

WEXTOR: A Web-based tool for generating and visualizing experimental designs and procedures

ULF-DIETRICH REIPS and CHRISTOPH NEUHAUS
University of Zurich, Zurich, Switzerland

WEXTOR is a Javascript-based experiment generator and teaching tool on the World-Wide Web that can be used to design laboratory and Web experiments in a guided step-by-step process. It dynamically creates the customized Web pages and Javascripts needed for the experimental procedure and provides experimenters with a print-ready visual display of their experimental design. WEXTOR flexibly supports *complete and incomplete factorial designs* with *between-subjects*, *within-subjects*, and *quasi-experimental* factors, as well as mixed designs. The software implements client-side response time measurement and contains a content wizard for creating interactive materials, as well as dependent measures (graphical scales, multiple-choice items, etc.), on the experiment pages. However, it does not aim to replace a full-fledged HTML editor. Several methodological features specifically needed in Web experimental design have been implemented in the Web-based tool and are described in this paper. WEXTOR is platform independent. The created Web pages can be uploaded to any type of Web server in which data may be recorded in logfiles or via a database. The current version of WEXTOR is freely available for educational and noncommercial purposes. Its Web address is <http://www.genpsylab.unizh.ch/wextor/index.html>.

Whereas any Web experiment¹ can also be conducted as a laboratory experiment with ease (Reips, 1997), almost no laboratory experiment can be conducted as a Web experiment without major reprogramming. So if an experimental idea lends itself to being conducted on the Web at all, it is wise to create a Web experiment and also to invite local participants to the laboratory in the traditional way. In accord with this logic, a Web-based experiment generator like the Web site described in this article may be an ideal starting point for experimenters to create experimental designs and procedures of any kind, even those that are not conducted on the Internet.

After several years of experience with Web-based experimenting (Reips, 2001), we feel that the time has come for a user-friendly on-line tool that creates true Web experiments of variable design. An earlier attempt, FactorWiz by Michael Birnbaum (2000), which is very useful for rapidly setting up certain types of Web experiments, is limited to within-subjects designs and does not create a visual display of experimental designs.

We developed a Javascript-based Web experiment generator named WEXTOR in order to widen the scope and improve the situation for potential Web experimenters without major programming and/or Internet knowledge. WEXTOR can be used to design Web experiments and Web-

technology-based conventional experiments in a guided step-by-step process. It dynamically creates the customized Web pages and Javascripts needed for a Web experiment and provides experimenters with a print-ready visual display of their experimental design in terms of directories (folders) and Web pages. These Web experiments can then be conducted in an on-line laboratory, such as the Web Experimental Psychology Lab² (see Reips, 2001). WEXTOR is platform independent, and its current version is freely available for educational and noncommercial purposes. The publicly available version (currently 1.3) has the Web address <http://www.genpsylab.unizh.ch/wextor/index.html>.

Features and Limitations

WEXTOR has a number of features, some of which have been shown to be of special value in Web experimenting. With the aim of developing a Web-based tool that helps experimenters to avoid the need for deep-level programming, we implemented a Web site that dynamically creates customized Javascripts. In other words, all the basic HTML (Web pages) and Javascript code needed for the experimental design and procedure is created automatically, depending on the user's input, on forms accessible via a regular Javascript-compatible Web browser.

WEXTOR is not designed to compete with full-fledged HTML editors, nor does it host the completed Web pages or record the responses (yet). Despite these limitations, we hope WEXTOR will be useful to the following types of users: (1) students, who are often familiar with basic Internet techniques and want to learn about experimenting;

All correspondence concerning this article should be addressed to U.-D. Reips, Allgemeine und Entwicklungspsychologie, Universität Zürich, Attenhoferstr. 9, CH-8032 Zürich, Switzerland (e-mail: ureips@genpsy.unizh.ch).

