Upcoming Privacy Issues in Asynchronous Adaptive Hypermedia Systems

Andreas Putzinger*

Abstract.

This paper introduces the concept of asynchronous adaptive hypermedia systems for the first time. Furthermore, the related upcoming new features are shown and put in context of privacy.

1. Bigger picture

It is widely acknowledged that Adaptive Hypermedia Systems (AHS) can successfully be applied to several different application domains. The first summarizing taxonomy was published in [7] and later updated in [8]. The term AHS identifies systems, which " [...]build a model of the individual user and apply it for adaptation to that user, for example, to adapt the contents of a hypermedia page to the user's knowledge and goals, or to suggest the most relevant links to follow.[...]" ([7]). To be more precise adaptive techniques empower applications to feature personalization, like individual recommenders (e.g. [24]) or personal guidance functions (e.g. [27], [9], [10]). Although the community originally working on User Model research reacted to the upcoming web by establishing AHS, technologies, commonly referred to as "Web 2.0" (e.g. [19]) have yet to make their mark on the scene.

The focus of research in the author's ongoing PhD thesis is the concept of asynchronicity in AHS, whereas "asynchronous" in the context of the World Wide Web refers to the actual transmission of data, which takes place out-of-band, i.e. not within the main HTTP request–response cycle. So far it seems that to date no research group has published results yet regarding this topic, which would specifically focus on the impact of asynchronous communication in AHS, related privacy issues or

*FIM - Institute for Information Processing and Microprocessor Technology, University of Linz, Altenbergerstr. 69, Linz, Austria, email: putzinger@fim.uni-linz.ac.at

the range of upcoming possibilities. Some individual publications address very specific parts. Barla in [4] e.g. uses asynchronous techniques to get more precise information from the client's context.

2. Person Related Data in AHS

The field of AHS has by nature always been tightly connected with privacy and legal issues, because many kinds of personal information as well as user performed actions are retrieved, transmitted, processed, deduced and finally persistently stored on the server side. These data are used to model different aspects of the real world, which are mostly user specific facets, as e.g. personal preferences, skills, interests and information about the user's environment, etc. Using acknowledged and well-established techniques, as e.g. collaborative filtering (e.g. [20]), data mining algorithms, stereotyping ([22], [23], [5]), help to deduce additional information. So the system has to handle a broad variety of data, which are uniquely linked with real people. The system itself as well as the processes must therefore correlate with international, country and state law. Legal precautions must be followed, so that the required level of privacy can be guaranteed.

3. Legal situation

Legal restrictions about privacy apply as soon as a "virtual" user in a system can be mapped to a real person. Some authors have already addressed privacy issues in AHS, especially Alfred Kobsa ([15], [16] or [17], to name but a few).

The European Union has released several directives which regulate privacy issues in electronic communication. In the year 2002 the "directive on privacy and electronic communications"[1] was published, and finally implemented in Austria within "Telekommunikationsgesetz 2003 (TKG 2003)"[2], whereas the usage of browser cookies, spamming, log files and location based services are mainly addressed, to name only internet related services. § 99 addresses traffic data, which are not directly relevant to asynchronous AHS, because the transmitted data in AHS are of a higher level. According

[1] http:europa.eu.int/eur-lex/pri/en/oj/dat/2002/l_201/l_20120020731en00370047.pdf

[2] http://www.rtr.at/web.nsf/deutsch/Telekommunikation_Telekommunikationsrecht_TKG+2003

to § 101 our data are classified as "content data", which may not be stored at all, if storing is not part of the service itself or storing is not a technical requirement for fulfilling the requested service. In this case the content data must be deleted immediately as soon as the service has been fulfilled. § 96 contains the main privacy paragraphs. The main points are, that the service provider must inform the clients, which person related data are retrieved, processed and transferred. The provider must refer to the law which allows him to use the data, for what reasons he would like to do this and for what period of time the data are being stored. All this information has to be included e.g. in the general "terms of business".

