A Model-Driven Architecture for Logging Navigation

Marco Winckler, Florence Pontico
LIIHS – IRIT, Université Paul Sabatier
118 route de Narbonne
31062 Toulouse, France
{winckler, pontico}@irit.fr

ABSTRACT
In this paper, we present a method for tracking information concerning the user navigation over Web sites based on a Model-Driven Architecture (MDA). We present a case study which demonstrates such as an approach and we discuss how it can be employed to improve the user interface design of Web interfaces.

Categories and Subject Descriptors
H.5.4 [**HCI**]: Hypertext/Hypermedia – H.5.2 [**HCI**]: User Interfaces.

General Terms
Measurement, Design, Human Factors.

Keywords
Web applications, Model-Based Architecture, Navigation, User Interfaces.

1. Introduction
The distributed nature of Web applications facilitates the collection of remote users' activity by tracing their interactions in log files. The log file analysis has been identified as a remarkably rich source of genuinely useful information concerning the user interaction with Web applications [5] since it takes the advantage of getting information from real users in their natural work environment.

Tools for inferential analysis of Web log files has become very popular in the last years at providing access hit of web sites, most requested elements on the Web site (files, information, etc), temporal analysis of requests, paths navigated by the users and so on. Automated tracking of user activity has been motivated by the development of the marketing over the Web, especially e-commerce [2], but not only, since it has also been used to improve the user interface design [7][13]. Thus, over the past years, the data collection over the Web had become a major issue for understanding the usage of Web applications. Currently these are several tools implementing methods for analysing log files based on statistical analysis [13], on-line analytical access [1] and/or visualisation techniques [3] [4][12]. Despite the diversity of data that can be collected by automated tools, in this paper we only analyse the data concerning the navigation between pages of the Web site.

The analysis of Web log files is usually cheap and it provides fast results when compared to other techniques for observing user activity [8]. However due to the client/server architecture of the Web, the data collection is not as easy as it might seam. Data collection on the client-side often requires specific plug-ins and/or browser-specific implementations. The users can deny the authorization to run client-side tracking tools, thus avoiding the collection of interaction data. The problem with server-side approaches is that proxy and cache can mask several users requests thus biasing the statistics of access to the Web server and putting down any tentative of understand the navigation path followed by the users.

In this paper we investigate a completely different approach for tracking user activity concerning the navigation of Web sites. In the past few years, the Software Engineering community has devoted a lot of work to the Model-Driven Architecture (MDA) [11] approach. Following the initial impulse of the OMG towards MDA, several advanced tools and environments have been developed to support meta-modeling, model-transformation languages that allow reconciling models that describe different facets of the system under design. Most recently MDA has been introduced into the context of Web to help the development of complex Web applications.

In the following sections we introduce a MDA approach for Web sites and we explore the models used to conceive the applications to track the user activity over the Web. Doing so, we can avoid both client-side and server-side problems for collecting data.

2. A Model-Driven Approach for the Web
To deal with the complexity of Web development, modelling support is essential as it provides an abstract view of the application thus leaving details to later phases in the development process [10]. By means of a formal description technique, models can help designers by decomposing complex applications in smaller and manageable parts, increasing communication into development team, reducing ambiguity, and providing support for verification prior to implementation.

Following a model-base approach presents several advantages for the effective Web design. Navigation model can not only be used to build the application but it can also be employed to control the dialog between the user and the application. To illustrate this approach, the next section provides a small case study;

2.1 Navigation Modeling of Web sites
Hereafter we present a case study for the Web site of the SpiderWeb project[1]. This Web site contains information about the project members, publications, current activities, and hyperlinks to partners' websites. The Web site contains just 9 pages, mainly static, but it is enough to illustrate the concept.

The Figure 1 shows the navigation modeling using the SWC notation [14] which describes the relationships between the pages. SWC notation is a formal description technique based on the Harel's StateCharts [6]. Nodes in SWC represent Web pages while transitions represent links between pages.

