Behavior Research Methods
2005, 37 (2), 293-300

WebDiP: A tool for information search experiments on the World-Wide Web

MICHAEL SCHULTE-MECKLENBECK
Columbia Business School, New York, New York

and

MORITZ NEUN
University of Zürich, Zürich, Switzerland

We describe WebDiP (Web Decision Processes)—an open-source, online tool—which enables a researcher to track participants while they search for information in a database, available through the Internet. After various instructions on setup and configuration are given, a detailed view of WebDiP explains the system's technical features. Furthermore, other open-source tools are mentioned that helped in programming WebDiP, running it, or analyzing data gathered with it. We present new approaches of how open-source thinking can be incorporated into a research process and discuss future perspectives of WebDiP.

With the rise of the World-Wide Web, the ease of accessing information and the corresponding search for it has grown quickly (see, e.g., http://www.searchenginewatch.com/). Search engines are the gatekeepers to information on the Internet, providing fast and easy access to every imaginable topic. The penetration of search engines into everyday life has even reached the language, wherein the process of searching for information on the Internet is termed "Googling" nowadays. This article introduces a research tool that is based on the ideas of a search engine but adds capabilities for conducting decision making experiments on information search in a controlled setting. The roots of the idea to construct such a program are twofold: They stem from the broad usage of search engines, as well as from various process-tracing approaches introduced in decision making research. We will outline two of these approaches briefly.

Huber, Wider, and Huber (1997) introduced the method of active information search (AIS), which turns upside down the process of how a decision task is traditionally done. It is common in decision making research to confront a participant with a task text and collect choices or judgments about this text as the behavioral measures. In an AIS experiment, the participant is given a more active position because he or she has to ask questions in order to gather information about the given problem. Through this procedure, it is possible to record what information is needed in a task and the order in which it is selected.

A second approach that influenced our ideas was introduced by Payne, Bettman, and Johnson (1993) with the adaptive decision maker framework named Mouselab. In the Mouselab system, the participant is confronted with a matrix of several alternatives and attributes. Through movement of the computer mouse over an information box, the otherwise hidden content is shown. Within this system, one can record detailed information about the sequence of information gathered, the time spent on an information item, and the total time needed for a decision. The latest development of the Mouselab family is MouselabWEB (Willemsen & Johnson, 2004), which enables the researcher to design and run a decision making experiment completely on the Internet.

From a more general perspective, WebDiP—Web Decision Processes—continues the tradition of a multitude of experimental tools using the Internet (e.g., Express by Yule & Cooper, 2003, or WEXTOR by Reips & Neuhaus, 2002) while adding features like integrated tools for data analysis. For recruiting participants, WebDiP can make use of available Internet sites that collect experiments (e.g., Reips & Lengler, 2005, or the "Psychological Research on the Net" Web site of John H. Krantz, psych.hanover.edu/research/exponnet.html).

Summing up, the starting points for the WebDiP project were the active search for information to solve a task (like that introduced in the AIS framework) as one cornerstone and the aim to automatically gather detailed information about the search process (used in the Mouselab system) as the other. The aims of the program were the following: (1) a system that lets various researchers run decision making experiments with one WebDiP installation; (2) Web-based set-up and simple administration of a research experiment; (3) export of data as well as whole decision making experiments or tasks; and

This article is based on a presentation given in the symposium Tools for Internet-Based Research at the 34th annual meeting of the Society for Computers in Psychology, Minneapolis, November 18, 2004. Correspondence should be addressed to M. Schulte-Mecklenbeck, Columbia University, Business School, 3022 Broadway, Uris Hall 5M, New York, NY 10027 (e-mail: research@schulte-mecklenbeck.com).

(4) an easy-to-use interface that lets participants concentrate on the task and not on handling issues.

On the basis of these aims, WebDiP was developed, tested, and released on sourceforge.net early in 2004. All this was done under the GNU General Public License (GPL), which is intended to provide freedom to share and change software to ensure its complete availability to all interested users. What does it mean for a research process if the tools with which a researcher is working are freely available? Given WebDiP as an example, this means that the whole program or parts of the code may be modified and also used in other software products if they themselves are again released under the GPL. Further advantages arise—anybody who has an interest in the suggested method of information search can access the sourceforge.net Web site, download the program, install it on a computer, and test whether the program fits his or her new research ideas. Most often, flaws appear when a new research idea is used for the first time. In this case, open-source software guarantees that the researcher who is interested in additions or changes can start where the last developer stopped. Progress in research becomes considerably faster with such an approach.

