH design patterns and our procedure to identify usable patterns, and introduce some of the patterns we have discovered so far.

AEH design tasks and key problems

Our research has started with the identification of the crucial tasks involved in the AEH design process, and of the issues addresses by each task. This analysis allows us to define the *problem space* for AEH design, i.e., the universe of problems that a designer may need to face. Since a pattern, in its simplest form, is a pair <design problem, design solution>, the structure of the problem space for AEH design provides the categories for classifying patterns.

In order to identify the main design problems involved in the design of AEH , we have first analysed the design tasks and issues that are involved in a hypermedia application without considering a specific domain. Then we have compared them with the design issues of educational hypermedia and of AEH .

Designing an AEH is significantly more complex than designing a non-adaptive/non- adaptable educational system, which is basically an education-oriented specialization of the traditional design tasks defined in hypermedia/web engineering (Garzotto 1995). In "conventional" (educational) hypermedia, the user always has the control on the sequence of visited "pages". Content and links on the pages are statically defined at design time. Different educational strategies, if needed, are be supported "simultaneously", leaving to the user the burden of exploring the contents and links more appropriate to his learning needs. In contrast, in AEH the system has (part of) the control on the sequence of visited "pages" and on their content, e.g., by dynamically hiding or showing links and contents depending on the current state of the *learner model* (which is build by the system in *adaptive* systems and is under the user control in *adaptable* systems) and the instructional strategy that is more appropriate for his profile.

The high level design tasks and problems that the AEH designer must address concern: *Educational Information Design, Learner model (LM), Instructional strategy (IS), Instructional View (IV), Detection mechanism (DM), Adaptation mechanism (AM).* While the choice of the types of content and relationships that the application should provide (educational information design) is an issue of any educational application (and will not be discussed here), all others are specific of AEH. AEH can support multiple instructional strategies and dynamically apply them according to the specific learner, by mapping the learner model characteristics on a different *educational view*, i.e., a different content and navigational schema. In addition, AEH must provide some mechanism to detect user model parameters, either explicitly specified by the user, or derived from user's inputs (e.g., via questionnaire) or deduced from the user behavior. Finally, AEH must provide some mechanisms to relate user attributes to customization effects, through a proper adaptation mechanism.

The following table highlights the key design tasks and issues in AEH design, pinpointing the increasing degree of complexity from hypermedia design to educational hypermedia design to AEH design.

Application type	**Requirement Issues**	**Design Tasks**	**Key Design Issues**
Hypermedia	• Context (e.g., device characteristics, situation of use, …) • Stakeholder's goals and needs • ….	• *Information design* (information resources – content and relationships - about the application domain that are relevant for the stakeholders goals and needs) • *Navigation Design* (navigation paths to explore contents and relationships) • *Lay-out design* (presentation properties)	Finding a *design compromise*: Building content, navigation, and lay-out structures that match the requirements of all possible users
Educational Hypermedia	• Context (e.g., device , learning situation, …) • *Teacher*'s and learner's goals • *Instructional/ Educational Strategy(ies)* (teaching style, e.g., problem solving, case-based, progressive/ sequential, "global" …) • …..	• *Educational information design* (information resources – content and relationships - focused on educational needs) • *Navigation design* • *Lay-out design*	- Defining the educational content, navigation and presentation structures that are more appropriate to meet the teacher's goals, educational strategy, teaching style - In case of multiple teaching and learning goals and styles, finding a design compromise to match all of them
Adaptive Educational Hypermedia	• Context (e.g., device , learning situation, …) • *Teacher*'s goals • *Instructional/ Educational Strategies* (teaching style, e.g., problem solving, case-based, progressive/sequential, "global" …) possibly associated to different learning styles • ….	• *User model* design (the representation of the user characteristics) • *Educational information design* (information resources – content and relationships - focused on educational needs) • *Instructional Views Design*: creating different navigational views (i.e., content and navigation structures defined at schema level) tuned to specific educational needs and instructional strategies • *Detection mechanism* (how the user model is built and updated) • *Adaptation mechanism* (which modifications on the content and link structures are needed by different user models)	- Defining the relevant learner attributes to be represented in the user model - Defining the instructional strategies that are more appropriate to the different learners, and the corresponding instructional views - Defining which specific contents or links are more appropriate for other user attributes - Defining how user attributes can be induced from the user behaviour (for adaptive hypermedia) - Defining which user attributes are explicitly provided by the user, and how (for adaptable hypermedia)

