Fine-grained user models by means of asynchronous web technologies

David Hauger
Institute for Information Processing and Microprocessor Technology
Johannes Kepler University, Linz, Austria
hauger@fim.uni-linz.ac.at

Abstract

Although asynchronous HTTP technologies have grown in importance with the emergence of Web 2.0, most web-based Adaptive Hypermedia Systems (AHS) still exclusively use server-side monitoring of user behaviour to set up the user model. This paper discusses how asynchronous technologies and client-side observation may lead to more accurate and detailed user models, and how that might benefit next-generation AHS.

1 Introduction

Typical user-adaptive systems gather information about the user, create a user model based on this information and then apply the user model to make predictions about the user or make decisions [Jameson, 2006]. Currently, most web-based AHS acquire information primarily by tracking HTTP requests of resources or by retrieving it directly from the users [Barla, 2006], which is mainly due to technical reasons [Putzinger, 2008].

The current paper points out some disadvantages of this approach to data acquisition and user modeling and puts forth ideas on how user models in AHS may be improved by means of asynchronous technologies.

2 Drawbacks of traditional user modelling techniques

Traditionally, observation of interaction between user and web-based AHS has been based on HTTP requests for pages or other system resources. These requests are logged, processed and interpreted in the context of the system's application domain. For instance, in AHA! [De Bra and Calvi, 1998] the requested pages are mapped to concepts; having requested a page the user is assumed to be familiar with the corresponding concept. Additional information about the AHS domain can add further semantics to the request-based approach; such information can be either directly expressed in the system (e.g., the definition of prerequisite concepts in AHA! [De Bra and Ruiter, 2001]), or extracted dynamically at runtime (e.g., keyword-based analysis of page content in PAADS [Bailey, 2002]).

More recent systems also take into account the timestamps of requests in order to calculate the time spent on a page [Posea *et al.*, 2006]. This additional information helps to set up a more detailed user model and allows more specific assumptions on the user's behaviour (e.g., by differentiating between quickly browsing through pages and having had the time to read the contents).

However, no matter how elaborate these systems are, the user's (inter-)actions on the browser/client side still remain a "black box". Due to this lack of information some assumptions about the user can not be drawn while others are inevitably ambiguous. Examples that demonstrate the limitations of such techniques are:

- Determining whether a user has spent time on the page: Even if users request a page, they may close it immediately afterwards, go away, work or look at something else (outside the system). There is no information on what happens after the request.
- Determining whether a user has had a look at the whole page: Especially on large pages, parts may have never been visible to users, if they have never scrolled down to the bottom.
- Determining whether a user has read a page: Even if the amount of time spent on a page is sufficient to assume the users might have read its contents, they might not have gone through the page with a speed that would have allowed them to read the contents thoroughly.
- Determining whether a user is interested in the page: There is no server-side possibility to tell whether the user has further processed the page, copied or printed parts of the page, etc., which could be used to determine special interests.
- Determining whether a user has understood a page: Of course this can be tested separately, but if unobtrusive observation should be used to set up the user model, it is even harder to tell whether users might have problems with understanding the contents if there is no information on their client side actions.
- Determining why a user has skipped a page: There are several reasons why a page might have been skipped after a short glance: the user might be an expert who is already familiar with the subject, a novice regarding the content to be to difficult or a person that is not interested in the topic at all.

Summing up it may be concluded that although the information obtained from logging the user's requests on system pages and resources allows some inferences, the gathered data might not be sufficient for setting up an accurate user model. Therefore, it would be helpful to receive more fine-grained information on the user's interactions. The approach described in the current paper tries to focus on continuous monitoring of interactions instead of data consumption, which will provide new ways for setting up user models in AHS.

3 State of the art / Related work

Farzan and Brusilovsky used the "time spent reading" in the "Knowledge Sea" system to get more information on user interaction [Farzan and Brusilovsky, 2005]. However, this information has not been directly used for user modelling, but for social navigation support in order to determine the relevance of the page. Goecks and Shavlik [Goecks and Shavlik, 2000] have used JavaScript to log mouse and scrolling activities, summed up into a "level of activity", which – based on the keywords extracted from the visited pages – have been used to determine the "interests of the user" in an agent-based system. Hijikata [Hijikata, 2004] showed that text tracing, link pointing, link clicking and text selection based on keywords that have been extracted from a page is better suited to determine the user's interest in a set of keywords than relying solely on the contents of a whole page.

Despite evolutionary steps such as the above, most of the past research done on this topic in terms of adaptive hypermedia has had to face technical boundaries. Whereas desktop applications have been able to use all types of user input, hypermedia systems had to face the limitations of browsers and the strict HTTP request-response cycle, which made it almost impossible to send information on client-side interactions to the server (except for workarounds like refreshing iframes sending data via hidden form fields). As a result, web server log files (or equivalent) have been primarily used as a basis for analysing user behaviour [Hofmann *et al.*, 2006].

