The Next Big Thing: Adaptive Web-Based Systems

Paul De Bra, Lora Aroyo and Vadim Chepegin
Department of Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands
Email: {debra, laroyo, vchepegi}@win.tue.nl

Abstract

At the ACM Hypertext Conference a panel discussed "The Next Big Thing Inc." in the area of hypertext. The Web has been the "Big Thing" during the past 10 years, but its success has also made two predictions for the future (within three to five years from publication):

- the emergence of a mass market for hypertext;
- the integration of hypertext and other computer facilities.

It has taken a bit longer, but other than that the predictions were spot on. To some extent World Wide Web technology has been used to come true. The Web itself is the best illustration of the mass market come true. With over 3 billion pages at the time of writing this paper, indexed by the largest search engines according to Sullivan (2003), the Web has indeed reached the masses and is widely used all over the world. The prediction we look at here is the emergence of a mass market for hypertext.

Hypertext, and the Web in particular, offers three ways to find information:

- When you know the precise location of the information you want, you type a URL in the browser's location field, and you have it available online as well and sometimes even only online. Traditional help systems, like Microsoft Windows help, also have hypertext functionality has been effectively integrated with other computer facilities.

- When you have a good description of what it is you wish to find, the modern search engines are very good at finding the location of information you found before.

- When you want to explore the Web hoping to find interesting information, for instance hoping to find the sorts of things "Paul De Bra" has mentioned on the Web, the real problem starts. You can browse and search all you want, but you will most likely not find what you are interested in at that time. When you start browsing the search results for "Paul De Bra", the above prediction comes true. The Web has indeed reached the masses and is widely used all over the world.

The core problem in finding the information you want, in all the above cases, is describing what you want. Results from search engine searches are too short and unspecific to yield good results. Once a Web site with interesting information is found, it is often interesting pages only, because the site can only be navigated using its predefined link structure, independently of the search request.
The community of user modelling and adaptive hypermedia offers solutions for this problem: using information gathered about the user change the information content and link structure on-the-fly. User modeling captures the mental state of the user, and thus allows to: the explicit queries (or links) in order to determine precisely what the user is looking for. To support the design of this user model-building adaptive hypermedia systems like AHAM (De Bra et al., 1999, Wu 2002) and Munich (Mohr and Wirsing 2002), both based on the Dexter Model by Halasz and Schwartz, an attempt to standardize and unify the design of adaptive hypermedia applications, used mostly in isolated information spaces such as shopping site, an online museum, etc.

To become "The Next Big Thing", adaptive hypermedia systems need to open their architecture to allow collaboration between sites that are currently possible. The latest developments within the scope of the Semantic Web are leading adaptive hypermedia towards the "Adaptive Web", organized information in concept structures, where collaborative Web services allow for the encapsulation of the diverse knowledge, architecture, and for a more dynamic and sharable framework for automated personalization or adaptation. In this way, current adaptation can be extended with powerful reasoning at the level of standardized concept schemes, as opposed to the traditional hand-crafting by user relations, thus allowing adaptation generation in open information spaces on a higher schema level.

This paper first briefly recalls and summarizes the overall architecture of (Web-based) adaptive hypermedia systems (AHSs) to show how architecture to enable different AHSs to work together at different levels: the conceptual structure, the user model, and the adaptation model. We also outline some legal problems that arise as a consequence of opening the user model to be shared among various applications, which is technically unacceptable in the improved adaptive hypermedia architecture we propose.

2 Adaptive hypermedia architecture

On the Web personalization is being realized in two ways:

* Some Web sites require users to register and provide information about their interests. The site takes the interests into account when the user logs in. The user profile is only updated when the user revisits it to make changes.

* Some Web sites require users to register only so they can be identified. Identification is also possible through cookies, in which the identification taking place. The site monitors the user's browsing behavior (and in particular the pages that are visited) in order to represent the user's interests and knowledge. The presentation is updated according to the user model and the model is updated when the user revisits the site.

The former category of systems is called adaptable, the latter adaptive. A lot of research has been performed in the field of adaptive hypermedia systems. On coming up with new ways of adaptation, to offer the user better, more suitable, more acceptable and more visible guidance, to adapt to the user's changing needs.

2. The system checks the suitability of the requested page for this user. For this it needs to look at concept relationships (for the user model) and at the page structure (for the page model). A typical type of concept relationship that plays a role in determining the suitability is the prerequisite rules to check whether the page is suitable are defined in the adaptation model. The prerequisites are used to determine the suitability.

3. The system performs updates to the user model through (other) adaptation rules, e.g. the knowledge value of the concept is increased, and is increased more when the page is considered suitable than when the page is considered not suitable.

4. The presentation of the requested page can be adapted through adaptation rules in two ways:

   * The information content of the page can be altered, e.g. by adding or removing fragments. When a prereq explanation can be inserted, or when an interest value is high, additional details can be shown. Another possible adaption rule is to show additional media items based on preferences of the user.