Technical Requirements

In this section, we will highlight the technical basics needed for setting up WebDiP. In accordance with the GPL, WebDiP is completely based on free technology. The choice of the programming language (PHP), the Web server (Apache), and the database (MySQL; available at http://www.mysql.com) were driven by this premise. Another important factor influencing the choice was the simplicity and clear structure of the technology.

PHP (available at http://www.php.net) was invented for creating dynamic Web pages. PHP is freely available, and many shared Web-hosting servers include support for it. The language borrows many functionalities and syntax from PERL as well as from C. Since it is relatively easy to learn, changes in the system can also be performed by nonprogrammers. On the support side, a large PHP developer community and many scripts are accessible on the Internet (e.g., http://www.hotscripts.com/PHP). The PHP interpreter can be used for a broad range of operating systems. Connectivity with many commercial or free databases is already included. PHP offers not only the creation of HTML but also, for example, of JPEG and PNG images, XML, ZIP, or PDF. Choosing PHP as the programming language allows one to use a broad range of different Web servers between which a migration of the system is possible. On the server side, we chose today's most frequently used Web server—Apache (available at http://www.apache.org)—which is very stable, reliable, and available for all important operating systems and platforms (WebDiP also runs on less popular Web servers such as Microsoft IIS, Netscape Enterprise, or WebStar).

In order to keep WebDiP as broadly usable as possible, a database abstraction layer can translate the functions of the program; thus, the PHP code can be used for different databases. Many shared Web-hosting servers offer the possibility of PHP and MySQL for creating dynamic Web pages. WebDiP was developed and tested on MySQL, but it will, because of the aforementioned database abstraction layer, also work with other databases (e.g., mSQL, PostgreSQL, ODBC, ODBC Adabas, Sybase and Interbase). We recommend phpMyAdmin (available at http://www.phpmyadmin.net), another open-source tool that offers a browser-based GUI (graphical user interface) for MySQL to facilitate database manipulations without syntax knowledge of MySQL.

Features

Next, we will introduce three features of WebDiP that were central in the development process. Security questions are discussed first; then a detailed description of the search capabilities is given. Finally, the built-in language support for multiple languages is described.

Security features. Reips (2002) discusses certain preconditions that should be met when an experiment is run on the Internet. Examples are the avoidance of unprotected directories, revealing of experimental structure through the URLs, or—on a more general level—the avoidance of multiple submissions. Because WebDiP is based on PHP, no access to the structure of an actually delivered Web page is possible. PHP generates requested Web pages on the server and delivers pure HTML to the participant. Therefore, no unprotected directories are possible within the system (assuming a regular server configuration). The URLs within WebDiP are generated with a session ID, which is a string of 32 characters and numbers that is generated by a random number and the current time, using the MD5 (Rivest, 1992) algorithm (a cryptography algorithm that guarantees that a session ID cannot be guessed). This technology makes it possible to do without other control mechanisms—for example, cookies, which often bring problems, because of local disabling in the participant's browser. In addition, session management lets the participant access an experiment only once, does not allow use of the "back button" to change an input, and increases security because stored links cannot be used again. The session ID does not reveal any information about the participant's current experimental condition.

The issue of multiple submissions in an online experiment is often raised (e.g., Birnbaum, 2000). In WebDiP, two mechanisms prevent multiple submissions. Participants are usually recruited by sending a link to a list (newsgroup, panel) of potential participants. This link redirects the participant to a registration page where he or she has to register with his or her e-mail address and demographic data. The provided e-mail address is registered in order to avoid multiple entries by the same person in one experiment. After registration, the participant receives an e-mail with a link containing his or her unique session ID, which he or she can use for a one-time access to the experiment. In addition to the e-mail registration,

the participant's IP address is logged. It is therefore easy to check whether two different e-mail addresses have been used from a single computer—a possible criterion for disqualification from an experiment. The mechanism of registration first and access through a second e-mail is widely used today—for example, in forums, newsletters or online communities. A researcher can identify double submissions through the session ID and IP address, but the mechanism per se helps to discourage participants from accessing an experiment twice, because of the extra effort (in comparison to an experiment where no registration is necessary).

Advanced search algorithm. Schulte-Mecklenbeck and Huber (2003) demonstrated that the use of a relatively small database with simple search mechanisms gives suboptimal results in participants' searches. As a consequence, a large number of dropouts was found in the Web-based group. One aim of WebDiP was to improve the search process, in order to keep the participants' attention and to receive a larger number of finished experiments. Two mechanisms improving the search process considerably are SOUNDEX and stemming, which are introduced now.