AEH Patterns: Taxonomy and Examples

The analysis of design tasks and key issues discussed in the previous session allows us to identify a set of "design variables" or "problem classes" - *Learner model (LM), Instructional strategy (IS), Instructional View (IV), Detection mechanism (DM), Adaptation mechanism (AM)* - that offers us the guideline to define a AEH pattern taxonomy. In our approach, a pattern category corresponds to a class of design problems, as shown in the following table. In some cases, we refine the general problem addressed in a pattern class in terms of sub-problems that have a direct counterpart in AEH patterns (as discussed later in this section). In other cases, the various subproblems may be further refined before associations with patterns. In addition, it is important to notice that, in general, a single problem (at the leaf level in the problem hierarchy) may offer multiple solutions, which can be modelled as a single pattern or as multiple patterns.

Pattern Category	**General problem**	**Sub-problems /Sub-categories of patterns**
Learner model (LM) definition	What relevant *learner* attributes are captured by the learner model?	- Which learner attributes are useful to model the user in an educational context? - How can the learner model be represented?
Learner model detection Mechanism (DM)	How can the system define the learner model and detect its updates?	- Are learner model updates fully under the system control (adaptivity) and how? - Are learner model updates fully under the user control (adaptability) and how? - Are there some learner attributes which are updated by the systems and others that are updated explicitly by the user? How? - Under which circumstances is one approach preferable to the other?
Instructional strategy (IS) definition	Which is the most appropriate instructional strategy of the different users?	- Which are the teaching goals addressed by the different instructional strategies? - Which instructional strategies better match which user characteristics?
Instructional view (IV) definition	How can educational views be defined?	- Which educational contents are more appropriate for a given educational strategy? - Which relationships among contents are more appropriate for a given educational strategy? - Which navigation/ presentation structures are more appropriate for each instructional strategy?
Adaptation mechanism (AM) definition	How does the system customize its features according to user model updates?	- How different user model attributes correspond to into different hypermedia properties of the application? In particular: - Which learner attributes determine specific instructional views, i.e., specific hypermedia design *schemas*? - Which learner attributes require different *instances* of information or navigation structures or lay-out features ? - How customisation effects are made evident to the user? - Once a user model is detected, what is the degree of user control on the customisation?

In the rest of this section we discuss some examples of patterns for some of the above classes. The generally acknowledge structure of patterns is usually composed by <pattern name, problem, solution, context, forces, related patterns, known uses>, for lack of space we formulate our patterns in the minimal form <pattern category, name, problem, solutions>.

Category	LM (Learner model definition)
Pattern name	Learner attribute types
Problem	Which are some classes of user attributes that are useful to model for educational and customisation purposes?
Solution	One useful classification of learner attributes distinguish between *stable* learner's attributes (i.e., those having no or a low degree of variability during a session of use) and *variable* learner's attributes (i.e., those having a significant degree of variability during a session of use). • *Stable* learner attributes may include: User's *Intrinsic characteristics* (e.g., personality, sex, physical and psychological abilities or disabilities*); Learning style* (the user preferences on how to perceive and process information); *Learner's background knowledge/ experience/ practice* outside the application domain; *Learner's goals; Amount and type of scaffolding required* • *Variable* learner attributes may include: *Learner's knowledge* about "concepts/topics" the application deals with; *Learner's task; Time of use* per (set of) concept(s); *Learning Performance* (measured though assessment) In general, the instructional strategy is defined based on the stable attributes, and so is the instructional view. Variable attribute may impact on finer-grained features of the application (i.e., specific instances of contents and links)

Category	DM (Learner Model Detection Mechanism)
Pattern name	User-driven vs. system-driven modelling control
Problem	Which learner attributes should be detected by the system and which should be under the user control?
Solution	Obviously, all learner attributes that the system cannot infer must be under the user control (e.g., the user's intrinsic characteristics). In general, *stable* parameters are more difficult to be detected on the basis of the analysis of the user interaction with the system: they can be either *explicitly asked* to the user (e.g., a learner inputs his learning preferences, or switches to a different learning style) or derived from indirect data provided by the user. Learning style for example can be captured by: - Completing a pre-test questionnaire that is related to the learner interests or attitudes, knowledge level, etc., from which the application *derives* the learning preferences or style. - Completing a learning activity (e.g. a self assessment exercise). The *Amount and type of scaffolding required*, although a stable attribute, can be calculated by the system, based on some educational "assumptions" (e.g., after "some" amount of time, the degree of scaffolding need tends to decrease). *Variable* learner attributes such as *Learner's knowledge* about "concepts/topics" the application deals with, *Time of use* per (set of) concept(s), and *Learning Performance* (measured though assessment) are typical attributes that are detected automatically from the user behavior. *Learner's task* may be updated by the system based on th learning flow and on some instructional "principles" or strategies that define "a priori" the most appropriate task flow for given types of users and is automatically updated once the task is "completed".

