High-Level Translation of Adaptive Hypermedia Applications

Ewald Ramp[1,2], Paul De Bra[1], Peter Brusilovsky[2]

[1]Information Systems Group
Department of Computing Science
Eindhoven University of Technology

[2]Dept. of Information Science and Telecommunication
School of Information Sciences
University of Pittsburgh

e_ramp@hotmail.com, debra@win.tue.nl, peterb@mail.sis.pitt.edu

ABSTRACT

In the early years of the adaptive hypermedia research a large number of *special-purpose* adaptive hypermedia systems (AHS) have been developed, to illustrate research ideas, or to serve a single application. Many of these systems are now obsolete. In this paper we propose to bring new life to these applications by means of translation to a *general purpose* adaptive hypermedia architecture. We illustrate that this approach can work by showing a high-level translation from InterBook [2] to AHA! [5]. Such a translation consists of three parts: the structure of *concepts* and *concept relationships* needs to be translated, the *adaptive behavior* for these concept relationships must be defined, and the *layout and presentation* of the source application must be "simulated". Our high-level translation covers all three parts.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces And Presentation**]: Hypertext/Hypermedia – *Architectures, Theory.*

General Terms

Experimentation, Human Factors, Standardization, Languages.

Keywords

Authoring, adaptation, conversion.

1. INTRODUCTION

The field of hypermedia was revolutionized in the early 90's by the introduction of the World Wide Web. Before 1990 a large number of independently developed hypertext systems were developed, with functionality that could roughly be described using a common reference model like the Dexter Model [8]. After 1990 the field converged on using World Wide Web technology as a common basis for other developments.

In adaptive hypermedia [1] we see a similar, but twofold evolution. First a number of adaptive hypermedia systems (AHS) were independently developed, to illustrate different research ideas. Then we saw a move towards Web-based AHS, and the definition of the Dexter-based AHAM reference model [6]. Recently we are beginning to see the move from *special-purpose* AHS development towards the use of *general-purpose* AHS. This paper describes, through an example, how special-purpose adaptive applications and systems can be replaced, yet also preserved, by translating them into a general-purpose AHS. Examples of special-purpose AHS are InterBook [2], KBS-HyperBook [9,10], and INSPIRE [7]. They each implement a particular approach to hyperspace structuring and adaptation. Such an approach (or *model*) comes with specific kinds of *concepts* and *relationships*, a specific *hyperspace structure*, a specific *user modeling* mechanism that defines how user navigation is translated into the change of user characteristics (i.e., knowledge, interests), and some *adaptation logics* that defines how pages and links are adapted to user characteristics. In contrast, general-purpose AHS such as AHA! [5] are flexible enough to support a whole range of AH models. In this paper we describe a *high level* translation of applications (courses) created for the InterBook system into the *concept* and *adaptation* structures of the AHA! system. This translation process is exemplary for a type of development that is taking place in different research and development areas. We are aware of translations from MOT to AHA! (at TU/e), MOT to WHURLE (with the University of Nottingham), SCORM to AHA! (at the University of Cordoba). etc. We see a growing separation between *authoring* environments and *delivery* platforms, and advocate a *high level* translation process, like the one described in this paper.

In general the translation process from a specific AH model to a generic AHS consists of several steps:

1. The *concept types* and *concept relationship types* of the source system have to be mapped onto corresponding types in the target system, or such types have to be created in the target system (whichever is appropriate or possible).
2. A *syntactic* translation must be performed from the conceptual application structure in the source system to that of the target system. It is clear that such translation would benefit from the existence of a common or standard adaptation language, but alas no such language exists to date.
3. The adaptation behavior (or *semantics*) of the source application has to be implemented on the target system. With AHA! as a target system a translation from the high-level concept relationships to low-level adaptation rules must be defined. The expressive power of the target system is a crucial factor in determining whether a translation is possible at all. We describe a translation to AHA! because AHA! has a compiler from high level concept relationships to AHA!'s powerful low level adaptation rule language. For more details on that low level language and on how AHA! performs user model updates and adaptation we refer to [5].