The second relevant European Union directive is "on the protection of individuals with regard to the processing of personal data and on the free movement of such data"[3]. The directive was implemented in Austria in the year 2000 within the "Datenschutzgesetz 2000" (DSG 2000)[4] and defines again among other things requirements for retrieving, processing and storing data. If we exclude the so called "sensitive data" of § 4 (ethnic, political attitude, religion, etc.), which must be handled more strictly, the law defines to name all the data retrieved, collected, processed and stored in the terms of business, to mention the exact reasons why these data are needed and to inform the user concerning what happens with the data. Furthermore, the service provider has to guarantee that the data will be deleted as soon as it is no longer needed. There are further restricitions applied in DSG 2000 which will not be elaborated on due their length.

4. The Upcoming Aspect of Asynchronicity

Techniques for asynchronous communication in the web are wide–spread and provide the substantial base for many so–called Rich Internet Applications (RIA, [2]). A rather well known catchphrase in this context is AJAX (Asynchronous JavaScript and XML), which was introduced in [13]. Usually XMLHttpRequest ([26]) is used as underlying communication technology. Many different frameworks inherently support AJAX by providing high–level components and therefore abstracting from

[3] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:EN:HTML

[4] http://www.dsk.gv.at/dsg2000d.htm

technical low–level details. The following are examples of such: Dojo[5], Echo2[6], DWR[7], Google Web Toolkit GWT[8] or ASP.NET AJAX[9] to simply name some important ones. One main goal of the Web 2.0 is to bridge the gap between desktop and web applications in aspects of the look and feel, but also in communication and latency matters. Investigations about former constraints in the web, i.e. without modern Web 2.0 technologies, have been made in [6].

Asynchronous calls make it possible for a page not to be blocked and still usable during the period of actual communication. Therefore, unless the user monitors his internet activity, they may not even be aware of the fact that there are data transfers going on in the background. The asynchronously sent data per se cannot differ from synchronously transmitted data due to sandbox restrictions. The fact, however, that detailed information about user context and actions can be sent in realtime[10] can mean a deep intrusion into one's privacy, as explained in more detail in the following sections.

The question may arise as to how far this asynchronous technology actually provides new possibilities. Since asynchronously transferred data could theoretically be also transmitted synchronously by transparently bundling it with the next page–request; therefore all messages have to be collected and cached locally at the client and "piggybacked" on the next HTTP request. Yet, on closer examination this technique is not of equal potential as asynchronous transmissions. E.g. if the user manually leaves the site, closes the browser window, etc. all accumulated data from the point of entering the site until the point of leaving are lost. In addition, and this represents the main drawback, the advantage of a communication with only a short latency is lost. Thus piggybacking data is not always an alternative to asynchronous communication. Further considerations concerning AJAX security can be found in [25] and [12].

The new facet in asynchronous systems is the aspect of the point in time when the actual communication takes place. Thus, data about any inspectable property or monitored user behaviour could

[5] http://www.dojotoolkit.org
[6] http://nextapp.com/platform/echo2/echo/
[7] http://getahead.org/dwr/
[8] http://code.google.com/webtoolkit/
[9] http://ajax.asp.net/
[10] In this paper "realtime" means the fact, that data are transmitted without a long latency on the client side, regardless of the time of physical transmission. The term used here does not imply such strict criterias as e.g. in the context of operating systems.

technically be sent to the server at any time without the user's explicit approval or even awareness. In the author's opinion, it is legally relevant to add the "any time" fact to the privacy declaration, wherever person related data is transferred that way, because it does not represent the state of the art. Therefore, the user must be additionally informed in detail when and how often his performed actions, properties, etc. are monitored and transferred. In many cases it is not obvious to the user when and what data is being transferred, and therefore the user's approval cannot legally be taken as implied, because asynchronous data transfer take place in the background and the user is mostly unaware of them.

5. Impact on Functionality of AHS and Privacy

The combination of AHS and asynchronous techniques opens a great variety of new possibilities in the adaptive field and will, at least in the opinion of the author, start a new era of AHS. Not only more responsive and desktop–like user interfaces will be possible, but also enhanced adaptive technologies empowered by constant bidirectional channels between browser and server ([1]), as described in detail below.

The pool of information which can be inspected and monitored is the same as in synchronous application, because in both cases the complete variety of javascript objects can be accessed. The following enumeration shows the most important examples for new low-level techniques together with their high-level impact in adaptivity as well as upcoming privacy issues.

