AHA! meets AHAM

Paul De Bra, Ad Aerts, David Smits, and Natalia Stash

Department of Computer Science
Eindhoven University of Technology
PO Box 513, NL 5600 MB Eindhoven
The Netherlands
debra@win.tue.nl

Abstract. After many years of hypertext research, the Dexter model was defined [7] to capture the features of most existing hypertext systems in a single, formal reference model. Likewise, the AHAM model [5] (based on Dexter) describes most features that are found in adaptive hypermedia systems (ahs). In the AHA! project funded by the NLnet Foundation we are extending the simple AHA system [4, 6] with the condition-action rules that were introduced in AHAM [8]. This results in a more versatile adaptation engine, with more intuitive and more powerful rules**.**

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 [4,6], for **A**daptive **H**ypermedia **A**rchitecture. Also in 1996, Peter Brusilovsky published a survey article [2] on adaptive hypermedia systems (ahs). This article was updated in 2001 [3] to include more recent developments in this research field. The overview articles show (informally) which kinds of features are offered by the different systems. In 1999 we developed a reference model for adaptive hypermedia, called AHAM (Adaptive Hypermedia Application Model) [5]. Based on the formal Dexter model for hypermedia [7] AHAM provides a framework to express the functionality of any ahs. Hence, AHAM is not the model behind the AHA! system (as some people have asked us in the past) but rather the model behind almost every ahs. In [8] we studied the adaptation rules and the behavior of adaptive engines that can be expressed in AHAM. In this paper we describe our current efforts to bring the simple AHA! system closer to the powerful adaptation systems that AHAM suggests. This effort is part of a project, funded by the NLnet Foundation, to turn AHA! into a general-purpose (Web-based) ahs.

In Sect. 2 we briefly describe the overall architecture of the AHA! system. Section 3 discusses the new rule system and adaptive engine. Section 4 provides an outlook to the future of AHA!.

2 The AHA! System

According to Brusilovsky [2] there are two types of adaptation in ahs: *adaptive presentation* and *adaptive navigation support*. Each time the user visits a page a *user model* is updated. This user model is used to determine how the presentation of the "next" page should be adapted and which advice should be given to the user regarding the links to follow next. While some systems take sequences of pages into account, and possibly even the behavior of user groups, AHA! is one of the simpler systems that work on a page (access) by page basis. The main functions of AHA! are:

- For every user there is a user model that represents how the user relates to the Webpages and to (higher-level) concepts. Each time a user visits a page a set of *generate rules* determines how the user model is updated. We describe these rules below.
- Each page contains (zero or more) fragments that are conditionally included in the presentation. The requirement rules for inclusion are also described below. The conditional inclusion of fragments is AHA!'s main *adaptive presentation* technique.
- For each page there is a similar requirement rule. It is used for *adaptive navigation support*. AHA! uses different colors for link anchors to indicate the "desirability" of the links. Depending on the choice of colors AHA! can be configured to use the *link hiding* technique (with black undesirable links) or the *link annotation* technique (with all links shown in visible, different colors).

The AHAM reference model proposes a rich user model, with for each page or higher level concept a set of attribute/value pairs, and with an adaptation model based on condition-action rules. The (old) AHA! system on the other hand, as described in [4, 6], has a very simple user model with for each page or concept a single numerical attribute (called *knowledge*), and with simple rules, mostly intended to describe the propagation of knowledge from pages to sections to chapters. In this paper we describe how the "new" AHA! system adopts user model and adaptation model elements introduced in the AHAM model. The result is a new AHA! that tries to preserve its initial simplicity while making the adaptation more versatile.

The "new" AHA! system uses multiple (user model) attributes per page or concept. The author of an application can choose attribute names that match what attributes are used for. It thus becomes possible to indicate e.g. *interest in* a certain concept as well as *knowledge about* a concept. Also, the attributes can have Boolean, numeric or string values.

