Interception of User's Interests on the Web

Michal Barla[*]

Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
barla@fiit.stuba.sk

Abstract. Current adaptive systems acquire information about users mainly by simple tracking of resources, a user has requested and by asking users to supply the needed information. In this paper, we discuss user modeling based on observing a user's interaction with the system. We propose to collect usage data on the server side as well as on the client side. Collected data are then processed into knowledge about user's intentions and preferences. This processing relies on a set of heuristics, which help to interpret the usage patterns found in the collected data.

1 Introduction

Adaptivity is becoming ever more important feature of web-based systems. An adaptive system reflects the particular needs of an individual user in a particular context and improves the efficiency of the user – system interaction. It is a response to the permanent information growth on the Internet, where finding the right information becomes difficult and time consuming.

Each adaptive system can only perform personalization if it already has some knowledge about the user. This knowledge is stored in various attributes in the user model. As the user continues to use the system, additional knowledge is acquired and added to the user model resulting in better adaptation. This leads to the cyclic loop "user modeling – adaptation" in an adaptive system, as mentioned in [1].

Our work focuses on the user modeling part of adaptive systems. Many user modeling systems gain information about users by simply asking them, however we chose to focus on an approach based on user observation. This includes the collection of data about user activity and the transformation of this data into knowledge about the user – creating the user model. We identify the main problems in this area and discuss possible solutions. Results of our work, as part of the project [2] are verified in the domain of job offers in a system used for job finding.

The paper is structured as follows. Section 2 discusses the approaches of gathering information about user actions in the information space in a non-intrusive manner. Next, in Section 3, we describe the methods used to transform acquired usage data to the user model. Finally, we draw some conclusions.

[*] Supervisor: prof. Mária Bieliková

2 Data Collection

There are several ways to acquire information about a user which serve for user model constructing. One is to monitor the user's interaction with the system – logging each user action for further analysis. One main drawback is the unreliability of user characteristics deduced from the acquired information, because there is no explicit relation between user actions and characteristics in the user model. However, this approach has the advantage that the system does not force the user to provide information explicitly, but instead it implicitly gathers the sequence of user actions during a session and interprets the acquired data to make statements about the user.

There are several approaches to user monitoring, which can either be performed on the server side or on the client side of the system. As a third option, one can combine both of these approaches.

Server side monitoring tracks user requests for resources. Its main drawback is that it does not provide precise time-related data, because it relies on the behavior of web browsers, which usually do not re-demand an already visited page from the server, but instead use the copy stored in the local cache. Thus the system does not know the exact time that the user spent viewing a certain page. The current most widespread web browsers do not respect the cache-control directives of the HTTP protocol forbidding the use of the local cache, so they cannot be used to bypass the cache problem. It is also mentioned in [3] that client side monitoring is necessary to get the precise records about a user's interaction with a system. Despite this, server side monitoring is still suitable for many adaptive systems. For example, AHA![1] uses server side logging to track what reading material is presented to the user [4].

Client side monitoring allows for the creation of a detailed log of user actions with exact timestamps. It can be performed by a specific client side application (e.g., User Action Recorder in [5]) or by employing a client web technology like JavaScript or Java applets. Since we consider the first approach as very invasive and not flexible enough, we focus on the second approach. The mentioned web technologies are common in the majority of web browsers on all major platforms. The possible drawbacks are that not every user accepts this kind of detailed monitoring and some of them block the execution of embedded scripts. Furthermore users may not have the necessary software installed on their computers (e.g., Java virtual machine).

Several tools with support for client side logging exist that exploit JavaScript such as WebVip[2] or WET[3]. Both tools are primarily designed for the purpose of web site usability evaluation. These tools are either too focused on the evaluation process or demand the entire copy of the web site for their operation. Therefore, we developed our own client side logging tool based on JavaScript combined with

[1] Adative Hypermedia for All, `http://aha.win.tue.nl/`

[2] Web Variable Instrumenter Program,
`http://zing.ncsl.nist.gov/WebTools/WebVIP/overview.html`

[3] Web Event-logging Tool, [6]

the DOM2 event handling and asynchronous server communication using AJAX technology.

To summarize, it is not possible to gather any data if the user is not willing to enable client side user monitoring, which is a strong argument against the sole use of the above mentioned tools. On the other hand, server side monitoring is more reliable since it always acquires some data, but carries the risk of loosing precious time-related information. Our approach is based on the idea of combining the two aforementioned approaches – on the extraction of a maximum amount of data from the server log and on the use of the client log as a source of optional additional, precise information about the user's activity.

3 Data Analysis

After the data collection stage, we are supposed to transform the sequence of user's activities into statements about her cognitive processes. In another words, we have to determine non-behavioral meanings, which are either implied by or associated with the users' behavior [7] (e.g., to find out user goals, estimate user knowledge about certain concepts). The binding between actions and cognitive processes is not deterministic and is never definite. This is why the problem is widely discussed in the user modeling community (e.g., [8, 7]).

Patterns and Heuristics. When interpreting the data we look for interesting usage patterns, which describe the implicit feedback of the user. We analyze the sequences of "clicks" on the web-site and usage of the *back* button in the browser. We use sequential pattern mining algorithms to find such sequences of actions that differ only slightly from the predefined ones. For the initialization of the system, we plan to define patterns related to the user's goal and evaluate these patterns as the system will be used by real users.

We assign higher weights to the "first click" of the found sequence, as it usually has stronger relation to the user's intentions than the rest of the sequence. Successively we identify the appropriate usage pattern and use heuristics to infer user characteristics (see fig. 1).

An example of a simple heuristic in the domain of job offers is: "If a user chose to view at least "sufficient number" of offers from sector A (e.g., health-care or IT), raise the relevance level of this sector in the model of the user's ideal job offer".