Monitoring the user's mouse In some cases of applications it is helpful to get realtime information about the mouse activity on the client–side, i.e. the current position of the mouse cursor, what's currently under the cursor (which text, picture, etc.), any cursor movement, miscellaneous timing data, i.e. how fast or slow the user is moving the mouse e.g. or how high the latency between double-clicks is, which mouse button is clicked on which position, etc. All these data are obviously interesting in Human Computer Interaction research (HCI) in order to remotely monitor users' behaviour while using a site and therefore to simply test certain aspects of usability. These data are also useful for user modeling purposes. First studies about recording and

evaluating mouse activity on the client side has already been done in [3]. Mouse movements are also a good example for the fact, that in many cases a single value at one certain moment in time is not relevant, but in fact the delta or the calculated trend of that value over a period of time could be significant. The information of mouse movements e.g. can be deduced from the position information of the cursor over a period of time.

The following example shows, how the position of the mouse cursor can help to retrieve valuable data for the short term user model. Many children at school portray a specific behaviour when they learn to read: they use one of their fingers as a "marker" and "guide" for the characters they are just reading, which helps them not to get lost in the text, to visually and therefore mentally focus on the specific word. Many users show a similar behaviour when using computer applications, as e.g. websites in browsers. [21] shows that if the task solving process requires mouse operations and the visual feedback of the results of these mouse operations appears close to the mouse cursor, then the visual focus and the mouse cursor position on the screen are highly correlated: between 76% and 95% correspondence. The current focus of users on a website is a very important and frequently asked-for information and could e.g. be used to unobtrusively offer context-sensitive help, etc. Further, many users have acquired a specific form of behaviour in the web, to position the mouse cursor over the object of interest, because in some cases yellow hint texts with further information appear. A consequent application of this pattern would be desirable in AHS research, because it could be an excellent source for interest derivation.

From the perspective of privacy transmitting mouse events in realtime is an extensive invasion. So the user must in any case be informed about that. The data could be processed on the server in realtime, so that the raw data are not saved permanently. Should this still be done e.g. for HCI purposes, then it is recommended to store the usage data in an anonymized form after the actual web session has ended. If the user decides not to accept the transfer of mouse data over the web, the application should be designed robust enough to handle the lack of this kind of non-vital data and should continue working as efficiently as possible.

Monitoring Key Strokes A second category of usage data contains raw key strokes, which could

be transmitted key-by-key in realtime or also bunched. So the application not only gets the finally submitted form data, but also the intermediary states, the involved timings, etc. The user's typing speed together with some other aspects can in some cases e.g. be regarded as a good indicator for the user's overall computer skills. A second example is the introduction of adaptive text completion or recommendation. Adapted to the user model the system suggests words or even complete paragraphs, which fit in with the context and presumably the user's current needs.

The two just mentioned groups (i.e. mouse events and key strokes) can individually or combined improve results in plan recognition ([11], [18], [14]) by modeling a clearer picture of the user's current activities. Furthermore the simple transmission of a key stroke or of a mouse event is an implicit indication for the fact, that the user is still working with the application, and that the application session should not timout, which e.g. sometimes accidently occurs when users write long emails in web mailing systems, because the users do not interact with the server a longer time span.

The privacy issues here are very similar to the ones previously mentioned. The user must be informed in detail about the fact, that key strokes are asynchronously transmitted, how the data are being processed, why this happens, and for how long it is stored. The application should again be built in a way that these data are treated as optional and not as vital.

"Still Alive" Messages If neither key strokes nor mouse events are asynchronously transmitted, explicit "still alive" messages could be introduced to inform the server about the fact that a person is still working with the web application, if this is really the case. This could e.g. be used in e-learning environments, where the user gets lots of texts to locally read and learn and therefore does not interact with the server in regular intervals.

If the "still alive" message does not contain any further personal information it is not absolutely necessary to add this fact to the privacy declaration, because no personal data per se are transferred or stored. Performing dummy requests to the server which uniquely identify the current web session in order to keep the session alive, is state of the art for years. Many web applications do e.g. a periodic reload of a frame whereby the "still alive" information is implicit

and for free. So there is no need to add the usage of this technique explicitely to the privacy declaration.