(2) experienced experimenters (faculty), who know much about experimental design and want to learn about the specifics of Web-technology-based experimenting and a little about Internet techniques; (3) experienced Web experimenters who need a fast tool to create *experiment skeletons* (procedures and designs) that can then be formed and modeled with their preferred Internet tools. As a platform-independent system, WEXTOR does not place any limitations on experimenters as to what operating system, Web server, data collection procedures, or proprietary software they may use to run their experiments. However, the user needs a basic understanding of HTML principles to know how to handle the Web pages generated in the interaction with WEXTOR (and to realize what WEXTOR can do). Inserting additional text, graphical stimuli, or further interactive elements into the code all become fairly routine procedures after an afternoon of instruction in the generation of Web pages. This task may be performed by using software with a complexity anywhere between a simple text editor and a full-fledged HTML editor, depending on the size of the design and the degree of graphical sophistication desired. For reasons of display, usability, and suitability as a teaching tool, the number of factors is currently limited to five between-subjects factor and one within-subjects factor with four levels (or two within-subjects factors with two levels each). The number of quasi-experimental (natural) factors is unlimited, but high numbers are discouraged by the way in which the form for this type of factor is designed. In principle, the size of the design is limited only by the memory allocated to the user's Web browser; in our own tests, we did not go beyond designs with several thousand experimental conditions.

Currently (in Version 1.3), WEXTOR implements the following criteria that are important in designing conventional experiments (e.g., Kirk, 1995; Martin, 2000) and Web experiments (Reips, 2000, in press): (1) complete and incomplete designs; (2) experimental (between-subjects and within-subjects) factors; (3) quasi-experimental (natural) factors; (4) automatic avoidance of page number confounding; (5) Javascript test redirect functionality to minimize dropout; (6) randomized distribution of participants to experimental conditions; (7) distribution of participants to experimental conditions on the basis of levels of quasi-experimental factors; (8) random or fixed order of blocks; (9) automatic avoidance of frequently made configuration errors when naming experimental conditions in Web experimenting (Reips, in press); (10) randomly generated continuous user IDs for enhanced multiple-submission control; (11) optional client-side response time measurement; (12) implementation of the warm-up technique for dropout control (Reips, 1997, 2000); (13) interactive dependent measures and materials (created via content wizard); (14) generation of ready-to-copy HTML and Javascript code; (15) visual display and listing of experimental design in terms of directories (folders), Web pages, and the desired form of distribution of participants to experimental conditions.

WEXTOR is quite a fast way of building an experimental design and the associated procedures; our tests have

shown that simple designs (2×2 with less than 10 Web pages in each condition, and similar) can be built in less than 5 min. However, the Web pages generated must be customized with additional materials, such as instruction text, graphical stimuli, or interactive procedures, and this may take anywhere from a few minutes to several hours, depending on the size and degree of interactivity of the design.

WEXTOR as a Teaching Tool: Didactical Value

WEXTOR is designed to fulfill several functions. As a Web-based tool, it naturally helps to reduce the workload and potential errors when (1) designing and (2) constructing conventional and Web experiments. A third important function of WEXTOR is its instructional value: WEXTOR helps in teaching how to design conventional and Web experiments.

Several criteria show how WEXTOR supports learning about experimental design. First, experiment design is implemented as a guided step-by-step process. The learner is taken through a chain of decisions to be made. Each decision introduces a new concept. For example, the concept of quasi-experimental (natural) factors, such as the participant's age, is introduced in Steps 4 and 8. Because users need to decide whether the experiment includes such a factor, they need to learn about the meaning of this concept and may do so by clicking on the "view help" link provided. The small pop-up window that opens contains a definition. WEXTOR's final version will include instructive materials linked to all important terms. All steps are named by key phrases, such as "defining factors," "naming levels," and "defining quasi-experimental factors," and most steps begin with a question inviting the experimenter to define important aspects of the design. Through this process, WEXTOR helps students to become familiar with the basic concepts of experimental methodology and to structure their ideas relating to the design.

Second, particular effort was devoted to visualizing the experimental designs created by the software tool. WEXTOR visualizes experimental designs in print-ready form, and the visual displays can be saved for later use. Also, the Web site itself was created to comply with the Bauhaus dictum "form follows function" (Sullivan, 1896/1947). Navigation is intuitively supported by a design that is visually appealing without distracting the user.