Participants enter searchwords with a variety of spellings. To account for this, SOUNDEX keys are used (Knuth, 1973). They have the characteristic that words with a similar pronunciation produce the same key and are used to simplify searches in databases where the pronunciation of a word is similar, but not the spelling. The SOUNDEX function returns a string that is four characters long, starting with a letter. The first letter corresponds to the first letter of the initial word. The following three digits are generated through a coding scheme in which the vowels are ignored and groups are built for the consonants. An example for a SOUNDEX coding is: HOLMES = H-452 (see Knuth, 1973, for details). Using such a coding speeds up the search process considerably and "corrects" participants' spelling errors.

Stemming searches for the heading variant form of words that share a common meaning. It then removes the more common morphological and inflexional endings from words. The algorithm most frequently used to perform such a task is the Porter-Stemmer (Porter, 1980), which is incorporated in WebDiP, too. The abovementioned inflection describes a variation of a word, typically by means of an affix, that expresses a grammatical contrast that is obligatory for the stem's word class in some grammatical context. In general, the stem or root of a word is on the left, whereas zero or more suffixes may be added on the right. If the root is modified by this process, it will normally be at its right end, whereas prefixes are added on the left. *Unhappiness* has a prefix, *un-*, and a suffix, *-ness* (the *y* of *happy* has become *i* with the addition of the suffix). Prefixes can alter the meaning radically, so usually they are left in place. But suffixes can, in certain circumstances, be removed; for example, *happy* and *happiness* have closely related meanings, and they can be reduced to their stems *happy* or *happi*. The application of this algorithm considerably increases the search speed and the number of results in information searches.

Multilanguage support. The last feature of the WebDiP system described here is the inclusion of a multilingual environment for the participants and the experimenter. Language files in English and German are included in the "language" folder within the WebDiP source tree. New languages can easily be added by simply changing the values in an existing language file and by saving the file under a new name. This can be done with any text editor. All files having a file name like "lang xx.php" (xx stands for the language code, e.g., fr or es) are automatically displayed in a dropdown menu within the WebDiP configuration and can be assigned to an experiment.

How To Set Up

In this section, we will present some simple instructions for setting up WebDiP on a server, creating an experiment/task and exporting results to a spreadsheet.

Setting up WebDiP. In order to set up WebDiP, it is necessary to have a Web server, a database, and PHP support of both. There are two possibilities for access to such a set-up. (1) Internet service providers (ISPs) most often offer packages including all of the above functionalities within the basic available packages. (2) The second option is to set up a personal server—for example, with a DSL line or at a university. All of the above applications are bundled within LAMP or XAMMP, which can be retrieved at http://www.apachefriends.org (an excellent introduction for installation of the two packages can be found at this address, too).

WebDiP was tested using the following configurations: Mandrake 9.0, Red Hat Linux 7.3, Apache 2.0, PHP 4.1.2, and MySQL 3.23.49. This does not necessarily mean that WebDiP will not work with other configurations (OSs, DBs, etc.), but installation instructions included on the WebDiP page are only provided for the above configurations. In fact, WebDiP should also work with Windows, FreeBSD, OS/2, and MacOS.

Given that one of the mentioned configurations is up and running, WebDiP has to be downloaded from http://webdip.sourceforge.net into the experimenter's Web server directory. The file is in .tar.gz format, a compression that can be expanded with, for example, 7-zip (http://www.7-zip.org) under Windows or with the "tar-xvzf" command on a Linux system. First, the database has to be built. This is done by executing the "webdip.sql" file that can be found in the "sql" subdirectory (a graphical and therefore easy-to-use environment for database management is the PhpMyAdmin program, which is installed within XAMMP). This file includes the commands to build the database structure, tables, and the other necessary default data. After the creation of a new database called, for example, webdip (in phpMyAdmin), the database can be filled with default data by executing "webdip.sql."

The file "config.inc" is located in the Web server's directory. This file contains values that have to be entered by the experimenter and are necessary for running a

