ALF: A Client Side Logger and Server for Capturing User Interactions in Web Applications

Myles Doolan, Leif Azzopardi
School of Computing Science
University of Glasgow
United Kingdom
{ 0805976D@student, leif@dcs }.gla.ac.uk

Richard Glassey
School of Computing
Robert Gordon University
United Kingdom
r.j.glassey@rgu.ac.uk

EXTENDED ABSTRACT

Modern day websites offer applications with sophisticated user interfaces and dynamically interactive data [1]. Understanding how a user interacts with a search service, in particular, is an effective way to determine the usability and quality of the search experience [1, 2]. Gathering the necessary data to analyse user interactions with a web based search application can be done either server side and/or client side. Server side solutions provide a course grained approach to logging specific discrete events such as when a query is issued, a web page is viewed, etc. Client side solutions, however, can provide a much richer set of data regarding the specific micro interactions performed by a user given the web application interface (i.e. when there is a mouse over, a scroll, a click, a key press, etc). While a number of logging applications have been previously developed, such as UsaProxy [1], MLogger [2] and Wrapper [3], there are number of issues with these loggers. UsaProxy only intercepts calls between a client/web browser and the webserver and requires the client to set up a proxy. While this is appropriate for lab based studies, it does not work well in less controlled setups. No implementation of MLogger is available (and it is based on Moschkit which is no longer supported), while Wrapper is a Windows/IE6 logger and requires client side installation. Other notable loggers require client side setup (e.g. Lemur Query Log Toolbar and CrowdLogger both require browser plug-ins), do not provide any facilities to remotely capture interaction data or are not freely available to download/use (e.g. Morea and Camtasia).

This demonstration paper introduces ALF which provides a light-weight client side logging application and a server for collecting user interaction data. ALF has been designed as a loosely coupled independent service that runs in parallel with the IR web application that requires logging. Figure 1 shows how the services are related, where (1) the IR application requests ALF to log interactions, this set ups (2) a record of the applications currently being logged, and returns a piece of Javascript (alf.js) to be included in the client interface (with certain parameters). When the Web Search interface is used, the ALF Javascript on the web interface client sends interaction data to the ALF Server. On receipt of the data an acknowledgment is send back (4), if there is no acknowledgment the data is re-sent. Once received the interaction data for that application is checked and then logged (5). The ALF javascript added to the client is parameterized with the application's name along with which interactions on the client side to log. Currently, ALF supports the logging of: Cursor entering/leaving elements, mouse clicks, key presses, scrolling within the page and within elements.

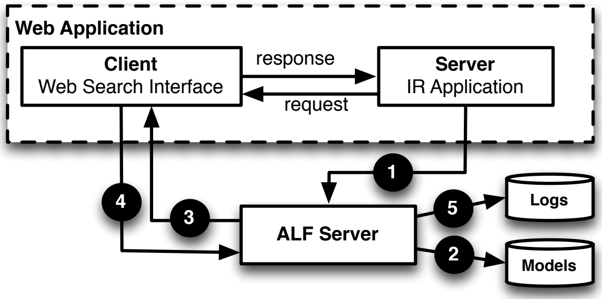

Figure 1: How ALF fits with an Web Application.

On the server side, ALF receives and then logs the datetime, IP address, the username (if a cookie called username exists), the event, and any associated data with the event i.e. the DOM id of the element in the page, the (x,y) coordinates on the page, the key pressed, etc. All the data is stored in JSON files for easy subsequent processing and analysis. ALF provides some basic visualizations of the data collected through a web interface so that researchers can inspect the data as it is collected. The client side scripts and back-end server is available for download along with an example web search interface called MySe (short for MySearch) that connects to a Bing backend[1]. In sum, ALF provides developers and researchers with the tools to log client side interactions without the need to install any client side software and to remotely collect and store interaction data via an ALF server.

Categories and Subject Descriptors:
H.3.3 [**Information Storage and Retrieval**]:
Information Search and Retrieval
General Terms: Design, Human Factors
Keywords: Information Retrieval, Implicit Logging

REFERENCES

[1] R. Atterer, M. Wnuk, and A. Schmidt. Knowing the user's every move: user activity tracking for website usability evaluation. In *Proc. of the 15th WWW*, pages 203–212, 2006.

[2] A. Edmonds, R. W. White, D. Morris, and S. M. Drucker. Instrumenting the dynamic web. *J. Web Eng.*, 6:244–260, 2007.

[3] Ramadoss R. Zhang M. Jansen, B. J. and N. Zang. Wrapper: An application for evaluating exploratory searching. In *SIGIR 2006 Workshop on Evaluating Exploratory Search Systems.*

Acknowledgements: PuppyIR is funded by the EC's FP7 2007-2013 under grant agreement no. 231507.

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.
ACM 978-1-4503-1472-5/12/08.

[1] See `http://www.dcs.gla.ac.uk/access/alfproject/`