Several solutions have been developed to overcome these difficulties and to allow client-side user-monitoring. One was to use JavaScript and hidden form fields [Hofmann *et al.*, 2006] sending information on idle time together with the subsequent request, which still does not allow continuous observation. Other approaches used custom browsers (like "The Curious Browser" [Claypool *et al.*, 2001] and "AVANTI" [Fink *et al.*, 1996]) or browser plugins [Goecks and Shavlik, 2000], which while effective, limited the applicability of the approach.

Thanks to asynchronous web technologies like AJaX and Flash it is now possible to monitor the user's actions directly, continuously and unobtrusively within common web browsers. However, although it has been stated that these technologies might be used to retrieve more detailed information on user interaction [Putzinger, 2008], they have not yet been used to improve user model granularity and accuracy (also because in most cases the research on users' interactions had a different focus). The current paper proposes potential approaches in this direction.

4 Improving user models with client-side activity data

As explained in section 2, actions on the client side are not yet being widely monitored continuously to set up a user model. An interaction-based user model having access to this data could make several additional assumptions on the user, which an adaptive system would definitely benefit from.

Technically, the information can be acquired by means of asynchronous technologies like AJaX, Flash or Java Applets. They all allow monitoring of user actions and provide a way to directly, immediately and unobtrusively send this information to the server where it can be further processed and evaluated. During analysis, the system will typically seek to identify activity patterns from which further inferences can be made.

As an example, consider the case of determining if a page has been read. A system enhanced as proposed, could have access to the following indicators to make that determination:

- Focus: Usually a page being read should have the focus. Reading on an inactive, but still visible browser window while in parallel taking notes in a different window can be regarded as an exception.
- Time spent: Reading a page requires the user to remain on it sufficiently long to read its contents.
- Scrolling: If a longer page (larger then the displayed window) is being read, the user has to scroll through the page. The scrolling has to be slow enough to be able to read the text (dependent on the number of words as well as on how many lines have been scrolled down in what time). Scrolling back up a few lines may indicate the user has not understood parts of the text and is now reading the text more thoroughly.
- Mouse activity: A user sitting in front of the screen and reading a page will probably (but not necessarily) move the mouse unintentionally. Mouse movements can also be used to determine the locus of attention. Text tracing, i.e. marking parts of a text, is a strong indicator of the user's attention, no matter whether it is done unintentionally during reading or intentionally in order to copy the text.
- Keyboard activity: Not only mouse events, but also keyboard events like shortcuts for copying, printing or selecting text indicate that the user is actively working with the currently displayed text. Especially cursor and function keys have to be monitored.
- Partial or repeated reading: Users might not open, read or skip a page just once, but also return to a page having been accessed before. It is also possible that a user reads part of a page, follows a link for further information and then returns to read the rest of the page. Therefore it would be helpful not to regard the page as a whole, but to split pages into smaller pieces and/or define the parts of the page that have been read.

It is argued that inferences based on such larger bodies of evidence will inevitably be more accurate than it was possible before. Furthermore, considering patterns in user behaviour, it might also be possible to make assumptions on the users' knowledge, even if they are just glancing at a page and skip it before reading the contents. For example, an expert user might scroll through a page quickly, but still slow enough to have a glance at the content and then move on to a more specific topic while a novice or a person not interested in the content of the page might stop scrolling at the beginning, read a few lines and then go back to a more basic or a completely different page.

In addition to enabling more accurate inferences, having access to additional data allows more fine-grained representations of user state. For instance, a user's state in relation to a page can now specifically indicate which parts of a page a user has read or skimmed through, whether any parts of the page have been revisited at a later time, etc. Or, alternatively, several levels of reading a page may be differentiated, like glancing at a page, reading half of a page, reading everything thoroughly and using parts of the page for further interactions like printing, copying, etc.

5 Anticipated benefits

The two main benefits of a user model based on information about the user's client side actions are improved accuracy and increased level of detail. Better accuracy results in a more reliable user model and, consequently, in higher-quality adaptations. As granularity is improved, more detailed information is available that can be used for adaptation, e.g. by defining new/more detailed rules within existing adaptation engines. For instance, prerequisites may not only be defined by "having requested the page once", but by "having read at least 50% of the page", which improves the system without complicating the authoring process.

Additionally, the information stored in the user model might not only be used to apply rules, but also serve to generate new information (e.g. by determining interaction patterns for user groups in order to be able to distinguish between "not needing a piece of information because of prior knowledge" and "not being interested in a topic"). As standardized web technologies are going to be used, the current approach will address a larger community (reaching also people that might not be willing to install and/or use extra software), which offers an additional benefit: More data will be available, which will help to generate information not only about particular users, but on user groups as well, which may support collaboration and group learning.