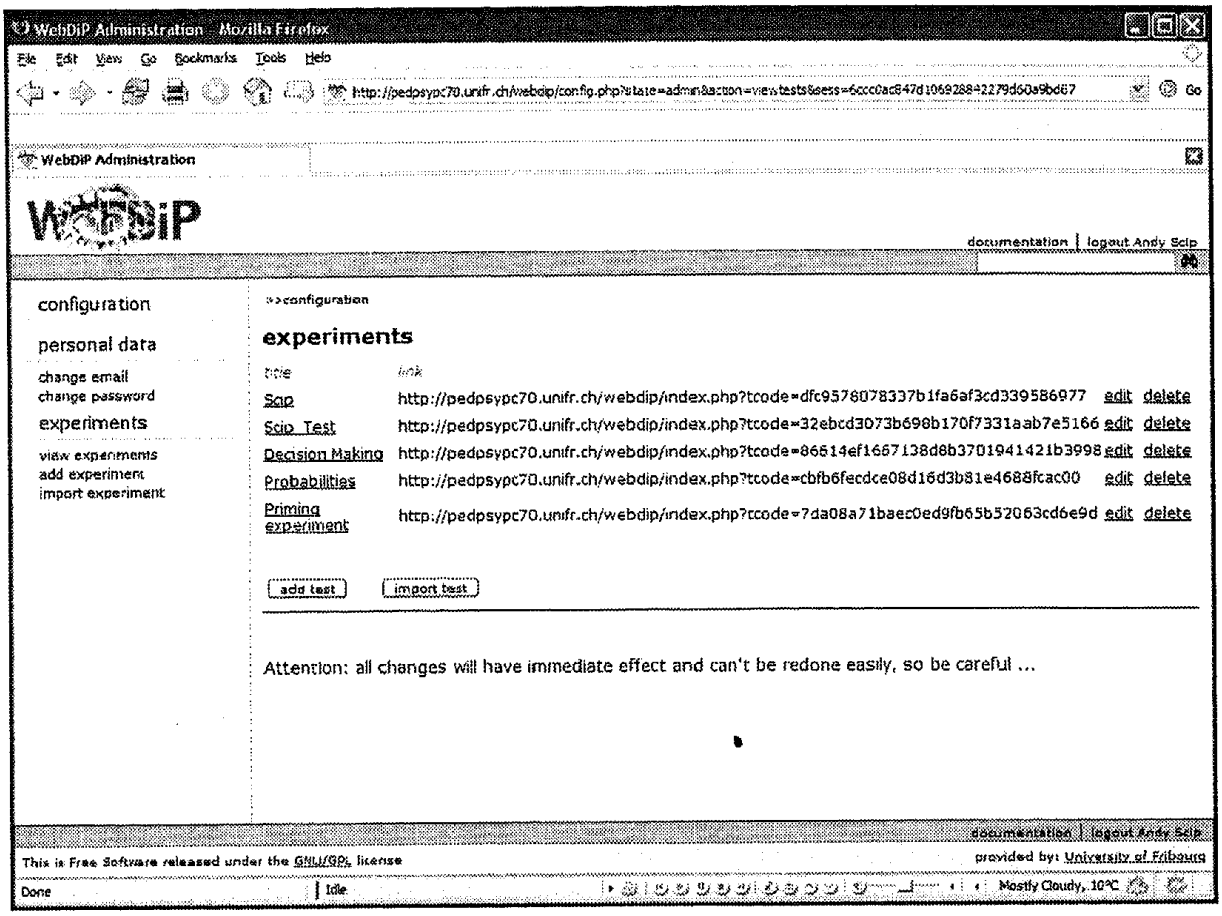

Figure 1. WebDiP's experiment options.

WebDiP installation (all of these can be changed using a simple text or HTML editor): dbhost, SQL database hostname; dbuname, SQL username; dbpass, SQL password; dbname, SQL database name; prefix, database table's prefix; and dbtype, database type. Within every installation, the initial password (root) for the superuser "root" is supplied, too. This password should be changed immediately after the installation.

Once WebDiP is running, there are several steps necessary for setting up an experiment.

Setting up experiment administrators. Every WebDiP installation has one superuser (called "root"). Only the superuser is able to create other users (researchers), who can create, edit, import, export, and delete experiments. Within one WebDiP system, many researchers can work in parallel without influencing each other. Every researcher gets his or her own username/password and can log in to a private WebDiP environment without having access to other researchers' experiments.

Setting up an experiment. Every researcher can manage multiple experiments (see Figure 1), which are displayed as a list on his or her entry page. Each experiment consists of a title, instructions, one introductory and multiple following tasks (the order of these is permuted for every participant), as well as one of four information search modes. The available modes are list (a list of all available information is presented to the participant), categories (a display of the available categories as a "filter" before the actual information is shown), full-text (a "Google-like" search with keywords), and no search (an option to generate items such as simple questionnaires). The different search types can be activated by checkboxes within the experiment's options. The title, instructions, and test language are further settings that apply to the whole experiment.

Setting up a task. Each task consists of a title, the task text, the choice text, the choices, and the corresponding information items. An arbitrary number of choices can be added, edited, or deleted. Every task can be exported and used again in another experiment. All the categories of information belonging to a task are listed in the information section. To add new information items, the researcher has access to either all information items or only one of the available categories. Within this view, the number of corresponding information items is presented as well (see Figure 2). An information item consists of a question and an answer as well as a category (the category is also used in the category search mode; see above). The SOUNDEX values and stems of the question are not displayed in the experimenter's view; they are directly written into the database and used for his or her searches.

