The next big thing: adaptive web-based systems

De Bra, P.M.E.; Aroyo, L.M.; Chepegin, V.

Published in:
Journal of Digital Information

Published: 01/01/2004

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
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 raised legal issues of user modeling and sharing of user models that is allowed. The paper then proposes a new, modular architecture that should lead to adaptive Web-based systems as the "Next Big Thing" indeed. Other applications can collaborate in creating and updating a user model. Shared user models can of course cause a "big brother" problem. Legislation is already in place to protect legal limits on the kind of user modeling and sharing of user models that is allowed. The paper made two predictions for the future (within three to five years from publication):

1. the emergence of a mass market for hypertext;
2. 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 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 the Web has indeed reached the masses and is widely used all over the world. It may be possible that the next big thing is all about searching for the right information and reach the goal of offering each individual user (or user group) the information they need. Discussion there was debate about whether the user should always have access and control over (hypertextual) information space. There were different views on whether the "right" to all the information can be found. We argue in favor of adaptation but at the same time the way adaptive hypermedia has been used until now. The paper then proposes a new, modular architecture that should lead to adaptive Web-based systems as the "Next Big Thing" indeed. Other applications can collaborate in creating and updating a user model. Shared user models can of course cause a "big brother" problem. Legislation is already in place to protect legal limits on the kind of user modeling and sharing of user models that is allowed. The paper made two predictions for the future (within three to five years from publication):

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

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

1. 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. Hypertext functionality has been effectively integrated with other computer facilities.
2. 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. Hypertext functionality has been effectively integrated with other computer facilities.
3. 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 (the first draft of) this article the first author was watching a program on hackers on National Geographic and remembered that security problem in the Solaris "tar" program. Typing the search term "Paul De Bra", "tar", "vulnerability" and "Solaris" into Google gave the CERT Advisory describing the problem that he reported 10 years ago.
4. 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 want. Results from search engines 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.
Many adaptive hypermedia systems have been developed, targeting various application domains, i.e. educational applications (ELM-AI and Wu 2002) and Munich (Hoehn and Wirsing 2002), both based on the Dexter Model by Halasz and Schwartz (1999). 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 as soon as possible. The latest developments within the scope of the Semantic Web are leading adaptive hypermedia towards the “Adaptive Web”, an organization in consent with the concept of the Semantic Web, where contextual Web services allow for the encapsulation of the diverse, knowledge-based, and for a more dynamic and shareable framework for automated personalization or adaptation. In this way, current adaptations can be extended with powerful reasoning at the level of standardized concept schemes, as opposed to the traditional hand-crafting by ad-hoc 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 enables different AHSs to work together at different levels: the conceptual structure, the user models, and the adaptation to content or links (e.g., the inclusion of a fragment) can cause user model updates as well.

### 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 visits it and makes changes.
- Some Web sites require users to register only so they can be identified. Identification is 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 with the information provided by the user.

The former category of systems is called adaptable, the latter adaptive. A lot of research has been performed in the field of adaptive focused on coming up with new ways of adaptation, to offer the user better, more suitable, more acceptable and more visible guidance. Another area has focused on finding higher-level ways of describing aspects of the application domain and the basis for performing adaptation. In the AHAM reference model, by De Bra et al. (1999) and Wu (2002), the core of the architecture is formed by three closely linked components: the domain model (DM), the user model (UM) and the adaptation model (AM). In AHAM and other architectures, a concept in the domain model is typically an overlay model of the domain model. How the user model relates to the concept model is then a corresponding “concept” in the user model that represents how the user relates to that concept. For instance, the user model will keep track of the user’s knowledge of each of the concepts in the domain model. The interplay between by explaining how an AHS processes a user’s request, using its adaptation engine:

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 concept’s rules to check whether the page is suitable are defined in the adaptation model.
3. The system performs updates to the user model through (other) adaptation rules, e.g., the knowledge value of the concept page is increased, and is increased more when the page is considered suitable than when the page is considered not suitable. The knowledge value of the concept page is increased when the page is considered suitable than when the page is considered not suitable. The propagation of pages from sections to chapters.