HT'05, September 6-9, 2005, Salzburg, Austria.
ACM 1-59593-168-6/05/0009.

4. The *presentation style* of the source system must be translated to the target system, if this is possible. The presentation may look very different from the source system depending on presentation and layout possibilities of the target system. We describe the layout and presentation capabilities of AHA! to show the kinds of features that are needed to realize different presentation styles. Some details can be found in [3], and much more information on AHA! can be found on AHA!'s website http://aha.win.tue.nl/.

2. AHS to AHS TRANSLATIONS

In this section we describe the difference between low- and high-level translations and advocate the latter.

2.1 Low Level Translations

A translation from InterBook to AHA! was first described in [3]. The concept structures and the adaptive behavior of InterBook was translated into the database of concepts and event-condition-action rules that the AHA! engine uses. Such a low level translation is like a compiler from a high-level programming language to machine code. The resulting behavior is exactly what you want, but you cannot reconstruct the high-level program from the machine code. This early translation resulted in an AHA! application that worked exactly like the original, but of which the concept and adaptation rule representation was unreadable.

2.2 Concept and relationship level translation

Our new translation makes use of the *concept* and *concept-relationship* representations used by AHA!'s Graph Author tool [5]. Many AHS rely on the use of concepts and concept relationships, such as *prerequisite relationships* (or *background concepts* in InterBook terminology) and *knowledge updates* (or *outcome concepts* in InterBook). In AHA! arbitrary *concept relationship types* can be defined, and bound to a generic translation into *adaptation rules*. The latter translation is completely local to AHA! and needs to be defined only once for the source system (InterBook in our case). It can then be used to translate all applications that exist on that source system (InterBook). For InterBook, a translation from relationship to adaptation rules was defined for the *outcome* relationship type, because the way in which "knowledge" is "generated" by reading pages and the way pages are connected to concepts are all specific to InterBook. For instance, InterBook defines three whole knowledge levels, and reading non-recommended pages generates 1/3 of one whole knowledge level. Such behavior can (of course) be defined in the AHA! adaptation rule language but is not the default knowledge update mechanism of the standard AHA! distribution. When translating from other AHS to AHA! similar new concept relationship types and translations to adaptation rules are needed.

The creation of the translation from source *concept structure* to target concept structure requires no knowledge of the internals of the target system or its adaptation rules. As long as unary or *binary relationships* are used (like *prerequisites* and *outcome* relationships) a one-to-one translation is possible.

A translation between AHS at the conceptual level is an important step towards separating the development of *adaptive hypermedia applications* from the development of *adaptive hypermedia systems*. Whereas the InterBook to AHA! translation shows that an InterBook application can be transformed into an AHA! application with essentially the same conceptual structure, a similar translation can also be used to provide an adaptation platform for applications developed with new authoring tools for which no corresponding delivery platform exists. An example of such an authoring environment is MOT [4].

2.3 Translating Look and Feel

An AHS is more than its concept structure (and its user model update and adaptation rules). Even when the adaptation behavior of a translated system is emulated perfectly users will still have a very different experience if in the target system the layout of windows and frames is different, the background and colors on the pages are different, the link colors and annotations are different, etc. All these aspects together form the *look and feel* of the AHS.