New information items can be entered into the system using an HTML form in the browser, or they can be imported from a external source. An import function for a CSV (comma separated values) enables the researcher to prepare the information locally on a computer in an edi-

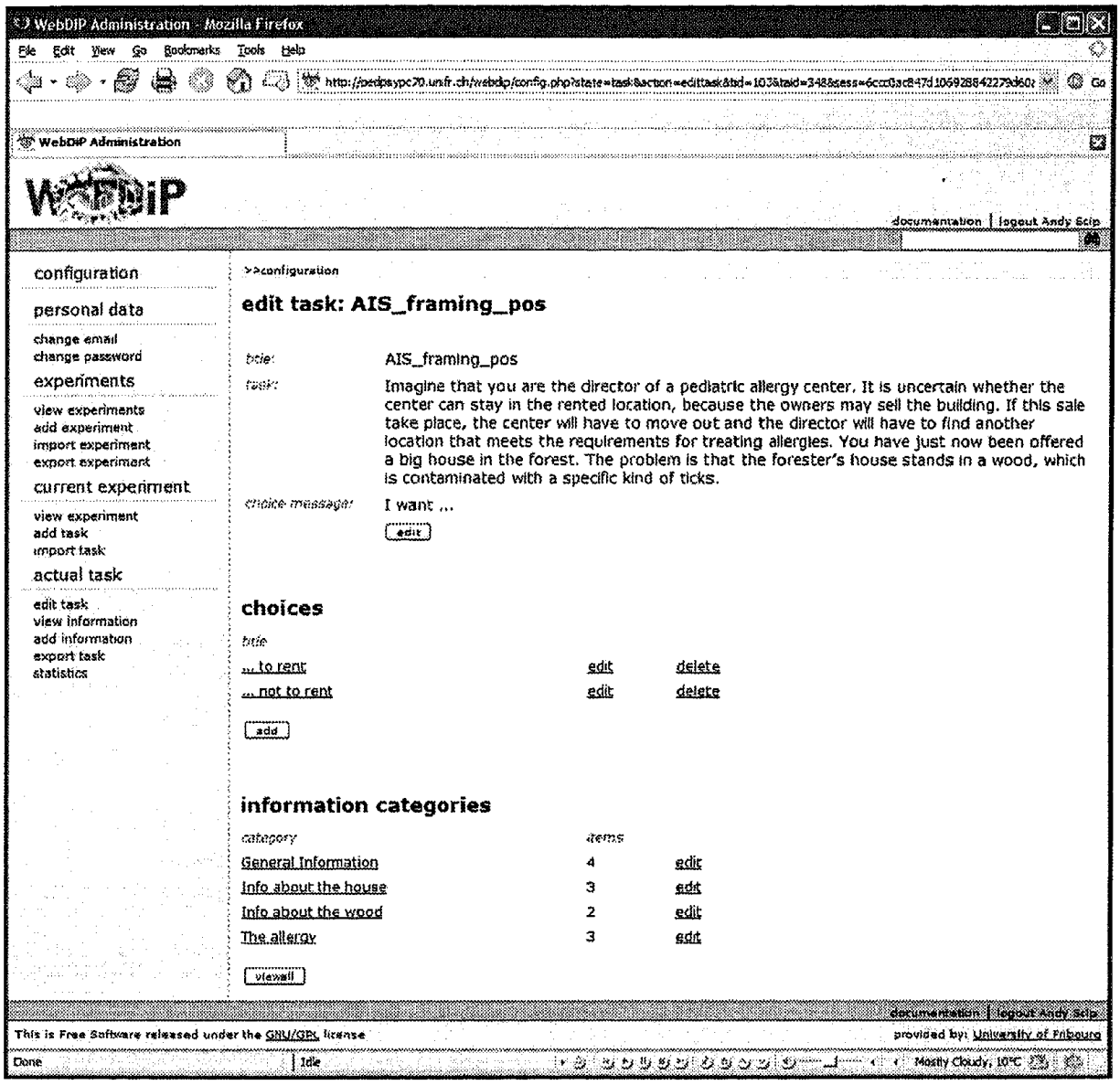

Figure 2. WebDiP's task options.

tor and share it with coworkers before uploading the values into the WebDiP system. Every line in the CSV file lines must be of the form "category;question;answer."