   * The links that emanate from this page can be manipulated. Links to pages that are considered not suitable can be can be hidden, or even removed. The link destination can be changed as well.

5. Optionally the adaptation to content or links (e.g. the inclusion of a fragment) can cause user model updates as well.

The presentation of the requested page can be adapted through adaptation rules in two ways:

1. The user requests a page by clicking on a link in a Web page. Every page corresponds to a "concept" in the domain model (a overlay model).

2. The system checks the suitability of the requested page for this user. For this it needs to look at concept relationships from the user model. A typical type of concept relationship that plays a role in determining the suitability is the prerequisite rules to check whether the page is suitable are defined in the adaptation model. The prerequisites are used to determine the suitability.

3. The system performs updates to the user model through (other) adaptation rules, e.g. the knowledge value of the concept is increased, and is increased more when the page is considered suitable than when the page is considered not suitable.

4. The presentation of the requested page can be adapted through adaptation rules in two ways:

   * The information content of the page can be altered, e.g. by adding or removing fragments. When a prerequisite explanation can be inserted, or when an interest value is high, additional details can be shown. Another possible adaption rule is to show additional media items based on preferences of the user.

   * The links that emanate from this page can be manipulated. Links to pages that are considered not suitable can be can be hidden, or even removed. The link destination can be changed as well.

5. Optionally the adaptation to content or links (e.g. the inclusion of a fragment) can cause user model updates as well.

Many adaptive hypermedia systems have been developed, targeting various application domains, i.e. educational applications (ELM-AI Systems project, 2001). Interbook developed by Brusilovsky et al. (1998), KBS-Hyperbook by Henri and Nejdl (2001), NetCoach by Weber et al. (2002), and general-purpose adaptive hypermedia systems, like AHAM! by De Bra et al. (2002) and De Bra et al. (2003). A detailed taxo-
Figure 1. Overall architecture of the AHA! system

Figure 1 shows that the three submodels (DM, UM, AM) all reside on a central Web server, together with the "local pages" (content from external servers, but the corresponding concepts must exist in the local domain model, updates to the user model. That user model is used to perform the adaptation.

The centralized architecture of AHA! (and several other systems) has the advantage that the DM, AM and UM can work closely together retrieve the DM and AM when the server starts, and the UM when the user logs on, and to keep all that information cached in memory UM to disk (or database). For adaptive systems to work together, as suggested in the introduction, a new, decentralized and modular presented in Figure 2. The issue of optimization is transferred towards the bridging protocols and the information request coordinatio that architecture. We recognize that performance remains very important as a key to the success of adaptive information services in performant as non-adaptive services.

Apart from the issue of collaboration between adaptive systems, the centralized architecture has some other drawbacks: the semantic domain is usually stored inside a local domain and adaptation model. Relationships between concepts are "hidden" inside adaptation r graphical interface - i.e. Graph Author in AHA!, Figure 1 - for the authors to define the concepts and relationships and make high-leve difficult to "export" the semantic information contained in the DM/AM combination to applications that deal with concepts and relati difficult to "import" semantic information, e.g. an ontology, into an AHA! application. Also, the user model cannot be accessed by e applications running on the same AHA! server. Knowledge gained by the user through an online course cannot be used by an online department in order to initialize the user model there.

The next section looks at various research efforts attempting to provide solutions for these problems while facilitatng adaptation to and distributed open information space. Methods are proposed to improve the quality, consistency and linking of Web documents whil authors when creating the adaptation. Reflecting on this work we propose a new, modular architecture of an open Web-based adapti above difficulties.

3 Modular adaptive hypermedia

While ubiquitous computing was still a far fetched vision when introduced by Weiser at the Computer Science Lab at Xerox PARC in 1989, and a promising part of our everyday lives. After the age of mainframes (one machine - many people), and after the boom of personal person), now we are moving towards "the age of calm technology", as described by Weiser and Brown (1996) one person - many comp refers to it: the "Third Paradigm" of computing.

The essence of ubiquitous computing is to offer to users systems with "invisible" (ambient) intelligence, which will be able to know ab desired service, content and presentation, without intrusion and unnecessary human computer interaction. Adaptation across applicatio and realized with the notion of adaptive Web-based systems, defines the goal today: to provide adaptation within software environm interact simultaneously with various applications. For this we need open and modularized architectures, which are able to interact, e components. A fundamental issue in such architectures relates to the coordination, handling and control of all the components (servic biggest challenges in this context is the sharing, synchronization and interpretation of the user model among the different application within each system will be permanently evaluated and more detailed, and richer user models will be achieved in order to allow for personalization of the content. This paper presents our approach for achieving this openness and modularity of the adaptive hyperme different AHSs to work together at different levels (e.g. conceptual, user model, and adaptation) we see the need for four main asp:

- supporting a strict separation of domain model, application model, adaptation model and user model, to ensure good modulariz
- maintaining generic sharable (dynamic) models, such as the user model, to serve as a communication point for different AHSs;
● providing semantically rich descriptions of the components' functionality and their internal formats, to allow for interoperability
● providing mechanisms to describe the management, i.e. coordination and orchestration, of the communication between the system components

The basic idea is that by augmenting encapsulated system modules with rich formal descriptions of their competence, we can further aspects of the system management. The role of sharable models such as the dynamic user model is crucial for enabling inter-system inter-operability and reusability of user-related modules. Finally, by applying open Web standards we can enable interoperability and wide hypermedia systems.

Currently research in the area of the Semantic Web, originating primarily from the knowledge engineering and artificial intelligence field, promises new perspectives on Web services. For example, by defining how we can build task specifications (the problems which need to be solved), method specifications (the ways to solve problems), and domain models (ontologies and Web services), provides a number of standards and accompanying solutions which can be used to achieve the above. With the notion of ontology, which plays a role in facilitating the sharing of meaning and semantics of information between different system components, number of representational formats have been proposed as W3C standards for ontology and metadata representation. The most current existing Web standards (e.g. XML, RDF and RDF(S)) and add the primitives of description logic as powerful means for reasoning services. Initiatives to illustrate this is the SHOE metadata annotation of Web content (Heflin et al., 1999). The idea was further elaborated in the IRS-II approach (Carr et al., 2001), with use of ontology concept instances, considering evolving ontologies and offering annotation in a semi-automated way.

Several other annotation tools are known to the research community, e.g. the Amaya Web editor for RDF-based mark-up of resources points out the advantages of a centralized server, and its unapplicability in "open" environments. It exemplifies a good scenario for co-multiplexed "closed" content spaces, and illustrates its difficult implementation in an "open" information space. Other examples of annotation system project (Mulholland et al., 2000), the CREAM-based Ont-O-Mat/Annotizer (Handschuh et al., 2001), with use of ontology concept instances, considering evolving ontologies and offering annotation in a semi-automated way.

As the annotation is only part of the content authoring, another rather labor-intensive part is the process of linking the annotated content to other content. The Hypermedia Services Environment (COHSE) developed by Carr et al. (2001) introduces an ontological reasoning service over domain concepts, with use of ontology concept instances, considering evolving ontologies and offering annotation in a semi-automated way.

Thus, the next step in the process of opening up AHS architectures is applying a Web services perspective on the system components. The IRS-II approach supports capability-driven service invocation (e.g. find a service that can solve problem X) by task specifications (the problems which need to be solved), method specifications (the ways to solve problems), and domain models (ontologies and Web services). This separation of system components actually fits quite nicely with the requirements for exploiting the context of AHSs, as shown by Figure 2.
Figure 2. Architecture for adaptive Web-based systems (adapted from Motta et al. 2003).

Figure 2 illustrates our vision of the modular architecture for adaptive Web-based systems (Chepegin et al. 2003). One of the first characteristics is that the different system components are all equipped with facilities to communicate with the (other) components in terms of service bridges. The use of bridges is in accordance with the UPML framework connector defined by Fensel et al. (1999), in order to specify mappings between the architecture. Ontologies also play an important role to define and unify the system's terminology and properties to describe service. Each service can be specified by means of a corresponding ontology, providing common ground for knowledge sharing, exploit the services. This leads to a highly modularized architecture which offers a high degree of flexibility.

In the case of adaptive systems access to the user model via a Web service is a good example of this flexibility. It means that designers can interact with or react to the user intelligently without knowing anything in advance about that user, but simply using the knowledge c interpreting it in the context of the current application. Crucial for achieving such a flexible architecture is the need for a standardization of information about users. As mentioned above, open standards (e.g. XML, RDF, OWL) allow for the specification of ontologies to stand to enable reuse and interoperability. Another key aspect is to facilitate mobile user models that follow the user across applications. If agents provide implementation views supporting mobility and autonomous behavior. A final but important requirement is for the user to reuse system and knowledge components, and thus benefit from other applications.