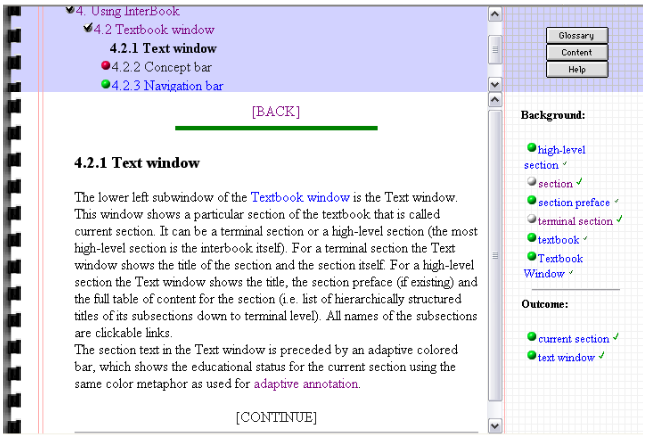
In order to be able to "translate" the look and feel we have considered the following aspect in our InterBook to AHA! translation:

- InterBook has different presentation formats. *Pages* are presented using a different layout from *glossary* concepts, and InterBook also uses separate windows to present them. In AHA! different layouts and windows can be defined for different concept types, so this aspect of the presentation is completely preserved in our translation.
- Each window is divided into frames. One frame (typically) displays a page whereas the other frames display some generated information, like a partial table of contents (or navigation menu), and in InterBook also a list of *background concepts* and a list of *outcome concepts*. In AHA! the information displayed in a frame is called a *view*. The (Java) code to generate different types of views, including a page view and a number of auxiliary automatically generated views, is part of the AHA! distribution but resides in a special directory. When a new type of view is needed new Java code needs to be developed and that code is dynamically loaded by the (running) server. The binding between views and the code to generate them is done through the name of the view (type), specified in the *layout file* for the application. For InterBook all necessary views were created for the translation of [3] already.
- Inside a frame the displayed content may be adapted by conditionally including objects or fragments. AHA! supports the conditional inclusion of external xhtml fragments, that can consist of just text but can also include images, applets or other artifacts, and can even include other conditionally included objects. AHA! also supports the conditional inclusion of a fragment of text that appears in the content page. InterBook does not perform content adaptation. During this project InterBook was extended with content adaptation, in order to make the authoring process suitable for creating applications that use more of the AHA! functionality.
- Inside a frame the links can be generated or adapted. Link adaptation can consist of *link hiding*, *link removal*, or *link annotation*. (See [1] for an overview of link adaptation techniques.) Link removal is in fact a form of content adaptation, so we do not discuss it further. In AHA! it is possible to adaptively change the colors of links, to choose between link underlining or not, and to place annotations (icons) in front of or after the link anchors. When the color *black* is used links can be hidden. When using other colors

the links are *annotated*. Annotations (icons) can be based on the value of user model attributes. They can be based on the "knowledge" of a concept but also on the "suitability" (which is for instance a condition on knowledge of background concepts). In AHA! it is possible to adapt links differently in every (simultaneously displayed) view. The InterBook to AHA! translation makes use of all these link adaptation features in order to recreate InterBook's link annotation through colored balls (green, white, red) based on the *suitability* of links, and through checkmarks (of different size) based on the *knowledge level* of concepts.

3. InterBook to AHA!: before and after

Because of lack of space we can only show the translation result of page concepts (in the Textbook window). However, other concept types are also correctly translated and presented. Figure 1 shows an InterBook page concept as presented by AHA!.



✓4. Using InterBook

✓4.2 Textbook window

4.2.1 Text window

●4.2.2 Concept bar

●4.2.3 Navigation bar

[BACK]

4.2.1 Text window

The lower left subwindow of the Textbook window is the Text window. This window shows a particular section of the textbook that is called current section. It can be a terminal section or a high-level section (the most high-level section is the interbook itself). For a terminal section the Text window shows the title of the section and the section itself. For a high-level section the Text window shows the title, the section preface (if existing) and the full table of content for the section (i.e. list of hierarchically structured titles of its subsections down to terminal level). All names of the subsections are clickable links.
The section text in the Text window is preceded by an adaptive colored bar, which shows the educational status for the current section using the same color metaphor as used for adaptive annotation.