Import–export experiments. For easy duplication, distribution, or backup of whole experiments/decision tasks, an export function for experiments and tasks is implemented. A reusable, object-oriented, hierarchical class structure has been created, which enables the experimenter to simply download a whole experiment (or a single task) onto a local machine. The generated file can be used as backup or distributed (e.g., via e-mail) to other researchers. On a second WebDiP system, the uploaded file results in a duplicated version of the original experiment/task.

Data analysis. The following data are collected within WebDiP and can be divided into three groups: (1) The experiment-, task- and participant-ID classify each participant to a condition in an experiment. (2) The entered keyword and the actually clicked on information provide insight about the participant's search process. The choice indicates his or her decision. (3) Each event a participant makes is logged (as described above) and labeled with a timestamp, consisting of the following information: 20041022141908—that is, year, month, day, hours, minutes, and seconds. A participant can use a timestamp to measure how long a task takes.

The statistics section provides access to the collected data of an experiment for further analysis. An online statistical analysis is already displayable in the browser while an experiment is in progress. This bar chart (see Figure 3) illustrates how many information items have been accessed within the categories. For further statistical analysis, various export possibilities are available to create CSV files that can be directly imported into a spreadsheet program or SPSS.

Patterns or sequences of accessed information items are the source for determining participants' search strategies. For the analysis of patterns in the search process,

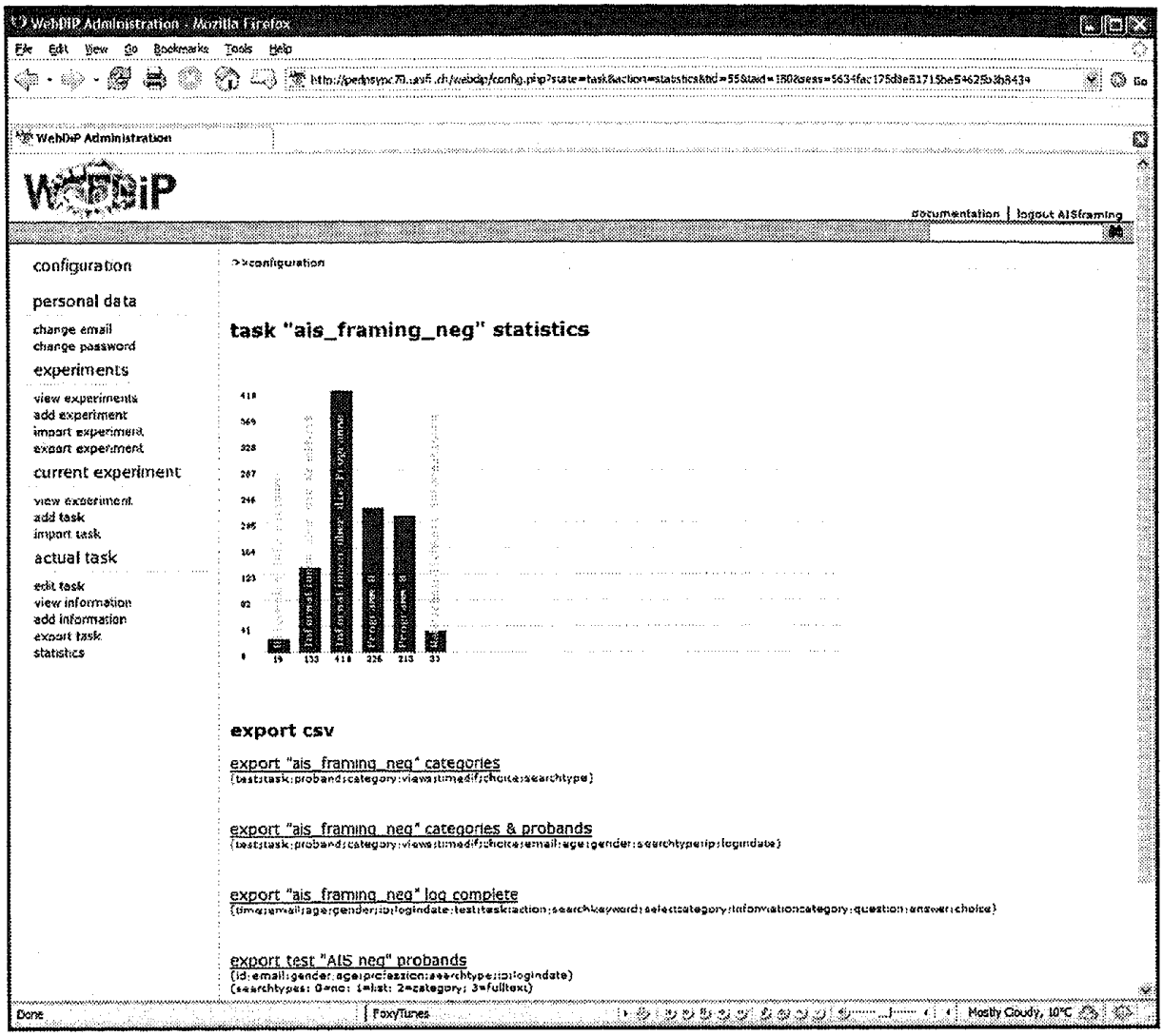

Figure 3. WebDiP's statistics view.