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 conditionally including or hiding fragments. When a prerequisite explanation can be inserted, or when an interest value is high, additional details can be shown. Another possible adaptation media items based on preferences of the user.

- The links that emanate from the page can be manipulated. Links to pages that are considered not suitable 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 and Brusilovsky 2001), Interbook developed by Brusilovsky et al. (1998), KB-Hyperbook by Herze and Nejdl (2001), NetCoach by Weber et al., systems, or general-purpose adaptive hypermedia systems, like AHA! by De Bra et al. (2002), subclass of the Dexter Model by Halasz and Schwartz (1999). A detailed taxonomy of the systems used in AHS is given by Brusilovsky (2001). The content and link adaptation possibilities given above are just a few from a much larger AHSs have been developed with one application or application area in mind. A notable exception is the AHA! system developed as a more centralized architecture shown in Figure 1.
Figure 1 shows that the three submodels (DM, UM, AM) all reside on a central Web server, together with the "local pages" (content from Java Applets handles page requests. Pages can come from external servers, but the corresponding concepts must exist in the local domain model, which is updated 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 to 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 (or a database). For adaptive systems to work together, as suggested in the introduction, a new, decentralized and modular architecture is necessary. Figure 2 presents the idea of optimization is transferred towards the bridging protocols and the information request coordination 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 model. Graphical interface - i.e. Graph Author in AHA!, Figure 1 - for the authors to define the concepts and relationships and make high-level difficult to "export" the semantic information contained in the DM/AM combination to applications that deal with concepts and relations difficult to "import" semantic information, e.g. an ontology, into an AHA! application. Also, the user model cannot be accessed by external applications running on the same AHA! server. Knowledge gained by the user through an online course cannot be used by another 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 facilitating adaptation to and distributed open information space. Methods are proposed to improve the quality, consistency and linking of Web documents while authors create the adaptation. Reflecting on this work we propose a new, modular architecture of an open Web-based adaptive hypermedia system to support a strict separation of domain model, application model, adaptation model and user model, to ensure good modularity and maintainability of the models. This paper presents our approach for achieving this openness and modularity of the adaptive hypermedia system. To achieve different AHSs to work together at different levels (e.g. conceptual, user model, and adaptation) we see the need for four main aspects:

- supporting a strict separation of domain model, application model, adaptation model and user model, to ensure good modularity and maintainability of the models,
- maintaining generic sharable (dynamic) models, such as the user model, to serve as a communication point for different AHSs;

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 computing, now we are moving towards the "age of calm technology", as described by Weiser and Brown (1996) (one person - many computers), and after the boom of personal computing, now we are moving towards the "age of calm technology", as described by Weiser and Brown (1996) (one person - many computers), now we are moving towards "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 about desired service, content and presentation, without intrusion and unnecessary human computer interaction. Adaptation across applications and devices is realized within the notion of adaptive Web-based systems, defines the goal today: to provide adaptation within software environments 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 (services) involved in a service. The biggest challenges in this context is the sharing, synchronization and interpretation of the user model among the different applications within each system will be permanently evaluated and more detailed, and richer user models will be achieved in order to allow for a personalization of the content. This paper presents our approach for achieving this openness and modularity of the adaptive hypermedia system. To work together at different levels (e.g. conceptual, user model, and adaptation) we see the need for four main aspects:
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's 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 sharing 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, ontologies and Web services, provides a number of standards and accompanying solutions which can be used to achieve the above. We have the notion of ontology, which plays a role in facilitating the sharing of meaning and semantics of information between different systems. Number of representational formats have been proposed as W3C standards for ontology and metadata representation. The current existing Web standards (e.g. XML, RDF and RDF) 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 (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 "closed" content spaces, and illustrates its difficult implementation in an "open" information space. Other examples of annotation system project (Muhtolland et al., 2000), the CREAM-based Ont-O-Mat/Annotizer (Handschiuh et al., 2001), etc.

As the annotation is only part of the content authoring, another rather labor-intensive part is the process of linking the annotated content. Hypermedia Services Environment (CHSE) developed by Carr et al. (2001) introduces an ontological reasoning service over domain specific combination with a Web-based open hypermedia link service: this enables documents to be linked via metadata describing their context. Another recent project inspired by CHSE and applying ontologies and semantic services for the automation of semantic content (2003). This project facilitates various interpretation views on the same content with no prior mark-up, but by means of adding an (see Domingue et al., 2003). On top of this semantic layer Magpie deploys semantic services provided to the user as a physically independent resource.