A second characteristic seen in Figure 2 is the separation of the different components. In the traditional AHS approach, exemplified by distinguished (section 2). When transforming these components into services the need for a fourth component, the application model, lies in the fact that in the traditional AHS approach the adaptation model unites the actual process of how to adapt with the decision: applications, e.g. those realized in AHA!, the designer's knowledge about why the user is served in a certain way is more or less left in the implementation of the way in which the information is adapted to the user and does so based on the designer's decisions, which are not made to share and exchange the different functionalities between systems, it becomes relevant to separate the "how" and "why" in the designer's intentions about the roles, goals and tasks in the application (related to domain model concepts and user model values) itself to the actual realization of the adaptation to follow the directions given by the application model service. In fact this aligns well with the Role-G approach, mentioned above. The application model service contains a generic description of the user tasks in the context of a Role-G. It is clear that this gives the application model service a crucial role in the system architecture. It divides the adaptation process into technical adaptation is performed by the adaptation model service, while the management of the service process is coordinated from the entire architecture as displayed in Figure 2 emphasizes the fact that the core knowledge about the application processes and the goals) lies in the application model service. In the interaction with the application the user is represented by a particular role.
administrator, student). This role defines for him/her a corresponding behavior in terms of goals to achieve. To accomplish the user's applications are used, which realize one or several corresponding methods. The adaptation model service receives the direct user in application model service in order to define the context for the user Input for its most precise adaptation. Further the adaptation model service in order to select the relevant content to be presented to the user. The domain model service is responsible for the ex domain knowledge in terms of concepts of a domain ontology. Finally, it updates the user model with new values. For instance, when application, every action they perform on the interface is communicated to the user model service, which is responsible for updating values. The user information is stored there and a reasoning engine infers new knowledge from it and makes predictions concerning the user model service allows for sharability of the user model between applications by following the user (inside and outside the system) analyze data about the user's activities.

The following section discusses some possible disadvantages and legal problems raised by this open modularized architecture with respect to access and by various collaborating parties.

4 Legal issues in adaptive hypermedia

We do not claim to have any meaningful expertise in legal matters. However, before people start envisioning a great future for user n companies to perform personalization and adaptation for new and existing customers based on a "big brother" type of server that sell: of everything a user has done online, it is only fair to summarize a few existing and emerging efforts to make offering such services ill so. Kobsa (2002) points out that whereas privacy laws vary between countries, there are common principles, and some countries dec: e everyone who performs actions that have an effect in these countries. As a result it is wise to abide to the strictest laws when creating noteworthy issues are:

- Usage logs must be deleted after each session. This implies that adaptive systems may store the effect of user actions in a user "raw" actions. AHA! version 1.0 for instance kept a permanent log of user IDs and timestamps of every access to a page and of the AHA! version 2.0 the logging functionality has been removed, and in a future version we will introduce a log that is kept for the be legal to keep a complete log when the user agrees, but only if the system would offer the same functionality when the user c not keep a log.
- Usage logs of different services may not be combined. It is still unclear what the impact of such laws is on the proposed modular model becomes a service that can be used by different and distributed applications. We would of course like to create user mod the user, gathered from different sites, but this may or may not be allowed depending on the interpretation of the word "services" course program is one service, thus justifying the sharing of user model information between applications or courses within that departments of an electronic shopping mall that interact with customers independently could be considered one service, and the user. However, it remains to be seen whether this interpretation will be followed by the laws.
- Users should have a real, voluntary choice between an "anonymous" login (or login using a pseudonym) and a normal login using I version 2.0 anonymous logs are possible, yielding the same adaptation as non-anonymous logs. It helps to provide users with advance notice about the data that is to be collected, and to indicate the purposes for which this is being done.
- User modeling should be "transparent", or "scrutable": users should be able to inspect their user model and understand what it is to change or erase data, especially the assumptions the system inferred about them.
- The user models should be protected using adequate security measures. When user modeling becomes a service, accessible by n non-trivial to shield user models from intruders and eavesdroppers.

This short list is far from complete, but it already shows that for adaptive Web-based systems to become a "big thing" there are a nuer clarified and then implemented into the systems.

5 Conclusions and future work

Adaptive Web-based systems are ready to make the jump from single applications to modular distributed frameworks in which multiple models and adaptation rules. The challenge for the future is to get research groups to work together to develop standards for exchange and adaptation model level, so that different systems can indeed start to share user modeling and adaptation information. While new also need clarity in the legal issues involved in sharing user modeling information. We are opposed to a "big brother" style of user mod with voluntary user consent some sharing will be allowed under controlled circumstances. With many applications of adaptive Web-ba collaborating, distributed, modular systems, the idea of adaptive systems can still become a "next big thing".

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments, and also Geert-J discussions leading to an improved description of the modular article.

References


based Systems’. In Conferentie Informatiewetenschap, Eindhoven, November, pp. 29-36
http://wwwis.win.tue.nl/infwet03/proceedings/3/

Adaptive Hypermedia Architecture’. In Proceedings of the fourteenth ACM conference on Hypertext and Hypermedia, College Park, MD, June, pp. 21-22
http://wwwis.win.tue.nl/~debra/ht02.pdf


Kobsa, A. (2002) "Personalized Hypermedia and International Privacy". Communications of the ACM, 45 (2)


Links

Amaya Web Editor http://www.w3.org/Amaya/

Annotea http://www.w3.org/2001/Annotea/


DAML Services http://www.daml.org/services/