[1] http://liihs.irit.fr/projects/spiderweb/ (on the 3/17/06)

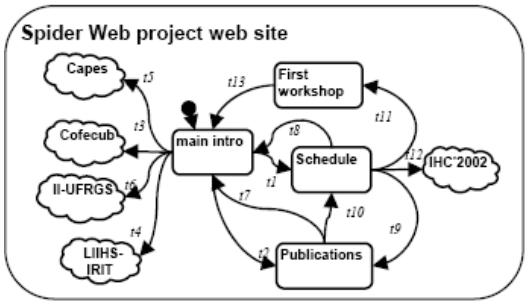

Figure 1. SWC navigation model of the SpiderWeb website

SWC is typically a finite state machine. At the highest level, a composed state *Spider Web project web site* groups all the states that are part of the website. They are represented as rounded squares (*main intro*, *first workshop*, *schedule*, *publications*) and correspond to static states according to the notation, that is, in the Web semantic, pages whose content is static – in contrast with dynamic pages, generated from the user's form inputs for example. The foreign pages that are the ones that don't belong to SpiderWeb website (i.e. *Capes*, *Cofecub*, *II-UFRGS*, *LIIHS-IRIT*, *IHC'2002*) are represented by clouds and correspond to external states in the SWC notation. These pages are out of the control of the designer, but are even so modeled in the navigation model of the application to avoid broken links.

The transitions between the different states (from *t1* to *t13*) correspond to hyperlinks between the pages. The *main intro* state is defined as the initial state that is the default state of the model. This concept is, among others, inherited from StateCharts.

2.2 Dialog Controller Architecture

Our approach for models relies on the complete separation of the elements compounding a Web site. In fact, our model-based architecture is actually made of three levels:

- The navigation model, which describes interconnections between website pages;
- The presentation model, which defines the graphical appearance of the website;
- The contents, which contains the data displayed in the website.

Technically speaking, the *contents* of the website are described as individual text files (XHTML) where each file correspond to the content of a single page, the *presentation* level is a style sheet (CSS) used for all pages, and the *navigation* is represented by a SWC model (XML format). A dialog controller is in charge of merging these files on execution time according to the user requests. The architecture of the dialog controller is presented in the Figure 2.

When a user requests a web page, the navigation controller engine execute the following steps:

1. Intercepts the request;
2. Records a trace of the interaction;
3. Handles the request.

When the requested page belongs to the website, the controller merges the three application levels to build the page: the presentation pattern is applied to the contents and the hyperlinks available from this page – according to what is described in the navigation model – are inserted into the contents. This page is finally downloaded on the client's computer and displayed by his browser. When the requested page belongs to a foreign website, the controller redirects the user to the required URL.

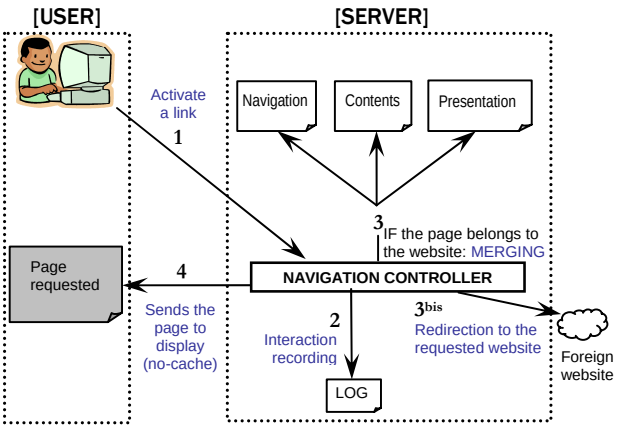

Figure 2. Model-based dialog architecture suggested

In order to exemplify how it works in the practice, the figure Figure 3 presents the transitions to states reachable from the *main intro* state.

