IRM PRESS
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.irm-press.com
ITB9281

Chapter XVI

Building a Custom Client-Side Research Tool for Online Web-Based Experiments

Stu Westin
University of Rhode Island, USA

ABSTRACT

This chapter describes a general software-based approach to conducting online Web research through the development of a custom research tool. Specifically, the tool is an Internet Explorer-like Web browser that can be designed to deliver experimental treatments and to collect experimental data with great precision and flexibility. The purpose of the manuscript is to introduce this approach to Web-based research, and to discuss the most salient issues, techniques, and problems that are involved in the development and use of such a research instrument.

Programming custom event handlers, for a preexisting software object called the WebBrowser Control, constitutes a major part of the research approach. Event handling techniques having to do with downloading and navigation, with browser interface emulation, and with window and session control are presented. Other relevant issues such as cache management, keyboard handling, and accessing HTML page elements through the Document Object Model are also presented.

INTRODUCTION

Research studies involving the use of the World Wide Web are becoming increasingly common in disciplines such as MIS, marketing, and e-commerce. The focus of these studies is quite varied and may involve issues of human factors (e.g., how does download time impact Web use?), issues of information processing (e.g., what search strategies are employed in various situations?), issues of information content (e.g., how much detail should be provided in the initial product description in an e-commerce application?), or a myriad of other questions. Regardless of the issue being studied, data collection for online Web research often proves to be a vexing problem, and ideal research designs are frequently sacrificed in the interest of finding a reasonable data collection mechanism. Similarly, the administration of experimental treatments under precisely controlled conditions is hardly a trivial exercise when the Web is involved. The researcher is often forced to sacrifice external validity in an attempt to obtain a reasonable degree of internal validity in these experiments.

This chapter describes a general software-based approach to conducting online Web research through the development of a custom research tool. The purpose of this discussion is not to describe a specific research design, or to detail a particular research methodology. Rather, the motivation is to introduce this rather atypical approach to the research area, and to discuss some of the salient issues, techniques, and problems that are involved. Much of the material contained herein was developed as the author struggled with this very topic. It is hoped the reader will be provided with some valuable insights into what to expect from the custom research tool, and how to proceed with its development.

BACKGROUND

Understanding the Problem

Server-side data collection mechanisms based on active server page (ASP) scripts or the like can prove useful in some research circumstances. Consider, for example, the situation where you want to investigate the impact of download time on user satisfaction. Using ASP scripts, a delay mechanism can be easily built into a Web page so that the server will delay serving the requested page to the client until some precise, predetermined time has passed. Different experimental treatment levels are accomplished by merely manipulating the delay time that is scripted into the Web page. Here, the experimental subject, using an ordinary browser, will have the perception that the page is slow to download because of the delay between when the page is requested (e.g., by clicking a hyperlink) and when the page is available in the browser. As another example, consider the situation where you want to study the end user's Web search strategy by recording which pages are accessed, along with the sequence of page access. In this case, we need to record the so-called "click-stream data." Again, ASP scripts in the Web pages could provide a simple