

Authoring and Management Tools for Adaptive Educational Hypermedia Systems: The AHA! Case Study

Paul De Bra¹, Natalia Stash¹, David Smits¹, Cristóbal Romero²,
Sebastián Ventura²

¹ Eindhoven University of Technology (TU/e),
PO Box 513, Eindhoven, The Netherlands
{debra}@win.tue.nl

² Córdoba University, Campus Universitario de Rabanales,
14071, Córdoba, Spain
{cromero, sventura}@uco.es

Summary. Creating and maintaining adaptive educational applications is hard work for teachers and developers. In order to help the author perform these tasks the e-learning systems must provide authoring and management tools. In this chapter we describe several useful tools for working with adaptive educational hypermedia systems, using the Adaptive Hypermedia Architecture (AHA!) system. AHA! is a well-known open source general-purpose adaptive hypermedia system. In the current AHA! distribution versions there are some general adaptive author tools as Concept Editor, Graph Editor, and Form Editor, all accessible through the overall Application Management Tool. There is also a specific educational tool: the Test Editor (and the associated Test Engine) and we are now developing some others such as a Course Editor and Mining tool. In this chapter we describe the AHA! system and the functionality of each of these authoring and management tools intended to help teachers and application developers.

1 Introduction

In recent years, we have seen an explosive growth in the use of web-based technology in distance learning systems. At the same time, more and more artificial intelligence techniques have been integrated into these systems to improve students' learning, turning them into Intelligent Tutoring Systems.

The union of web-based hypermedia with Intelligent Tutors has led to the current Web-based Adaptive Educational Hypermedia Systems that allow adapting the teaching to each individual student [6]. Adaptive Educational Hypermedia (AEH) is an alternative to the traditional one-size-fits-all approach in web-based education. It combines the concepts of hypertext/hypermedia with user modeling and user adaptive systems to adapt the hypertext to the needs of each particular student. Adaptive Hypermedia Systems (AHS) [4] started to appear around 1990, when researchers combined the concepts of hypertext/hypermedia with user modeling and user adaptation. The first and foremost application of adaptive hypermedia was in education, where the navigational freedom of hypermedia was introduced into the area of intelligent tutoring systems. But since then applications in information systems, information retrieval and filtering, electronic shopping, recommender systems, etc. have been realized. The advent of the Web has made the use of (basic) hypermedia facilities easier, through the use of HTML. However, creating adaptive hypermedia on the Web requires server-side functionality for user modeling and for the adaptive generation of (HTML) pages. Until recently, almost every adaptive hypermedia application was based on a special-purpose (server-side) system. The development of adaptive hypermedia applications and systems has had a one-to-one relationship. This has seriously hindered the development of interesting new adaptive applications by researchers with insufficient skills or financial means to develop their own adaptive hypermedia system. Despite the potential benefits of adaptive educational hypermedia for educational applications, the use of adaptive hypermedia has not been as widely accepted as might be expected. One of the reasons might be in relation to difficulties in authoring. So, it is necessary to provide authoring and management tools in order to help the teacher or application developer in performing a multitude of authoring tasks.

There are several modern authoring systems to develop adaptive educational hypermedia [12] such as InterBook, Web DCG, DCG+GTE, AHA!, ACE, ALE, NetCoach/ART-WEB, ECSAIWeb, MetaLinks, SIGUE, etc., all oriented to educational practitioners. There are two major approaches used by authoring tools: the markup approach and GUI (Graphic User Interface) approach. The markup language uses a regular word processor with the help of special markup language. The GUI uses special authoring user interfaces. In general, it could be a command-based, form-based or a direct manipulation interface, but all existing AHS authoring tools use form-based GUI. This kind of interface provides very good support for the