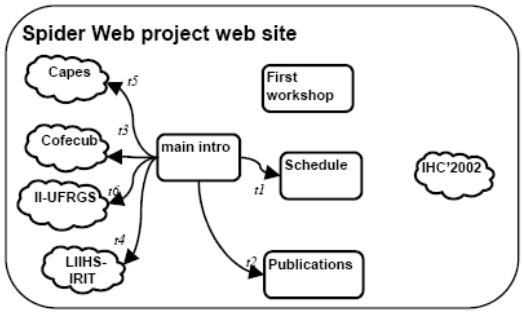

Figure 3. SWC navigation model of the SpiderWeb website: set of available transitions from the *main intro* state

Actually, when a user asks for the displaying of a page (here *main intro*), the navigation controller identifies on the SWC model the set of transitions available from it and generate on the fly the corresponding links allowing the navigation. In our case study, the navigation links are presented by the means of a navigation bar on this base. This bar is then integrated in the *main intro* page that will be downloaded on the user's computer. The page built after this merging is presented in Figure 4. The graphical definition of the generated page (e.g. font and color of the text) comes from what is described in the presentation model.

It is worth noticing that if a transition is deleted in the navigation model, it will also disappear from the navigation bar. Similarly, if the presentation model is updated, the modifications are taken into account immediately. This principle assures a total conformity between the modeling and the actual implementation.

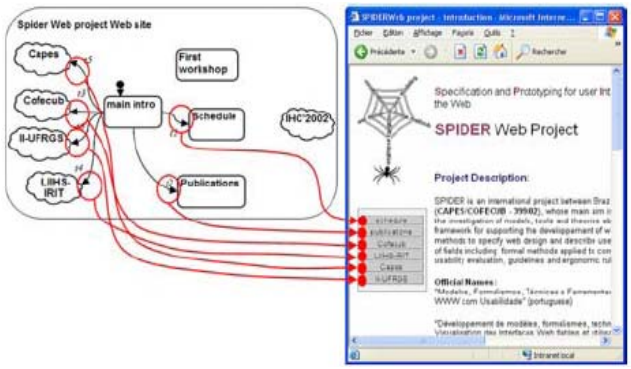

Figure 4. HTML page *main intro* after the navigation bar generation from the SWC model

Each link of the navigation bar, which encodes a parameter containing the name of the corresponding transition on the navigation model, is directed to the navigation controller. So, when a link is activated by the user, the dialog controller receives a request with a kind of signature allowing the verification over the model the last state visited. If users use the browser toolbar (e.g. buttons Refresh or Back) none parameter is send back to the dialog controller which can easily infer these user actions.

2.3 Logging interaction traces

All user navigations over the Web site are recorded by the navigation controller (cf. LOG component in the Figure 2). Thanks to the parameters encoded at each transition, the navigation controller is able to determinate which link has been activated on the navigation bar, or if the user has used the buttons Refresh or Back.

Let's consider the following scenario:

"The user starts at the *main intro* homepage. He wants to consult the details about the *workshop*, the *publications*, and then leave to the LIIHS-IRIT website. He actually goes through the path sequence: *schedule (1)*, *workshop (2)*, *main intro* again (3), *publications (4)*, *main intro* (5) this time using the Back button of the web browser, and finally *LIIHS-IRIT* (6)."

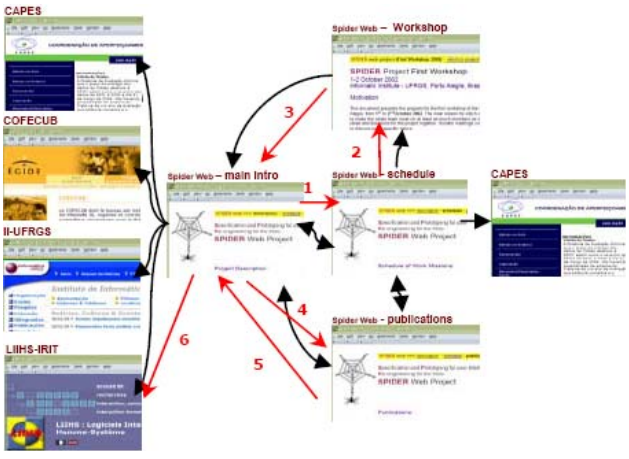

Figure 5. User's path throughout the SpiderWeb website.

The Figure 5 presents the scenario over a graph describing the navigation over the web site. The corresponding log file, as it is recorded by the dialog controller, is presented in the Figure 6.