Subscriber Model Another concept, which becomes possible by using asynchronous techniques, is the general concept of an on-change subscriber model. The server can subscribe to certain values on the client–side. If these values change, a notification is forwarded to the server containing the property, the old and new values and a timestamp.

As an example the server could subscribe to the scroll position of the browser window. As soon as the user scrolls up, down, left or right on the page, the server gets informed about this action. This also works well with the mouse position, for example, or window width and height, content selection, etc.

Because the subscriber model is a very general concept and only becomes possible in AHS through asynchronicity, privacy issues are not directly relevant in this context, but must be treated individually in consideration of the subscribed value.

On–Demand Data Retrieval Due to the possibility of establishing a bidirectional communication channel also the server can initiate a request. Whenever some data are needed from the client, the server simply asks for it. The communication itself is of course done asynchronously in the background and even without the user's awareness.

Again privacy issues have to be solved individually in respect of the content.

Instant Adaptation The author has developed concepts which allow page fragments to be dynamically exchanged according to results of the underlying adaptive system[11]. The chosen name for this technique is "instant adaptation". Changes to the user model, which in turn cause changes on the currently shown page, can instantly be pushed to the client as fragments. Thus the possibilities for adaptation are becoming much richer. In traditional adaptive web application the actual adaptation takes place once, when the page is generated. From this point onwards, the page is static in respect of adaptation, because it is sent back to the client and no further adaptation takes place until the next complete page is generated.

[11] To be published in the author's PhD thesis

The technique of instant adaptation enables AHS to exchange parts of pages, which are already shown in the browser, on the fly by pushing this fragment to the client. Some client logics dynamically replaces, adds or modifies the specified part. This technology is new and seems to be quite powerful, but nevertheless the designer of a such system has to be very careful in using these methods. Studies have shown, that a dynamically changed user interface often confuses people and therefore does not always have a positive impact on the overall system ([28]), as for example the Microsoft Office Assistant shows[12].

Due to the fact that the nature of instant adaptation only foresees a server to client communication, privacy issues are not relevant in this context.

6. Conclusion and Future Work

This paper has presented certain parts of the results in context of the author's ongoing PhD thesis with a special focus on privacy impacts. As far as asynchronous AHS are concerned it has to be investigated how the proposed techniques can successfully be applied in real and non-academic adaptive hypermedia applications. Empirical studies have to show how stable these systems can be built because of the variety of different browsers, latencies, etc. It has also to be investigated which further information could be deduced from the retrieved raw data.

As far as privacy is concerned synchronous web applications have already had a tremendious impact due to the broad variety of inspectable information in the browser environment. The aspect of asynchronicity raises the problem to the next step. Realtime communication facilities again lower the level of privacy in the web. In theory the existing legal regulations are still good enough to handle the current situation, but in practice users are often overcharged in understanding and assessing the impact of the shown terms of business and therefore in the privacy declaration in case that these are correct or existent at all. Further, user do not always have a fair chance to decline, because they would simply be barred from the service completely instead of getting a feature-reduced version of the same application.

[12] see http://www.microsoft.com/presspass/features/2001/apr01/04-11clippy.mspx

For service providers it is now an extensive effort to legally be on the safe side, especially when the application relies on user models, which is the case in the majority of AHS. In the future efforts will be dedicated to the investigation if and to what extent it is possible to programmatically assist in the process of creating the textual terms of business regarding privacy by inspecting rules within the user modeling process. As a result the obligatory individual declaration of privacy should be created automatically. The resulting text should contain information about the quality and quantity of collected data, modalities in the retrieving process, information about the processing, deducing and storing of the data. In a further step a complete framework could be designed and implemented, which features complete processes and implements a lifecycle of privacy enhanced user modeling.

References

[1] Alessandro Alinone. Changing the web paradigm - moving from traditional web applications to streaming-ajax. *published online: http://www.lightstreamer.com/Lightstreamer_Paradigm.pdf*, 2005.