Third, WEXTOR reacts to the user's input. Its dynamic structure picks up the user's choices and written text and subsequently structures itself accordingly. WEXTOR also shows the user dialog windows containing feedback whenever a wrong decision is made—for example, if the user attempts to give the same name to two experimental conditions.

There are also other ways in which WEXTOR teaches experimental design intuitively. It lets users create design variants interactively by using "back" and "refresh" buttons and pop-up menus, thus supporting their design skills by means of trial and error and comparison.

A way of using WEXTOR in class is, for example, to demonstrate the construction of a well-known psycholog-

ical experiment by going through each of WEXTOR's steps in front of the class. Along the way, there are many opportunities to explain important concepts such as *factor*, *level*, *page number confounding*, or *within-subjects*. WEXTOR's flexibility and usability makes it easy to show what impact a decision will have on the subsequent steps. For example, students may see how just one additional level in a factor can significantly increase the number of experimental conditions in complex designs. Students can be asked to complete a logical order of tasks—for example, (1) create the exact same experiment as that demonstrated by the teacher, (2) create a minor variation of the experiment, (3) create their own experimental design and materials with WEXTOR. Students can be asked to submit screenshots and resulting HTML code as part of their class work.

WEXTOR is currently used for teaching in our department at the University of Zurich on a preliminary basis and for evaluation purposes. It will be built into our "Experimental design and Web-based experimentation" module in the Swiss Virtual Campus project entitled "Methodological Education for the Social Sciences."

Ten Steps: An Example of Designing a Web Experiment

WEXTOR offers a 10-step route to experimental design and the associated procedures. What happens during each of these steps will be described in the following section. As an example, we will show how a bilingual Web experiment by Frick, Bächtiger, and Reips (2001) on the influence of incentive information, placement of personal questions, and social desirability on dropout rates and reported weekly television viewing times could be implemented with WEXTOR.

In Step 1, the experimenter is asked to name the experiment; let us name it "Dropout."

Step 2, "Creating a Javascript test page," is important for research based on Javascript technology. WEXTOR creates Javascript-driven Web pages. So it must first be determined whether a participant's Web browser is compatible with Javascript in order to minimize dropout caused by technical problems (see Buchanan & Reips, 2001, for demographical differences between Javascript users and those whose Javascript is deactivated, and Schwarz & Reips, 2001, for Javascript-dependent dropout rates). If the test result is negative, the user will remain on the Javascript test page; otherwise, he will be forwarded to the true start page, which is named in Step 3. In practice, Steps 1–3 consist of typing a word and clicking with the mouse twice—a matter of seconds for experienced users.

Step 4 involves defining the number of factors contained in the Web experiment. Figure 1 shows how factor types, factor numbers, factor names, and level numbers are chosen for each between-subjects, within-subjects, and quasi-experimental factor. Note that WEXTOR responds dynamically to the user's choices by arranging the following steps accordingly and by integrating chosen names into the generated texts. As has already been mentioned, an ex-

perimental design created by WEXTOR is currently limited to five between-subjects factors and two within-subjects factors. Quasi-experimental factors can be defined as requiring different Web pages for different levels. For example, the experiment designed by Frick et al. (2001) was conducted in English and German, so the quasi-experimental factor of *language* requires Web pages written in English and Web pages written in German. Figure 1 shows that short names and level numbers for all four factors—namely, placement of incentive information, placement of personal information, order of the questions for television viewing and charity work, and language were put into the four text fields that appeared as soon as the question on the presence of the factors was answered. Additional procedures for quasi-experimental factors may be defined in Step 8.

Step 5 is dedicated to naming the levels of the factors defined in Step 4. Accordingly, WEXTOR asks us to provide it with names for the two levels for the factor "incentive-1" (dynamically including what was typed in Step 4), and we happily reply by writing "beginning" and "end." We use the same names for the levels of "position PI," the names "TV->CO" and "CO->TV" for the factor "order TV/CO," and "English" and "German" for the factor of "language."

