

Using NetCloak to develop server-side Web-based experiments without writing CGI programs

CHRISTOPHER R. WOLFE
Miami University, Oxford, Ohio

and

VALERIE F. REYNA
University of Arizona, Tucson, Arizona

Server-side experiments use the Web server, rather than the participant's browser, to handle tasks such as random assignment, eliminating inconsistencies with JAVA and other client-side applications. Heretofore, experimenters wishing to create server-side experiments have had to write programs to create common gateway interface (CGI) scripts in programming languages such as Perl and C++. NetCloak uses simple, HTML-like commands to create CGIs. We used NetCloak to implement an experiment on probability estimation. Measurements of time on task and participants' IP addresses assisted quality control. Without prior training, in less than 1 month, we were able to use NetCloak to design and create a Web-based experiment and to help graduate students create three Web-based experiments of their own.

In recent years, there has been a revolution in the use of the Internet to collect behavioral data in virtually every area of psychology (e.g., Bailey, Foote, & Throckmorton, 2000; Birnbaum, 1999; Buchanan & Smith, 1999; Krantz, Ballard, & Scher, 1997). The ubiquity of the Internet, and particularly the World-Wide Web (Web), means that researchers have the potential to collect data from heterogeneous populations around the world and to recruit from special populations, ranging from the clergy to the color blind. It is not uncommon for researchers to report on Web-based experiments with more than 1,000 participants (e.g., Birnbaum, 1999). However, accompanying the diversity of populations is a tremendous heterogeneity in experimental conditions, including a variety of computers, operating systems, monitors, network connections, and browsers.

Researchers have developed a number of techniques for creating Web-based experiments, including the use of commercial software and plug-ins, such as Authorware and ShockWave (e.g., Williams, McGraw, & Tew, 1999; Wolfe, 1992), Java and JavaScript (see Baron & Siepmann, 2000), and common gateway interface (CGI) programs scripted in programming languages such as Perl and C++ (e.g., Morrow & McKee, 1998). The majority of Web-based experiments rely on client-side applications, especially Java and JavaScript. Unfortunately, Java and JavaScript have not been consistently implemented across Web browsers. Although a settlement of the Microsoft antitrust

suit may reduce the variability caused by Java implementations, at this writing, the lack of consistent cross-platform standards remains a serious issue. Indeed, browsers older than Netscape 3.0 and Internet Explorer 3.0 may not run Java and JavaScript at all (Baron & Siepmann, 2000). Moreover, according to a 1999 Georgia Tech Graphics, Visualization, & Usability Center study, 15% of respondents use nonstandard browsers (i.e., those other than Netscape and Microsoft products) and expect to continue using them in the future, and "43% of users reported that JavaScript and Java incompatibilities were among the biggest problems they encountered while using the Web" (reported in Schmidt, 2000, p. 292).

Server-Side Implementation With NetCloak

Server-side implementations are a means of getting around many of the problems with client-side applications such as Java. Server-side experiments are those in which the Web server, rather than the participant's computer, handles tasks such as randomly assigning subjects to experimental conditions, differentially presenting stimuli to participants, collecting data about the participant's behavior (including time on task, IP address, referring Web site, browser, and operating system) and putting those data into files for subsequent statistical analysis. Having data processed on the server gives the researcher greater control over both the presentation of stimuli and the collection of responses. Heretofore, experimenters wishing to create server-side experiments have had to borrow CGI scripts, use specialized software that modifies scripts for specific purposes (e.g., WWW Survey Assistant; Schmidt, 1997), or exercise expertise in programming to create CGIs of their own. Indeed, even JavaScript-based experiments generally use CGIs to store data in text files. Diffi-

We express our thanks and appreciation to Hays Cummins for all of his assistance with WebSTAR and NetCloak. The opinions expressed herein are those of the authors and do not represent those of the U.S. Department of Education. Correspondence concerning this article should be sent to C. R. Wolfe, Western College Program, Miami University, Oxford, OH 45056 (e-mail: wolfecr@muohio.edu).

culties associated with server-side implementation, owing to the demands of programming, have been the weakest link in the creation of Web-based experiments. We were able to overcome these difficulties with NetCloak on a Macintosh Web server running WebSTAR.