Moreover, the proposed approach has the potential to help shift the focus of user modelling from a content-based orientation to an activity-based one, which will create new challenges, but also a large variety of additional possibilities for adaptivity. Furthermore, the model may continuously be modified and held up-to-date. Combined with the fact that asynchronous technologies offer the possibility to initiate actions and adaptations detached from the request-response cycle, just-in-time adaptations may take place, e.g. immediately offering possibilities to contact a currently available user possessing the required knowledge when determining a user might need help.

6 Outlook and future work

Ongoing work is focusing on the development of libraries for such detailed and action-based user-monitoring and -modelling and on setting up a prototype of an AHS including these libraries based on the proposed modifications. It will use AJaX to monitor user behaviour; especially interactions mentioned in section 4 indicating attention and interest. The gathered information and the deriving user model will be used within an existing AHS (in addition to its existing user model) that by this time will be integrated into the e-learning platform "Sakai" [Sakai, 2008] as adaptivity features need to be integrated into commonly used systems in order to be used more frequently and offer their benefits to a larger community [Hauger and Köck, 2007].

A first evaluation will show the contents/properties of both user models to the learners having used the system. They will be asked to evaluate quality and accuracy of both models and to compare their values by telling which ones are fitting better. Moreover, a second step of evaluation will compare the system's adaptations based on both models. This is because for some users it may be difficult to e.g. specify a "percentage level of interest" while it is easier to tell whether an adaptation has been helpful or not.

System and evaluation aim at validating the approaches proposed in this paper and at demonstrating the capabilities of asynchronous technologies, which will hopefully help to enhance and enrich future AHSs.

Acknowledgements

The work reported in this paper has been partially funded by the Socrates Minerva "Adaptive Learning Spaces" (ALS) project (229714-CP-1-2006-1-NL-MPP). For information on the ALS project please refer to the project's web site: http://www.als-project.org.

References

[Bailey, 2002] C. P. Bailey. *An Agent-Based Framework to Support Adaptive Hypermedia*. PhD thesis, University of Southampton, 2002.

[Barla, 2006] M. Barla. Interception of user's interests on the web. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 435–439, 2006.

[Claypool *et al.*, 2001] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *Intelligent User Interfaces*, pages 33–40, 2001.

[De Bra and Calvi, 1998] P. De Bra and L. Calvi. AHA! An open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia*, 4, 1998.

[De Bra and Ruiter, 2001] P. De Bra and J.-P. Ruiter. AHA! Adaptive Hypermedia for All. In *Proceedings of the WebNet Conference*, pages 262–268, 2001.

[Farzan and Brusilovsky, 2005] R. Farzan and P. Brusilovsky. Social Navigation Support in E-Learning: What are the Real Footprints? In *Third Workshop on Intelligent Techniques for Web Personalization (ITWP '05). At 19th Int. Joint Conference on Artificial Intelligence*, 2005.

[Fink *et al.*, 1996] J. Fink, A. Kobsa, and A. Nill. User-oriented adaptivity and adaptability in the avanti project. In *Proceedings of the Conference 'Designing for the Web: Empirical Studies'*, 1996.

[Goecks and Shavlik, 2000] J. Goecks and J. W. Shavlik. Learning Users' Interests by Unobtrusively Observing Their Normal Behavior. In *Proc. of the 5th International Conference on Intelligent User Interfaces*, 2000.

[Hauger and Köck, 2007] D. Hauger and M. Köck. State of the art of adaptivity in e-learning platforms. In *LWA 2007: Lernen - Wissen - Adaption, Halle, Workshop Proceedings*, pages 355–360, 2007.

[Hijikata, 2004] Y. Hijikata. Implicit user profiling for on demand relevance feedback. In *IUI '04: Proc. of the 9th int. conference on Intelligent user interfaces*, 2004.

[Hofmann *et al.*, 2006] K. Hofmann, C. Reed, and H. Holz. Unobtrusive Data Collection for Web-Based Social Navigation. In *Workshop on the Social Navigation and Community based Adaptation Technologies*, 2006.

[Jameson, 2006] A. Jameson. User-adaptive systems: A user-centered overview. In *4th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems*, 2006.

[Posea *et al.*, 2006] V. Posea, D. Mihaila, S. Trausan-Matu, V. Cristea, and A. Gartner. Evaluation of virtual learning environments using logs and social networks. In *Proceedings of the TEL-CoPs'06: 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice*, 2006.

[Putzinger, 2008] A. Putzinger. *Asynchronitätsuntersuchungen in adaptiven Hypermedia Systemen*. PhD thesis, Johannes Kepler Universität Linz, 2008.

[Sakai, 2008] Collaboration and learning environment for education. http://sakaiproject.org, 2008.