Category	AM (Adaptation Mechanism)
Pattern name	Information Overview
Problem	How can we help the user to summarize what he has already read /studied and speed up his/her work of revisiting previously studied material?
Solution	When the learner revisits a page already "read" and returns to that page during the session, provide him(her an *overview* of the page, which presents the core concept and omits detailed information. Include a {visiting state} associated to pages among the attributes in the learner model (which is set to "read", for example, when the user has spent "enough" time on the page)

Category	AM (Adaptation mechanism)
Pattern name	Link Hiding
Problem	Assuming that the user has a preference for a sequential learning style, how can we help him to focus on the most relevant material first, and explore additional contents only after he has enough knowledge about a topic?
Solution	When the learning style in the Learner model is "sequential", provide the learner only with links to material that is strictly compatible with his knowledge level. At the beginning of a a learning process, show only very few links, and increase their number as the learner's knowledge about the topic proceeds. Include in the learner model a representation of the knowledge level and in the adaptation mechanisms rules for conditional link inclusion.

Conclusions

This paper has presented the preliminary results of our research of a pattern language for Adaptive or Adaptable Educational Hypermedia. We have first identified the main tasks involved in AEH design, and the problems that a designer needs to face during each task. From the classification of design problems, we have identified a taxonomy for AEH design patterns, and a preliminary set of design patterns for AEH. Our work is based on the on-going analysis of different existing AEH systems (Brusilowsy 2003, Cristea 2002, Cristea 2003, Stash 2003) and on a survey of AEH design experiences. The survey involves AEH developers who are asked, through a questionnaire, to validate our design taxonomy and to report the design solutions for the different kinds of problems. Finally, we are also exploring existing approaches to design patterns in the field of traditional (i.e., non computer-mediated) education, by analyzing the patterns developed in the Pedagogical Patterns Project (www.pedagogicalpatterns.org). Our belief is that some of these patterns may have their counterpart not only in e-learning applications in general, but also in AEH .

Acknowledgement

This research is linked to the EC Socrates Minerva project ADAPT: "Adaptivity and adaptability in ODL based on ICT" (101144-CP-1-2002-NL-MINERVA-MPP) http://wwwis.win.tue.nl/~acristea/HTML/Minerva/index.html

References

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. & Angel, S. (1997). *A Pattern Language*. Oxford University Press, New York.

Brusilovsky, P. (2003). Developing adaptive educational hypermedia systems: From design models to authoring tools. In: T. Murray, S. Blessing and S. Ainsworth (eds.): *Authoring Tools for Advanced Technology Learning Environment.* Dordrecht: Kluwer Academic Publishers.

Brusilovsky, P. (2001). Adaptive hypermedia, *User Modeling and User Adapted Interaction*, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2) 8 (pp. 7-110).

Cristea, A. & De Mooij, A. (2003). Adaptive Course Authoring: MOT, My Online Teacher. *Proceedings of ICT-2003*, IEEE LTTF International Conference on Telecommunications, "Telecommunications + Education" Workshop (Feb 23 - March 1, 2003 Tahiti Island in Papetee - French Polynesia).

Cristea, A.I. & Aroyo, L. (2002). Adaptive Authoring of Adaptive Educational Hypermedia, *AH 2002, Adaptive Hypermedia and Adaptive Web-Based Systems*, (pp. 122-132) LNCS 2347, Springer.

Garzotto, F., Paolini, P., Bolchini, D. & Valenti , S. (1999). Modeling-by-patterns of web applications. In Advances in Conceptual Modeling, P. Chen. D. Embley, J, Kouloumdjian, S. Little (eds.), *Lecture Notes in Computer Science 1727*, Springer.

Garzotto, F., Mainetti, L. & Paolini, P. (1995). Hypermedia Design, Analysis, and Evaluation Issues". In *Communications of the ACM, 38* (8), Aug.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1996). *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley.

G. Rossi, D. Schwabe and F. Lyardet, "Improving Web Information Systems with Design Patterns". In *Proc. of the 8th International World Wide Web Conference*, Toronto (CA), May 1999, Elsevier Science, 1999

Stash, N. & De Bra, P. (2003). Building Adaptive Presentations with AHA! 2.0. *PEG'03* Conference, St. Petersburg, Russia, June. WHURLE, http://whurle.sourceforge.net/