According to Maxum, the company that produces NetCloak,

NetCloak is a plug-in or CGI that works with all Mac-based web servers, including WebSTAR, WebTen and Apple-Share/IP. Since NetCloak is a server add-on, there are no potential browser incompatibilities. You can use NetCloak's capabilities in your pages without a lot of testing or worrying about whether it will work for all of the visitors to your site. . . . NetCloak's functions are all implemented by simple, HTML-like commands. Variables, cookies and math functions can be used to track visitors to your site. . . . NetCloak Pro adds forms processing and e-mail integration. . . . With NetCloak Pro, you can turn information from HTML forms into data files, formatted e-mail messages or even new web pages. . . . With NetCloak Pro, your intelligent error page can automatically notify you via e-mail when users encounter a bad link on your site. (Maxum, 2001)

At the time of this writing, there is a beta version of NetCloak for Mac OS X, and a commercial version is expected soon to work with both WebStar V and Apache

(Mac OS X Server). NetCloak Pro has a number of features that experimenters will find useful. The show and hide commands allow the experimenter to customize the look of Web pages—and thus the presentation of stimuli to participants—on the basis of a number of variables, including the participant's earlier responses, the date, the time, the user's browser, the IP address, cookie values, and system-generated random numbers. NetCloak's security commands can be used to protect individual pages, paragraphs, or even single words within a page. NetCloak can be used to automatically send and receive e-mail, track participants by IP address or cookie value, and perform calculations.

NetCloak Pro uses what Maxum calls forms definition markup language (FDML) files to tell NetCloak Pro what to do with the data submitted in a form. The FDML file allows the experimenter to create new HTML documents, update existing HTML documents, store text data, and send e-mail messages. We found NetCloak easy to learn and use. During a busy part of the academic year, starting with no knowledge of NetCloak, in less than a month, the first author was able to design and implement a server-side Web-based experiment and to assist three graduate students in designing and implementing their own experiments (Campbell, End, Kretschmar, & Wolfe, 2001; Kretschmar & Wolfe, 2001).

Charles was walking in the woods when he spotted a brown bear. It was exciting to watch the bear break open the hive and lick the honey with his long tongue. Suddenly, Charles felt a sharp pain and realized that he had been stung.

Please rate the probability of each of these statements about the story above using a rating scale from 0% (impossible) to 100% (completely certain).

What is the probability that Charles was stung by a scorpion? %

What is the probability that Charles was stung by a bee? %

What is the probability that Charles was stung by a wasp? %

What is the probability that Charles was stung by a bee that is a wasp? %

What is the probability that Charles was stung by either a bee or a wasp? %

Please check your answers. When you are done, please click on the button marked "I'm Done" below.

Table 1
Hidden Commands to Process a Form With FDML

```
<form METHOD="POST" ACTION="1AAN.FDML">
<input TYPE="hidden" NAME="userdomain" VALUE=<INSERT_DOMAIN>>
<input TYPE="hidden" NAME="problem" VALUE="Valerie">
<input TYPE="hidden" NAME="time" VALUE=<INSERT_TIME>>
<input TYPE="hidden" NAME="page" VALUE="1AAN.html">
```

An Experiment on Joint Probability Estimation

We used NetCloak to implement a server-side Web-based experiment on the estimation of joint probabilities, including the conjunctive and disjunctive probabilities of two events. Although the details of the experiment are beyond the scope of this paper, we will briefly describe the experiment, with a focus on methodology. The experiments were run on a dedicated Macintosh server running WebSTAR, which can be found on the Web at <http://tappan.wcp.muohio.edu/>. This research examines the way people estimate the likelihood of events. Participants were given a series of 12 vignettes, two to three sentences long, and were asked to rate the probability of some events on the basis of those vignettes—for example, "What is the probability that Charles was stung by a bee?" The participants were asked to enter their probability estimates expressed as a percentage between 0 and 100, in a box beside each item. Figure 1 presents a sample problem.

