AHA! The Next Generation

Paul De Bra, Ad Aerts, David Smits, Natalia Stash
Department of Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
+31 40 2472733
debra@win.tue.nl

ABSTRACT
AHA! is a simple Web-based adaptive engine, that was originally developed to support an on-line course. This paper describes AHA! version 2.0, a new major release that aims to significantly increase the adaptive versatility of AHA! without sacrificing AHA!'s simplicity that makes it easy to use. The new features in AHA! are inspired by AHAM [4], a Dexter [6] based reference model for adaptive hypermedia systems.

ACM Categories
H5.2 (user interfaces), H5.4 (hypertext/hypermedia)

General Terms: Design, Experimentation, Human Factors

Keywords: Adaptive hypermedia, condition-action rules, adaptation engine.

1. INTRODUCTION
In 1994 we started a course on Hypertext, consisting of lectures and a Web-based course text. From 1996 on the lectures were discontinued and the course text was augmented with *adaptive content and linking*. Our adaptive software later became the AHA! system [3, 5], for **A**daptive **H**ypermedia **A**rchitecture. AHA! was made available as Open Source. Its simple architecture (based on Java Servlets, and XML and HTML for the pages) made it possible to add adaptation to very different applications. AHA! has been studied and used independently by several researchers/ educators from different parts of the world.

In 1999 we developed a reference model for adaptive hypermedia, called AHAM (Adaptive Hypermedia Application Model) [4,7]. Based on the formal Dexter model for hypermedia [6] AHAM provides a framework to express the functionality of most adaptive hypermedia systems (or ahs). See [3] for a recent overview of existing ahs and adaptive features. AHAM provided most of the inspiration for the additions to AHA! that have resulted in version 2.0, described in this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
HT'02, June 11-15, 2002, College Park, Maryland, USA.
Copyright 2002 ACM 1-58113-477-0/02/0006…$5.00.

In this paper we shall only describe AHA! 2.0 features and properties. Information on older versions can be found in previous papers about AHA!, including [5]. Different versions of AHA! and documentation about them can be found at the Website http://aha.win.tue.nl/.

2. BRIEF OVERVIEW OF AHA!
AHA! is a typical adaptive Web-based system, in the sense that each time the user requests a page (by following a link) some server-side software is activated that performs the following steps:

1. The requested page is retrieved from the local file system or from a remote source. (AHA! can load pages from other web servers through HTTP and adapt them just like local pages.)
2. The system loads a *domain model* of the application (or has it in memory already). This model reveals how the requested page relates to other pages or to higher level *concepts*.
3. The system loads a *user model* (or has it in memory). This model contains some of the user's aspects such as preferences, and an *overlay model* that shows how the user relates to the pages and concepts of the domain model. In AHA! the user model consists of a set of attribute/value pairs for each concept or page. The author can define and name arbitrarily many attributes (of type string, integer or Boolean).
4. The system loads an *adaptation model* (or has it in memory). It contains rules that are used to update the user model based on the access to the requested page, and rules to adapt the presentation of the page to the individual user. The rules are explained in Section 3.
5. The system executes the rules, and thus updates the user model and presents the adapted page. The rule execution is explained in Section 4.

The AHA! system deviates from this general scheme (dictated by the AHAM model) in one detail: the *domain model* and *adaptation model* are stored together.

According to Brusilovsky [3] there are two types of adaptation in ahs: *adaptive presentation* and *adaptive navigation support*. In AHA! these are implemented as follows:

- Fragments in a page can be *conditionally included*. This is most useful to add (short) extra explanations for users who need them, or to remove unwanted elements from lists.
- Links (actually, link anchors) can be shown in different colors. AHA! uses three colors, called *good*, *neutral* and *bad*. The color of a link anchor depends on the *desirability* of the link, which in turn represents the desirability of the

destination (page) of the link. This desirability is expressed using the same kind of *condition* as used for the conditional inclusion of fragments. The *good* links, leading to desired pages, are shown in *blue* by default. The *neutral* color, *purple* by default, is used for links to desired pages that were visited before. The *bad* color, *black* by default, is used for links to *undesired* pages. AHA! disables the underlining of links, so link anchors with the same color as normal text are effectively *hidden*. The color scheme can be changed by the end-user, to make the *bad* links visible if the user wants this.