we recommend two algorithms: SPAM (sequential pattern mining; Ayres, Flannick, Gehrke, & Yiu, 2002) and MAFIA (mining maximal frequent itemsets; Burdick, Calimlim, & Gehrke, 2001), published as open-source projects (available at http://himalaya-tools.source forge.net/) by the Cornell Database Group at Cornell University. SPAM lets one find all frequent sequences in a data search stream (clickstream), whereas MAFIA searches for patterns that are used most frequently. Both tools are an ideal addition to more traditional (psychological) data analysis performed—for example, an analysis of variance within SPSS.

Participants' view. It was important that this project offer a common environment for the participants in order to keep instructions short and allow them to concentrate on specific tasks and not on technical explanations. Therefore, search engines' layouts were taken as an example of the WebDiP interface.

The participant enters every experiment (after the registration process; see above) by clicking on a link sent by the system. This link brings the participant directly to the instruction pages generated by the experimenter, which explain the basic components of a WebDiP experiment. When these instructions are followed, more specific instructions for the actual experiment lead to a warm-up task. After completion of the warm-up task, the participant works on the actual task (see Figure 4 for a schema of an experiment from the participant's point of view).

Future Developments and Discussion

Four aims were outlined at the beginning of this article, all of which have been reached within the current WebDiP development stage. In a test installation, 15 experiments were run independently by 15 researchers (in this case, students). No problems were reported concerning usability of the interface or data integrity between the experiments. The export of the generated data was performed smoothly either to a spreadsheet or even directly into SPSS. Several studies have already tested about 500 participants, and the only instance of technical problems concerned the relatively long URLs used within the e-mails sent to them, since some e-mail programs seem to have trouble handling URLs longer than one line.

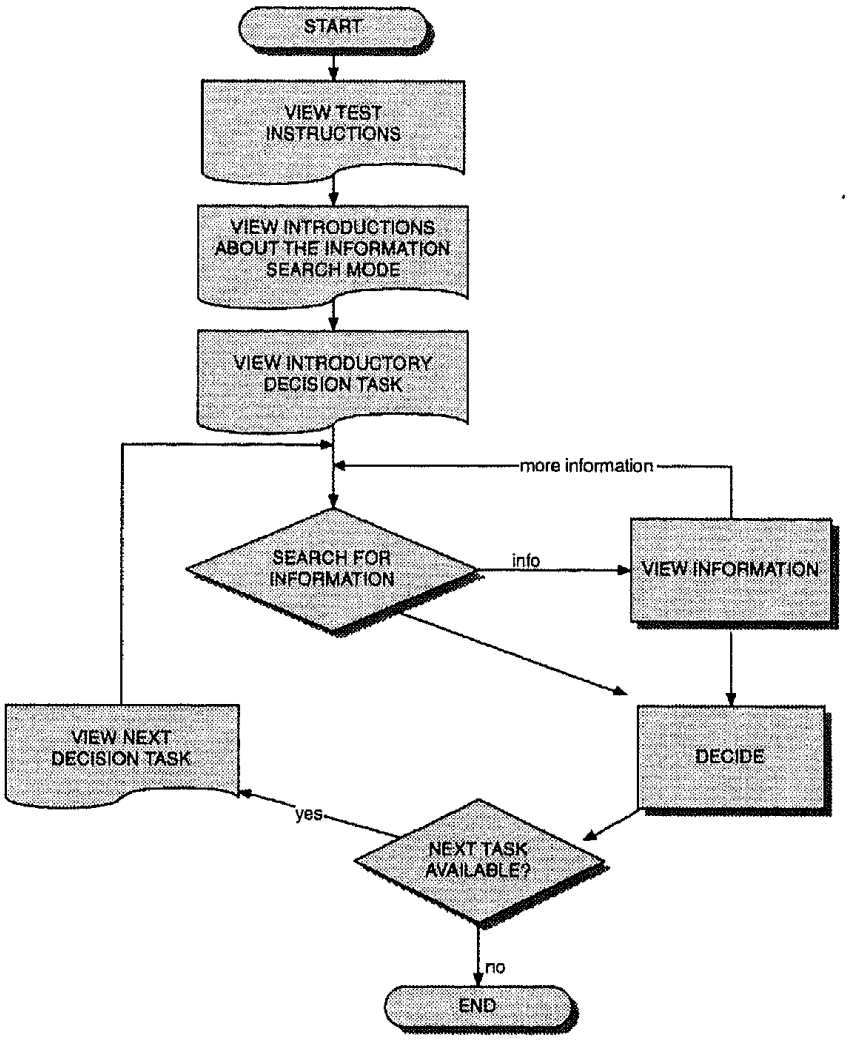

Figure 4. Structure of an experiment.