Thus, the next step in the process of opening up AHS architectures is applying a Web services perspective on the system components. The above-mentioned semantics and offer means for flexible composition of services (system components) through automatic selection, verification of service properties, and execution monitoring. In an approach such as DAML-S/OWL-S (DAML), for example, the ProcessC definition in terms of its state, initial activation, execution, and completion. The ServiceModel, on the other hand, provides a means to control the flow in the case of a composite service, and the ServiceGrounding specifies the service access to information by communicating mechanisms, etc.

Another relevant approach for describing the role of Web services in system architecture is the Web Service Modeling Framework (WS-MF). That research shows that Web services appear to be a useful solution for achieving modularization. We can achieve reasonable autonomy of the main aspects of Web services (e.g. Web service location, composition and mediation) by extending them with rich formal description of standardized languages such as RDF or OWL). In this way we can allow adaptive Web-based systems to reason about the functionality of services, to locate the best ones for solving a particular problem, and automatically to compose the relevant Web services for dynamically changing situations.

An interesting approach that could serve as the basis for a successful application of the Web service perspective on AHS architecture is the Internet Reasoning Service (IRS-II) introduced by Hector et al. (2003). They show how we can support the public administration, and execution of heterogeneous semantic-rich Web services. The service uses UPML (Unified Problem-solving Method description Language) knowledge-based systems by defining how we can build elementary components and how these components can be integrated into a Web service. The IRS-II approach supports capability-driven service invocation (e.g. find a service that can solve problem X) and task specifications (the problems which need to be solved), method specifications (the ways to solve problems), and domain models (the problems need to be solved). 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 illustrates our vision of the modular architecture for adaptive Web-based systems (Chepegin et al. 2003). One of the first characteristics that the different system components are all equipped with facilities to communicate with the (other) components in terms of service bridges are used in accordance with the UML framework connector defined by Fensel et al. (1999), in order to specify mappings between the components within 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 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 standardized 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. The 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 the application model service contains a generic description of the user tasks in the context of a Role-Goal-Task model. 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 (e.g. a...
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 user interface is communicated to the user model service, which is responsible for updating the values. The user information is stored there and a reasoning engine infers new knowledge from it and makes predictions concerning it. The user model service allows for shareability of the user model between applications by following the user (inside and outside the system) activities. 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 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 companies to perform personalization and adaptation for new and existing customers based on a "big brother" type of server that sells everything a user has done online, it is only fair to summarize a few existing and emerging efforts to make offering such services illegal.

Kobsa (2002) points out that whereas privacy laws vary between countries, there are common principles, and some countries do: 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. AHAI version 1.0 for instance kept a permanent log of user IDs and timestamps of every access to a page and in the AHAI version 2.0 the logging functionality has been removed, and in a future version we will introduce a log that is kept for the life of the user so that the system would offer the same functionality when the user cannot 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 a user model at the gathering from different sites, but this may or may not be allowed depending on the interpretation of the law "services course program is one service, thus justifying the sharing of user model information between applications or courses within that department 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 law.
- Users should have a real, voluntary choice between an "anonymous" login (or login using a pseudonym) and a normal login using version 2.0 anonymous logins are possible, yielding the same adaptation as non-anonymous logins. It helps to provide users with advance notice about the data that is to be collected, and to indicate the purposes for which this data is being done.
- User modeling should be "transparent", or "scrutable": users should be able to inspect their user model and understand what it contains 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 number of clarifications 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 exchanging 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 modeling with voluntary user consent some sharing will be allowed under controlled circumstances. With many applications of adaptive Web-based collaborative, 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-Jan de Bra for 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/infwit03/proceedings/3/

http://wwwis.win.tue.nl/~debra/ht02.pdf

http://citeseer.ist.psu.edu/debra99aham.html


http://informatikuibk.ac.at/~c70385/wese/wsmf.paper.pdf


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


Links

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

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


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