[2] Jeremy Allaire. Macromedia flash mx - a next generation rich client. *published online: http://download.macromedia.com/pub/flash/whitepapers/richclient.pdf*, 2002.

[3] Richard Atterer, Monika Wnuk, and Albrecht Schmidt. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 203–212, New York, NY, USA, 2006. ACM Press.

[4] Michal Barla. Interception of user's interests on the web. In *Proceedings of the International Conference on Adaptive Hypermedia (AH)*, volume 4018 of *Lecture Notes in Computer Science*, pages 435–439. Springer, 6 2006.

[5] I. Beaumont. *User Models and User Adapted Interaction*, chapter User modeling in the interactive anatomy tutoring system ANATOMTUTOR, pages 21–45. Springer, 1994.

[6] Alan Borning, Richard Lin, and Kim Marriott. Constraints for the web. In *MULTIMEDIA '97: Proceedings of the fifth ACM international conference on Multimedia*, pages 173–182, New York, NY, USA, 1997. ACM Press.

[7] Peter Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.

[8] Peter Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87–110, 2001.

[9] Peter Brusilovsky. Adaptive navigation support: From adaptive hypermedia to the adaptive web and beyond. *PsychNology*, 2(1):7–23, 2004.

[10] Peter Brusilovsky and Mark T. Maybury. From adaptive hypermedia to the adaptive web. *Commun. ACM*, 45(5):30–33, 2002.

[11] Sandra Carberry. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11(1-2):31–48, 2001.

[12] Stefano Di Paola. Subverting ajax. In *Proceedings of 23rd CCC Conference, Berlin*, 2006.

[13] Jesse James Garrett. Ajax: A new approach to web applications. *published online: http://www.adaptivepath.com/publications/essays/archives/000385.php*, 02 2005.

[14] Kristina Hook. Evaluating the utility and usability of an adaptive hypermedia system. In *IUI '97: Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 179–186, New York, NY, USA, 1997. ACM Press.

[15] Alfred Kobsa. Tailoring privacy to users' needs. In *UM '01: Proceedings of the 8th International Conference on User Modeling 2001*, pages 303–313, London, UK, 2001. Springer-Verlag.

[16] Alfred Kobsa. Personalized hypermedia and international privacy. *Commun. ACM*, 45(5):64–67, 2002.

[17] Alfred Kobsa. *Privacy-Enhanced Web Personalization*. Lecture Notes in Computer Science, Vol. 4321. Springer, Berlin, 2007.

[18] H. Kristina, J. Karlgren, A. Waern, N. Dahlback, C. Jansson, K. Karlgren, and B. Lemaire. A glass box approach to adaptive hypermedia, 1996.

[19] Tim O'reilly. What is web 2.0? *published online: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html*, 09 2005.

[20] Stefano Guida Antonio Piccinno Giuseppe Tesoro Paolo Buono, Maria Francesca Costabile. Integrating user data and collaborative filtering in a web recommendation system. In *Proceedings of the Eight International Conference on User Modeling, 2001*, 2001.

[21] Matthias Rauterberg and Christian Cachin. Locating the primary attention focus of the user. In *VCHCI*, pages 129–140, 1993.

[22] E. Rich. User modeling via stereotypes. *Cognitive Science*, 3:329–354, 1979.

[23] Elaine Rich. Users are individuals: Individualizing user models. *International Journal of Man-Machine Studies*, 18(3):199–214, 1983.

[24] Alan F. Smeaton and Jamie Callan. Personalisation and recommender systems in digital libraries. *International Journal on Digital Libraries*, V5(4):299–308, 2005.

[25] Michael Sonntag. Ajax security in groupware. In *32nd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO'06)*, 2006.

[26] Anne van Kesteren. The xmlhttprequest object. *http://www.w3.org/TR/XMLHttpRequest/*, 2007.

[27] G. Weber and M. Specht. *User Modeling and Adaptive Navigation Support in WWW-based Tutoring Systems*, pages 289–300. Springer–Verlag Wien, 1997.

[28] Jun Xiao, John Stasko, and Richard Catrambone. An empirical study of the effect of agent competence on user performance and perception. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 178–185, Washington, DC, USA, 2004. IEEE Computer Society.