Future developments will include the following: (1) The language support of the base system will be enlarged through systematic generation of as many language versions as possible. (2) At the moment, some data manipulation is necessary to use WebDiP data with the SPAM and MAFIA algorithms, an issue that could be overcome by the generation of specific export filters. Finally, (3) the generation of an open-source experiment database is planned. Through this database, it should be possible to up- and download ready-made experiments. Such a database would speed up the research cycle considerably, because task generation, which takes the most time within the experiment preparation, could be minimized.

A central aim of this article was the introduction of WebDiP, but a second claim concerning the open-source idea is important. Using open-source tools provides up-to-date programs developed by a large number of programmers. Web technologies like the Apache server have extremely short patching times when errors or security flaws occur. The release of new products in the open-source family considerably enlarges the group of available tools. The Web site www.sourceforge.net is an excellent example of the enormous amount of free software available in the open-source community—in November 2004, over 90,000 projects were listed, among them such important ones as PHP, Apache, PHPMyAdmin, or Moodle (see Schulte-Mecklenbeck, 2004, for details). WebDiP might never be a big player on sourceforge.net, but it already provides an interesting alternative to standard paper-and-pencil decision making tasks.

REFERENCES

AYRES, J., FLANNICK, J., GEHRKE, J., & YIU, T. (2002). Sequential pattern mining using a bitmap representation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 429-435). New York: ACM.

BIRNBAUM, M. H. (2000). *Psychological experiments on the Internet*. San Diego: Academic Press.

BURDICK, D., CALIMLIM, M., & GEHRKE, J. (2001). MAFIA: A maximal frequent itemset algorithm for transactional databases. In *Proceedings of the 17th International Conference on Data Engineering* (pp. 443-452). Heidelberg: CDE.

HUBER, O., WIDER, R., & HUBER, O. W. (1997). Active information search and complete information presentation in naturalistic risky decision tasks. *Acta Psychologica*, **95**, 15-29.

KNUTH, D. (1973). *The art of computer programming: Vol. 3. Sorting and searching*. Boston: Addison-Wesley.

PAYNE, J. W., BETTMAN, J. R., & JOHNSON, E. J. (1993). *The adaptive decision maker*. New York: Cambridge University Press.

PORTER, M. (1980). An algorithm for suffix stripping. *Program*, **14**, 130-137.

REIPS, U.-D. (2002). Standards for Internet-based experimenting. *Experimental Psychology*, **49**, 243-256.

REIPS, U.-D., & LENGLER, R. (2005). The *Web Experiment List*: A Web site for the recruitment of participants and archiving of Internet-based experiments. *Behavior Research Methods*, **37**, 287-292.

REIPS, U.-D., & NEUHAUS, C. (2002). WEXTOR: A Web-based tool for generating and visualizing experimental designs and procedures. *Behavior Research Methods, Instruments, & Computers*, **34**, 234-240.

RIVEST, R. [L.] (1992, April). The MD5 message-digest algorithm. *Request for Comments*, No. 1321. Available at www.rfc-editor.org-index2.html.

SCHULTE-MECKLENBECK, M. (2004). Brave new World ... Wide Web. Blending old teaching methods with a cutting-edge virtual learning environment. *APS Observer*, **17**, 48-53.

SCHULTE-MECKLENBECK, M., & HUBER, O. (2003). Information search in the laboratory and on the Web: With or without an experimenter. *Behavior Research Methods, Instruments, & Computers*, **35**, 227-235.

WILLEMSEN, M. C., & JOHNSON, E. J. (2004, November). *MouselabWEB: Performing sophisticated process tracing experiments in the participant's home!* Paper presented at the Society for Computers in Psychology annual meeting, Minneapolis.

YULE, P., & COOPER, R. P. (2003). Express: A Web-based technology to support human and computational experimentation. *Behavior Research Methods, Instruments, & Computers*, **35**, 605-613.

(Manuscript received November 15, 2004;
revision accepted for publication March 31, 2005.)