For every page and fragment there is a *requirement* to decide whether links to the page should be shown as "desirable" or whether the fragment should be included in the presentation. The requirement is a Boolean expression in (user model) attribute values for concepts (or pages). An example, using the non-existent website on Belgian products we used in [6]:

```
beer.interest > 70 and chocolate.interest < 30
```

While browsing the site, visits to pages about beer contribute to the system's confidence that the user is interested in beer and not in chocolate. So a user who is interested in beer can be shown additional information that is hidden for chocolate lovers.

For every page or concept there is a set of *generate* rules. Each rule has a *condition* that is checked to see whether the associated *action* should be performed. (It is also possible to specify an *alternate action* to be performed when the condition is not satisfied.) Each rule is associated with an attribute of the page or concept, and the rule is "triggered" by an update to that attribute. For a page there are also rules that are triggered by an access to the page. (This is treated as an update to an `access` attribute).

The following rule expresses that when the user reads a page the "knowledge" about a concept increases by an amount that depends on a condition. Assume that the interest in beer should be high to understand everything on the page about "Duvel":

```
C:    beer.interest > 70
A:    duvel.knowledge := 100
AA:   duvel.knowledge := 35
```

As another example, the following rule, to be associated with the `interest` attribute of `beer`, expresses that if the user shows a low interest in chocolate, then any increase of the user's interest in beer will induce a small (namely 20% of that increase) decrease in interest for chocolate. If the user has already shown great interest in chocolate then reading about beer no longer affects that.

```
C:    chocolate.interest < 50
A:    chocolate.interest -= 20
```

There is no limit to the number of (condition-action) rules that can be associated with an attribute of a concept.

AHA! uses XML to represent the *generate* rules, but that syntax is too verbose to write down in this short paper. Furthermore, the actual syntax is irrelevant as we are also developing authoring tools that will hide the syntax from the author, and (mySQL) database support that will use XML only for import and export of rule sets.

3 The AHA! Adaptation Engine

When a user accesses a page the *generate* rules for (the `access` attribute of) that page are executed (as if `access` was changed from 0 to 100). AHA! distinguishes between two kinds of updates that can be defined in actions:

- *absolute*: a value, specified in the action, is assigned to an attribute of a page or concept. (symbol := used above)
- *relative*: the specified value is used as a percentage of the triggering update to be added to or subtracted from the attribute; relative updates only make sense for numerical attributes of course. (symbol `+=` or `-=` used above)

The adaptation engine keeps track of which attributes of which pages or concepts are updated by rule actions, and "triggers" the execution of their associated rules. This process continues until 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* and may also not be *confluent* (meaning it may generate different results depending on the order in which triggered rules are executed). In [8] we showed how such problems can be detected at authoring time.

4 Conclusions / Future Work

The “new” AHA! user model and adaptation engine greatly improve the versatility of AHA!. The extensions are based on the rule system presented in [8] for the AHAM reference model. In the future we want to bring AHA! even closer to AHAM through at least the following new features:

- 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.

Also, authoring tools for the concept structure and the adaptation rules will be extended with static analysis methods to warn authors of potential *termination* and *confluence* problems.

Acknowledgement

The development of the AHA! system is supported by a NLnet Foundation grant, through “Adaptive Hypermedia for All!” project (conveniently abbreviated to AHA!). More information about AHA! can be obtained at the Website http://aha.win.tue.nl/.

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. Methods and Techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction, Vol. 4, pp. 1−19, Kluwer academic publishers, 1996.
3. Brusilovsky, P. Adaptive Hypermedia, User Modeling and User-Adapted Interaction, Vol. 11, nr. 1−2, pp. 87−110, Kluwer academic publishers, 2001.
4. 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.
5. 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.
6. De Bra, P., and J.P. Ruiter. AHA! Adaptive Hypermedia for All. Proceedings of the AACE WebNet Conference, pp. 262−268, Orlando, Florida, 2001.
7. Halasz, F., and M. Schwartz. The Dexter Hypertext Reference Model. Communications of the ACM, Vol. 37, nr. 2, pp. 30−39, 1994.
8. 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.