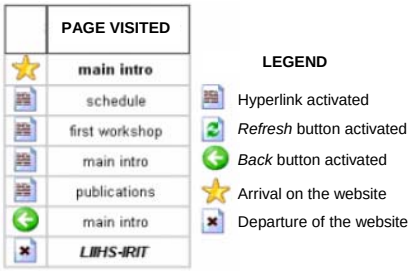

Figure 5. Data collected during the scenario execution of the case study.

3. Discussion

Model-Driven Architecture is a very promising field for research and development over the Web since it provides, by the means of models, the adequate support for deal with complex applications [9]. In addition, models can be checked before the implementation to ensure the correctness of requirements.

In this paper, we have shown how we can take the advantage of a MDA approach to trace the user navigation user a dialog controller. This kind of log files does not present the limitations presented by client-side solutions since it is completely managed by the application. It does not suffer of the limitation of the Cache and Proxy since every page should be generated on-the-fly, so that every user request must be executed by the dialog controller. In addition, since we can compare the requests according to the navigation models is quite easily to detect the cases when users are using browsers buttons such as the Back button. This kind of information is especially useful for usability studies.

In this paper we have presented a small case study which demonstrates the concept. Extension implement the same approach can be done to support larger web sites and including not only navigation issues but also data flow over Web applications.

4. ACKNOWLEDGMENTS

This work is partially supported by the WebAudit project.

5. REFERENCES

[1] Büchner, Alex C.; Mulvenna, M. D. (1998). Discovering Internet marketing intelligence through online analytical web usage mining. ACM SIGMOD Record 27, 54-61.

[2] Cheung, C. M. K., Chan, G. W. W. and Limayen, M. A. A Critical Review of Online Consumer Behavior : Empirical Research. Journal of Electronic Commerce in Organisations, Vol. 3, N.4, 2005.

[3] Chi, E. H., Pitkow, J., Mackinlay, J., Pirolli, P., Gossweiler, R. and Card, S. K. (1998). Visualizing the evolution of Web ecologies. In Proceedings of the SIGCHI conference on Human factors in computing systems, ed. anonymous, pp. 400-7: ACM Press/Addison-Wesley Publishing Co.

[4] Cugini, J. and Sholtz, J. (1999). VISVIP: 3D Visualization of Paths through Web sites. In Proceedings of the International

Workshop on Web-Based Information Visualization WebVis'99, ed. anonymous, pp. 259-63, Florence, Italy: IEEE Computer Society.

[5] Guzdial, M., Santos, P., Badre, A., Hudson, S. and Gray, M. (1994). Analyzing and visualizing log files: A computational science of usability. GVU Center TR GIT-GVU-94-8, Georgia Institute of Technology. Available at: ftp://ftp.cc.gatech.edu/pub/gvu/tr/1994/94-08.pdf

[6] Harel, D. Statecharts: a visual formalism for complex systems. Science of computer Programming, vol. 8; 1987.

[7] Ivory, M. Automated Web Site Evaluation. Kluwer Academic Publishers. Bordrecht, The Netherlands. 200p. 2003

[8] Kohavi, R. Mining e-commerce data: the good, the bad, and the ugly. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, 2001. San Francisco, USA. ACM Press.

[9] Kurtev, I., van den Berg, K. Model driven architecture based XML processing, Proceedings of the 2003 ACM symposium on Document engineering, November 20-22, 2003, Grenoble, France.

[10] Murugesan, S., and Deshpande, Y. Web Engineering: Managing Diversity and Complexity of Web Application Development, Berlin: Springer. (2001)

[11] OMG. Model-Driven Architecture. Available at: http://www.omg.org/mda/

[12] Pitkow, J. & Bharat, K. (1994). WebViz: A Tool for World Wide Web Access Log Analysis.

[13] Sullivan, T. (1997). Reader Reaction: A proposal for Inferential Analysis on Web Server Log Files. In 3rd Conference on Human Factors & the Web, ed. anonymous, Boulder, CO.

[14] Winckler, M. and Palanque, P. StateWebCharts: a Formal Description Technique Dedicated to Navigation Modelling of Web Applications. International Workshop on Design, Specification and Verification of Interactive Systems - DSVIS'2003, Funchal, Portugal, June 2003. (Lecture Notes)