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 always been mixed. We argue in favor of adaptation but at the same time make two predictions for the future (within three to five years from publication):

1. to reach the goal of offering each individual user (or user group) the information they need.
2. to enable the development of ambient information systems.

The paper then proposes a new, modular architecture that should lead to adaptive Web-based systems as the "Next Big Thing" indeed. Different 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 legal issues of user modeling and adaptation in order to provide not just a future outlook but based on a realistic vision of what will not only become technically possible but also of what can be done.

1 Introduction

Nielsen (1990)

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 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. The prediction we look at here is 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. Different 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 legal issues of user modeling and adaptation in order to provide not just a future outlook but based on a realistic vision of what will not only become technically possible but also of what can be done.

1 Introduction

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.
- 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 terms "Paul De Bra", "tar", "vulnerability" and "solaris" into Google then proposed a new, modular architecture that should lead to adaptive Web-based systems as the "Next Big Thing" indeed. Different 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 legal issues of user modeling and adaptation in order to provide not just a future outlook but based on a realistic vision of what will not only become technically possible but also of what can be done.

1 Introduction

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.
- 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 terms "Paul De Bra", "tar", "vulnerability" and "solaris" into Google then proposed a new, modular architecture that should lead to adaptive Web-based systems as the "Next Big Thing" indeed. Different 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 legal issues of user modeling and adaptation in order to provide not just a future outlook but based on a realistic vision of what will not only become technically possible but also of what can be done.
The community of user modeling 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 apply the explicit queries (or links) in order to determine precisely what the user is looking for. To support the design of this user model-based system, like AHAM (De Bra et al. 1999, Wu 2002) and Munich (Koch and Wirsing 2002), both based on the Dexter Model by Halasz and Schwartz (1990), an attempt is made to standardize and unify the design of adaptive hypermedia applications, used mostly in isolated information spaces such as shopping sites, an online museum, etc.

To become "The Next Big Thing", adaptive hypermedia systems need to open their architecture to allow collaboration between sites currently possible. The latest developments within the scope of the Semantic Web are leading adaptive hypermedia towards the "AdaM" 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 be extended with powerful reasoning at the level of standardized concept schemes, as opposed to the traditional hand-crafting by author 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 an architecture to enable different AHSs to work together at different levels: the conceptual structure, the user model, and the adaptation framework. Out 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 and builds a user profile 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 case 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.

The former category of systems is called adaptable, the latter adaptive. A lot of research has been performed in the field of adaptive hypermedia focused on coming up with new ways of adaptation, to offer the user better, more suitable, more acceptable and more visible guidance. The AHAM reference model, by De Bra et al. (1999) and Wu (2002), is one of the architectures for performing adaptation. In the AHAM reference model, the domain model (DM), the user model (UM) and the adaptation model (AM). In AHAM and other adaptive hypermedia systems, like AHAM, typically an overlay model of the domain model "concept" in the domain model there is a corresponding "concept" in the user model that represents how the user relates to that concept. The interplay between the user model and the model is updated.

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.

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 fro 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.
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. Propagates in 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 is the inclusion of a fragment that can be hidden, or even removed. The link to the page is changed.
- 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. The page is considered not suitable, adapted, or removed.

Many adaptive hypermedia systems have been developed, targeting various application domains, i.e. educational applications (ELM-AD, Brusilovsky 2001), Web systems, or general-purpose adaptive hypermedia systems, like AHAM by De Bra et al. (2002) and KBS-Hyperbook by Henze and Nejdl (2001). Another possibility is that AHS used in AHS is given by Brusilovsky (2001). The content and link adaptation possibilities given above are just a few from a much larger space of possibilities.
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 handles page requests. Pages can come 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 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 UM to disk (or database). For adaptive systems to work together, as suggested in the introduction, a new, decentralized and modular architecture. We recognize that performance remains very important as a key to the success of adaptive information services in a 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-level difficult to "export" the semantic information contained in the DM/AM combination to applications that deal with concepts and relational difficult to "import" semantic information, e.g. an ontology, into an AHA! application. Also, the user model cannot be accessed by 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 facilitating adaptation to and distributed open information space. Methods are proposed to improve the quality, consistency and linking of Web documents while 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 1991 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 about desired service, content and presentation, without intrusion and unnecessary human computer interaction. Adaptation across applications and realized with 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 (servic 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 personalization of the content. This paper presents our approach for achieving this openness and modularity of the adaptive hypermedia 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 AHS;
● 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 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 a number of representational formats which have been proposed as W3C standards for ontology and metadata representation. The most common existing Web standards (e.g., XML, RDF and RDFS) 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 a project 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 projects include the CREAM-based Ont-O-Mat/Annotizer (Handschuh et al., 2001), Web (Vargas-Vera et al., 2002). Lett

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 resources. Hypermedia Services Environment (COHSE) developed by Carr et al. (2001) introduces an ontological reasoning service over domain concepts by RDF combination with a Web-based open hypermedia link service: this enables documents to be linked via metadata describing their contexts. Another recent project inspired by COHSE and applying ontologies and semantic services for the automation of system content is the CREAM-based Ont-O-Mat/Annotizer (Handschuh et al., 2003). This project facilitates various interpretation views on the same content with no prior mark-up, but by means of adding an annotation layer. On top of this semantic layer Magpie deploys semantic services provided to the user as a physically independent service, i.e., coordination and orchestration.

Thus, the next step in the process of opening up AHS architectures is applying a Web services perspective on the system components. 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+OWL-S (DAML; for example, the ProcessC model for DAML-S) or 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 communication mechanisms, etc.

Another relevant approach for describing the role of Web services in system architecture is the Web Service Modeling Framework (WSMF), which introduces a useful solution for achieving modularity. 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). This way we can allow different 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 dynamic 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 framework like the Internet Reasoning Service (IRS-II) introduced by Hector et al. (2003). They show how we can support the publically 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 framework like the Internet Reasoning Service (IRS-II) introduced by Hector et al. (2003). They show how we can support the publically execution of heterogeneous semantic-rich Web services.
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 each other. Service bridges are used in accordance with the UPML 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, exploiting 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 of 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 standarize to enable reuse and interoperability. Another key aspect is to facilitate mobile user models that follow the user across applications. Mobile agents provide implementation views supporting mobility and autonomous behavior. A final but important requirement is for the user to be able 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 separation model, lies in the fact that in the traditional AHS approach the adaptation model unites the actual process of how to adapt with the domain model, 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 implicit. The way in which the information is adapted to the user and does so based on the designer's decisions, which are not made in advance, is an important factor in the actual realization of the adaptation to follow the directions given by the application model. In fact this aligns well with the 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 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. user).
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 exogenous 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 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 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 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 sells 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. Kehsa (2002) points out that whereas privacy laws vary between countries, there are common principles, and some countries decriminalize 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 legal requirement to keep a complete log when the user agrees, but only if the system would offer the same functionality when the user did 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 model service, gathered from different sites, but this may or may not be allowed depending on the interpretation of the word “service” 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 law.
- Users should have a real, voluntary choice between an anonymous “login” (or login using a pseudonym) and a normal login using a 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 is being done.
- User modeling should be “transparent”, or “scrutable”: users should be able to inspect their user model and understand what it means 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 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 issues that need to be 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 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 systems, the idea of 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-John discussions leading to an improved description of the modular article.

References


Links

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

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


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