Each problem was presented on a separate Web page with hidden commands such as those seen in Table 1. The first line tells the server that the data is to be processed by a FDML file we called 1AAN.FDML. The other lines create hidden variables for the user domain (IP address), the problem name, the time, and the page. The data were passed to the FDML file we created called 1AAN.FDML. The FDML can be seen in Table 2.

The first line tells the server to present the next Web page that we called 2AAN.html after the page has been processed. The second line verifies that each box has been filled in by the participant. If one or more boxes are left blank, the participant receives a Web page we called missinginfo.html and is then sent back to the previous page. The insertfile command opens a text file we called data.txt and inserts data following a string of characters that we specified (<!--previousdata-->). Finally, the replace commands insert data into the text file. Problem and page were attached to the Web page by the experimenters, user domain and time were taken from the server software, and the participant's responses were captured by variables we called distracter, PA, PB, PAANDB, and PAORB.

The participants were randomly assigned to one of eight experimental conditions representing a $2 \times 2 \times 2$

factorial design with two counterbalanced randomly selected problem orders, the presence or absence of a 2×2 table, and the presence or absence of an analogy. This was accomplished with the FDML file shown in Table 3.

For each participant, an integer between 1 and 1,000 was generated by a pseudo-random number generator built into NetCloak. If the number, Random1000, was greater than or equal to 875, the participants were assigned to one condition and were sent to the Web page AAT.html. If the number was less than 875 and greater than or equal to 750, they were assigned to a second condition and were sent to the Web page ANT.html, and so forth.

Using the Time Function

Using NetCloak's time function (see Table 1), we measured time on task. The participants who completed all of the problems took a mean of 17:05 to complete the entire experiment ($SD = 12:42$), with a range of 4:43 to 1:44:09 (the second-longest time on task was 51:17). A problem with Web-based experiments is that they do not permit direct observations of the participants' behavior. However, we were able to measure time on each problem. The participant who took 1:44:09 reported that she was a 27-year-old white college-educated female from the U.S. South, with English as her first language. This participant took 46:03 on Problem 4 and more time than average on all of the other problems.

We found NetCloak's time function quite useful in helping us assess the quality of our data. However, we make no claim of millisecond accuracy for NetCloak. The mean time on task was consistent with our expectations and our experience with off-line versions of the task. Yet some participants took a good deal longer than the task required. It may be prudent for experimenters to set *a priori* criteria for time on task, such as excluding data from participants who take more than two or three standard deviations from the mean time on task.

Conclusion

The Web has tremendous potential as a vehicle for conducting psychology experiments. However, the lack of experimenter control over experimental conditions remains

Table 2
Sample FDML File to Process Data From a Form

```
<RESPONSE>"2AAN.html"</RESPONSE>
<VERIFY distracter PA PB PAANDB PAORB> "missinginfo.html"</VERIFY>
<INSERTFILE "<!--previousdata-->"
"data.txt"
</INSERTFILE>      <REPLACE problem>      <REPLACE page>      <REPLACE userdomain>
<REPLACE distracter> <REPLACE PA>
<REPLACE PB>      <REPLACE PAANDB>      <REPLACE PAORB>
```