3. THE DOMAIN/ADAPTATION MODEL

Concepts and pages and their associated adaptation rules are represented using an XML file (stand-alone version) or a mySQL database table (server version). We give a part of a tiny imaginary example, in which a system has information about chocolate and beer. We use an alternative syntax (more compact than AHA!):

```
<concept>
   <name>de-koninck</name>
   <desc>Beer from Antwerp</desc>
   <requirement>beer.interest > 20</requirement>
   <attribute name="access" type="bool" isPersistent="false">
      <desc>standard attr: true when page accessed</desc>
      <generate>
         <requirement>beer.interest < 100</requirement>
         <trueAction>beer.interest += 10</trueAction>
      </generate>
      <generate>
         <requirement>chocolate.interest >= 5 and
               chocolate.interest < 50</requirement>
         <trueAction>chocolate.interest -= 5</trueAction>
      </generate>
   </attribute>
   <attribute name="interest" type="int" isPersistent="true">
   …
</concept>
```

The concept "de-koninck" corresponds to a page about the first author's favorite beer. This page is only *desirable* if the user's interest in beer is already greater than 20. (Links to the page will be *bad* if the interest is too low and *good* or *neutral* otherwise.) Each page has a volatile attribute, called "access" that is false by default and becomes true temporarily when the page is accessed. Within certain limits ready about de-konink raises the interest in beer and decreases the interest in chocolate. The "interest" of de-koninck is a persistent attribute, meaning that it is permanently stored in the user model. It too can have *conditional actions* to update attributes of other concepts. Updates can be based on constants (as in the example) or on the update to the rule's attribute. (Instead of the constant amount 10, a part of a change in interest in de-koninck can be *propagated* to beer.interest.)

4. THE AHA! ADAPTATION ENGINE

The adaptation rules in AHA! are *condition-action* rules, like rules studied in the field of active databases [1]. The adaptation engine maintains a queue of pending rule instantiations. Whenever an attribute value of some concept is modified the adaptation rules for which the requirement (condition) is true (at that time) are added to the queue. When a user accesses a page (by following a link) the "access" attribute for that page becomes true. The adaptation rules associated with this attribute are the first ones examined and put in the queue (if their condition is true). The process of rule executions (and adding more rules to the queue) continues until the queue is empty and thus there are no more rules to execute.

From research in active databases [1] we know that the rule execution process is not guaranteed to *terminate*. In [7] we showed how such problems can be detected at authoring time. (AHA! currently does not yet warn authors about this.)

5. CONCLUSIONS / FUTURE RESEARCH

The "new" AHA! user model and adaptation engine greatly improve the versatility of AHA!. Especially arbitrary adaptation rules (unlike simple propagation of knowledge in educational applications) are easier to express, as the example in section 3 has shown. The extensions are based on the rule system presented in [7] for the AHAM reference model.

In the future we want to extend AHA! by borrowing more ideas from the AHAM:

- links to concepts (not just pages), and a method to "select" the best page to present when following such a link;
- a way to express *generic* rules, so that rules don't need to be replicated for every page or concept they apply to.

6. ACKNOWLEDGEMENT

The development of the AHA! system is supported by a grant of the NLnet Foundation, through the "Adaptive Hypermedia for All!" project (conveniently abbreviated to AHA!).

7. REFERENCES

[1] E. Baralis, and J. Widom. An algebraic approach to static analysis of active database rules. ACM Transactions on Database Systems, Vol. 25, nr. 3, pp. 269–332, 2000.

[2] Brusilovsky, P. Adaptive Hypermedia, User Modeling and User-Adapted Interaction, Vol. 11, nr. 1–2, pp. 87–110, Kluwer academic publishers, 2001.

[3] De Bra, P., A. Aerts, G.J. Houben, and H. Wu. Making General-Purpose Adaptive Hypermedia Work. Proceedings of the AACE WebNet Conference, pp. 117–123, San Antonio, Texas, 2000.

[4] De Bra, P., G.J. Houben, and H. Wu. AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. Proceedings of the ACM Conference on Hypertext and Hypermedia, pp. 147–156, Darmstadt, Germany, 1999.

[5] De Bra, P. and Calvi, L., AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, vol. 4, pp. 115-139, Taylor Graham Publishers, 1998.

[6] Halasz, F., and M. Schwartz. The Dexter Hypertext Reference Model. Communications of the ACM, Vol. 37, nr. 2, pp. 30–39, 1994.

[7] Wu, H., E. De Kort, and P. De Bra. Design Issues for General-Purpose Adaptive Hypermedia Systems. Proceedings of the ACM Conference on Hypertext and Hypermedia, pp. 141–150, Århus, Denmark, 2001.