[CONTINUE]

Background:

●high-level section ✓
○section ✓
●section preface ✓
○terminal section ✓
●textbook ✓
●Textbook Window ✓

Outcome:

●current section ✓
●text window ✓

Figure 1. InterBook Textbook Window translated to AHA!

Readers who are intimately familiar with InterBook [2] will notice that the translation comes close to the original, but that there are a few differences:

- Link anchors in InterBook are always blue. In AHA! it is possible to use three distinct link colors to annotate link anchors. We have chosen to use the typical AHA! scheme of *blue* and *purple* for recommended unvisited and visited links, and *black* for non-recommended links. This provides the user with more information than InterBook would give.
- The "BACK" and "CONTINUE" buttons are textual buttons (instead of graphical), because this too enables the use of colors to show the status of these links, thus providing the user with more adaptive information.
- Our current translation does not automatically generate links to glossary concepts. In InterBook the page of Figure 1 would contain more links, some of which leading to a glossary item.
- InterBook has the option to actively *guide* the user to the pages that are needed to learn the current concept, through a "teach me" button. This is missing in our translation because rather than creating a new custom view for the display of this list of pages, we have used views that are already provided by the standard AHA! distribution, and it does not contain such a "teach me" view..

4. Conclusions and Future Work

Many adaptive (Web-based) hypermedia applications that were developed over the years can be given new life by translating them to a general-purpose AHS. We have shown that a *high level* translation is possible in the case of translation to AHA!, providing a resulting application with an understandable structure of *concepts* and *concept relationships* and that even in the case of InterBook, with its complex layout and presentation format an appearance that comes close to the original can be achieved.

We will be working towards the definition of a general-purpose adaptation language that should enable us to perform many more translations from different authoring tools to different delivery platforms.

Acknowledgement

We thank the NLnet foundation for the financial support for realizing the AHA! system, the PROLEARN Network of Excellence for providing support for networking and dissemination of adaptive educational systems, and the University of Pittsburgh for their hospitality.

References

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

[2] Brusilovsky, P., Eklund, J., and Schwarz, E. Web-based education for all: A tool for developing adaptive courseware. Proceedings of Seventh International World Wide Web Conference, pp. 291-300, 1998.

[3] Brusilovsky, P., Santic, T., and De Bra, P. A Flexible Layout Model for a Web-Based Adaptive Hypermedia Architecture. Proceedings of the AH2003 Workshop, TU/e CSN 03/04, pp. 77-86, Budapest, Hungary, 2003.

[4] Cristea, A.I. and De Mooij, A. Adaptive Course Authoring: MOT, My Online Teacher. In Proc. of ICT-2003, IEEE LTTF "Telecommunications + Education" Workshop, 2003.

[5] De Bra, P., Aerts, A., Berden, B., de Lange, B., Rousseau, B., Santic, T., Smits, D., Stash, N., AHA! The Adaptive Hypermedia Architecture, Proceedings of the 14th ACM Hypertext Conference, pp, 81-84, Nottingham, 2003.

[6] De Bra, P., Houben, G.J., Wu, H., AHAM: a Dexter-based reference model for adaptive hypermedia, Proc. of the 10th ACM Hypertext Conference, pp. 147-156, Darmstadt, 1999.

[7] Grigoriadou, M., Papanikolaou, K., Kornilakis, H., Magoulas, G., INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment., Third Workshop on Adaptive Hypertext and Hypermedia, pp. 31-40, Sonthofen, July 2001.

[8] Halasz, F., Schwartz, M., The Dexter Hypertext Reference Model. Comm. of the ACM, Vol. 37, nr. 2, pp. 30-39, 1994.

[9] Henze, N., Nejdl, W., Adaptivity in the KBS Hyperbook System. Second Workshop on Adaptive Systems and User Modeling on the WWW, Toronto, and Banff, 1999.

[10] Nejdl, W., Wolpers, M., KBS Hyperbook - A Data-Driven Information System on the Web. WWW8 Conference, Toronto, May 1999.