During the analysis stage we also consider the navigation model of the system, which actually determines the possible sequences of user actions and thus makes all heuristics system-specific. Educational systems with sequential structure of pages forming an e-course, would have different usage patterns compared to a job offer portal, whose content is not sequential. The system must support easy navigation and searching in the information content, what results in a hierarchically organized navigation structure of the portal.

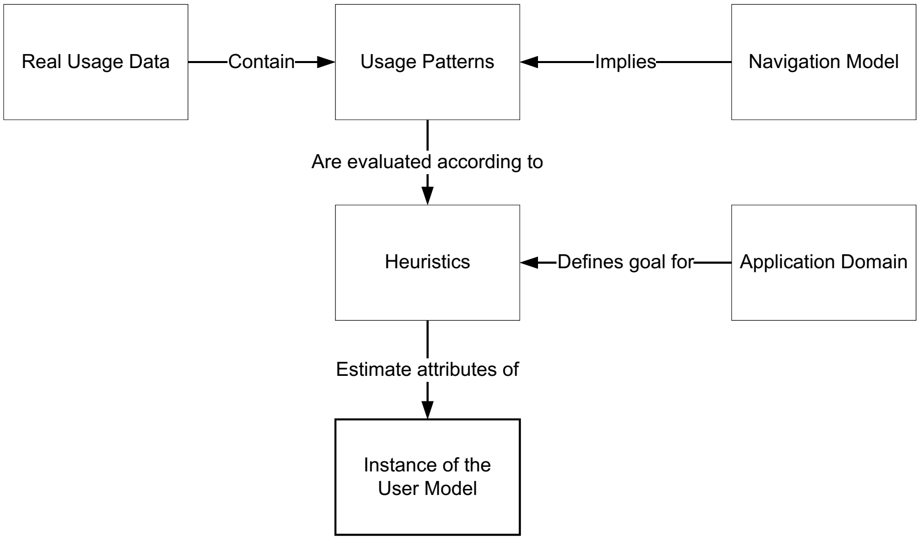

Fig. 1. Sources for creating an instance of the user model. We search for interesting *Usage Patterns* in the *Usage Data*. These patterns are determined by the *Navigation Model*. Knowing the related *Heuristics*, we can evaluate the located *Usage Patterns* to estimate attributes of an *Instance of the User Model*. *Heuristics* are bound to the goals of the user, determined by the *Application Domain*.

Relations between concepts. User actions in the context of adaptive web-based systems can be regarded as navigation between concepts. Our idea is that by comparing the visited concepts and finding out their common and different aspects, we may gain knowledge about reasons (user preferences), why a user reacted differently to each of them. This comparison can either point out the values of different attributes of two concepts or compute their distance.

The distance represents the measure of dissimilarity of two concepts. For instance, C# is different from JAVA but it is definitely closer – less different to JAVA than to Lisp. A heuristic, which use the differences of two concepts must consider the distance of these concepts and its impact on the user model.

Afterward, it is possible to estimate user characteristics from the different or similar user actions related to the compared concepts. For instance, if the user "refuses" one offer but "accepts" another and these offers are quite close to each other, with their main difference being in the duty location, we can surmise that the user prefers the region from the second offer.

4 Conclusion

We described our research in the field of user modeling that is focused on user modeling based on user observation, which comprises the collection of information about the user activities within an information system and the successive analysis of the acqired information to create a user model.

We have identified several ways of user activity data acquisition, where client side monitoring appears to be the most efficient based on the richness of data, but also has a serious drawback in the unreliability of execution. Therefore, we use a combination of client and server side monitoring to achieve good results.

Analysis of the acquired data transforms the acquired user behavior into the knowledge about user characteristics or about user goals. We identified aspects, which influence the creation of heuristics that estimate some user characteristics from the recorded usage patterns.

Future work includes the design and verification of a method for the creation of an instance of the user model based on analysis of server and client side logs. This method would map the preferences of a user to a particular sequence of actions, use the comparison of the visited concepts to reveal user preferences and semi-automatically fill the user model with relevant data. To verify our design, we evaluate it with the user model used in the project [2] for adaptation in the domain of job offers.

Acknowledgments

This work was partially supported by the Science and Technology Assistance Agency under the contract No. APVT-20-007104 and the State programme of research and development “Establishing of Information Society” under the contract No. 1025/04.

References

1. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction **6**(2-3) (1996) 87–129
2. Návrat, P., Bieliková, M., Rozinajová, V.: Methods and tools for acquiring and presenting information and knowledge in the web. In: International Conference on Computer Systems and Technologies – CompSysTech' 2005, Varna, Bulgaria (2005)
3. Paganelli, L., Paterno, F.: Intelligent analysis of user interactions with web applications. In: IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces, New York, NY, USA, ACM Press (2002) 111–118
4. Bra, P.D., Calvi, L.: AHA: a generic adaptive hypermedia system. In: 2nd Workshop on Adaptive Hypertext and Hypermedia. (1998) 5–12
5. Thomas, R., et al.: Generic usage monitoring of programming students. In Crisp, G., Thiele, D., Scholten, I., Barker, S., Baron, J., eds.: 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE). (2003)
6. Etgen, M., Cantor, J.: What does getting wet (web event-logging tool) mean for web usability? (In: 5th Conference on Human Factors & The Web)
7. Judd, T., Kennedy, G.: Making sense of audit trail data. Australasian Journal of Educational Technology **20**(1) (2004) 18–32
8. Kay, J., Lum, A.: Creating user models from web logs. In: Intelligent User Interfaces Workshop: Behavior-Based User Interface Customization. (2004)