Table 3
FDML File to Randomly Assign Participants to
One of Eight Experimental Conditions

```
<IF random1000 GT= 875 THEN "<RETURN_PAGE>"
"AAT.html"></RETURN_PAGE>">
<IF random1000 LT 875 AND random1000 GT= 750 THEN
"<RETURN_PAGE>" "ANT.html"></RETURN_PAGE>">
<IF random1000 LT 750 AND random1000 GT= 625 THEN
"<RETURN_PAGE>" "AAN.html"></RETURN_PAGE>">
<IF random1000 LT 625 AND random1000 GT= 500 THEN
"<RETURN_PAGE>" "IANN.html"></RETURN_PAGE>">
<IF random1000 LT 500 AND random1000 GT= 375 THEN
"<RETURN_PAGE>" "BAT.html"></RETURN_PAGE>">
<IF random1000 LT 375 AND random1000 GT= 250 THEN
"<RETURN_PAGE>" "BNT.html"></RETURN_PAGE>">
<IF random1000 LT 250 AND random1000 GT= 125 THEN
"<RETURN_PAGE>" "BAN.html"></RETURN_PAGE>">
<IF random1000 LT 125 THEN "<RETURN_PAGE>"
"IBNN.html"></RETURN_PAGE>">
```

a serious issue. Server-side implementations eliminate some of the problems associated with heterogeneous computing environments, and NetCloak Pro is a powerful and easy-to-use tool for creating and managing server-side Web-based experiments. We found NetCloak Pro easy to learn, and our experience suggests that it would be useful to other experimenters.

REFERENCES

- BAILEY, R. D., FOOTE, W. E., & THROCKMORTON, B. (2000). Human sexual behavior: A comparison of college and Internet surveys. In M. H. Birnbaum (Ed.), *Psychological experiments on the Internet* (pp. 141-168). San Diego: Academic Press.
- BARON, J., & SIEPMANN, M. (2000). Techniques for creating and using Web questionnaires in research and teaching. In M. H. Birnbaum (Ed.), *Psychological experiments on the Internet* (pp. 235-265). San Diego: Academic Press.
- BIRNBAUM, M. H. (1999). Testing critical properties of decision making on the Internet. *Psychological Science*, *10*, 399-407.
- BUCHANAN, T., & SMITH, J. L. (1999). Using the Internet for psychological research: Personality testing on the World Wide Web. *British Journal of Psychology*, *90*, 125-144.
- CAMPBELL, J., END, C., KRETSCHMAR, J., & WOLFE, C. R. (2001, November). *Electronic scapegoating: Attributions of blame and responsibility in human computer interaction*. Poster presented at the 31st Annual Meeting of the Society for Computers in Psychology, Orlando, FL.
- KRANTZ, J. H., BALLARD, J., & SCHER, J. (1997). Comparing the results of laboratory and World-Wide Web samples on the determinants of female attractiveness. *Behavior Research Methods, Instruments, & Computers*, *29*, 264-269.
- KRETSCHMAR, J., & WOLFE, C. R. (2001, November). *How the presence and appropriateness of gender-related sport links affect impressions of WWW home page owners*. Poster presented at the 31st Annual Meeting of the Society for Computers in Psychology, Orlando, FL.
- MAXUM INC. (2001). [NetCloak dynamic, interactive Web sites]. Available on-line 5/30/01 at <http://www.maxum.com/NetCloak/>.
- MORROW, R. H., & MCKEE, A. J. (1998). CGI scripts: A strategy for between-subjects experimental group assignment on the World-Wide Web. *Behavior Research Methods, Instruments, & Computers*, *30*, 306-308.
- SCHMIDT, W. C. (1997). World-Wide Web survey research made easy with WWW Survey Assistant. *Behavior Research Methods, Instruments, & Computers*, *29*, 303-304.
- SCHMIDT, W. C. (2000). The server side of psychology Web experiments. In M. H. Birnbaum (Ed.), *Psychological experiments on the Internet* (pp. 285-310). San Diego: Academic Press.
- WILLIAMS, J. E., MCGRAW, K. O., & TEW, M. D. (1999). Undergraduate labs and computers: The case for PsychExps. *Behavior Research Methods, Instruments, & Computers*, *31*, 287-291.
- WOLFE, C. R. (1992). Using Authorware Professional for developing courseware. *Behavioral Research Methods, Instruments, & Computers*, *24*, 273-276.

Manuscript received November 1, 2001;
 revision accepted for publication March 10, 2002.)