The experimental conditions are named in Step 6. The chosen factors and levels are first listed, and it is then explained, with an example created dynamically from this list, how the experimental conditions arise from the combination of factor levels. WEXTOR proposes a name for the experimental conditions; these names will then be used for the folders containing all the Web pages for this condition. If the desired design is incomplete, the superfluous experimental conditions may be erased by unchecking boxes. Because Frick et al. (2001) implemented a complete design, we leave all the boxes checked. Figure 2 shows the display in Step 6 of our example.

Step 7 involves defining Web pages. The user must first type in the number of Web pages for each between-subjects condition, "5" in our case. The number of Web pages must be the same in every experimental condition; otherwise, a confounding variable would be introduced. Once the number of Web pages is entered in the textbox provided and the experimenter clicks on the button "Define pages," textboxes will be created for entering the name of each Web page. If the user attempts to use the same name for two pages in this step, an error message will appear. Step 7 also contains an option for measuring and recording the time each Web page is shown. This feature can be used as a (rough) measure of response times.

Quasi-experimental or natural factors are defined in Step 8. WEXTOR offers the opportunity to distribute participants among experimental conditions, depending on their attributes in terms of quasi-experimental factors (e.g., *English* or *German* for *language*, as in the experiment by Frick et al., 2001). Experimenters may choose between answer formats (radio buttons or pop-up menu). By simply checkmarking experimental conditions, they then tell WEXTOR the desired rules for distribution among these

Address: <http://www.gerpyslab.unizh.ch/wextor/index.html> go

[>--wextor--<] 10 steps to your Web-experimental design

Step 4. Defining factors.

How many *between-subjects factors* [[view help](#)] does your experimental design consist of? Does your experiment include at least one *within-subject factor* [[view help](#)]? Does it include at least one *quasi-experimental factor* [[view help](#)]? At this stage, please think of your experimental design as a complete design. Don't worry for now, if the experimental design you have in mind is incomplete. You will be able to erase the superfluous experimental conditions in Step 6. Additional procedures for quasi-experimental factors may be defined in Step 8. For display reasons, an experimental design created by [>--wextor--<] may consist of 1 to 5 between-subjects factors. The maximum number of levels (blocks) for within-subject factors is 4 (1x4 or 2x2). An experimental design created by [>--wextor--<] seemingly includes only one quasi-experimental factor at most. However, you may include more than one factor through direct naming of level combinations (e.g., "female & age 20-40").

Please choose the number of **between-subjects factors** your experiment consists of:

Does your experiment include at least one **within-subject factor**?

Does your experiment include at least one **quasi-experimental factor**?

If your design includes a quasi-experimental factor: if it requires different Web pages (e.g. different language versions), please tick the checkbox below.
 Levels of quasi-experimental factors require different Web pages.

All factors defined? When you have named all the factors and indicated the number of levels for each factor, then please click on the button "Next step".

Please name all the **between-subjects factors** and indicate the number of levels for each factor.

Name of factor 1: Number of levels:
 Name of factor 2: Number of levels:
 Name of factor 3: Number of levels:

Please name your **quasi-experimental factor** and indicate the number of levels (answer options). Also think of including a 'no answer' option.

Name of the factor: Number of levels:

wextor: help

Quasi-experimental design

An experimental design in which experimental conditions depend on participants' attributes on quasi-experimental (natural) factors, e.g. "female" on "sex". WEXTOR allows you to send participants to different Web pages, depending on their answers. In Step 8, you may even define sets of several target Web pages, one of which will be chosen randomly.

Related topics:
[between-subjects design](#)
[within-subject design](#)
[mixed design](#)

Figure 1. The type, number, and names of factors and the number of levels are defined in Step 4. Pop-up windows (see example on lower right) provide explanations for methodological concepts.

conditions. The distribution among experimental conditions can be either fixed or randomized between choices. Multiple quasi-experimental factors can be implemented directly by naming level combinations (e.g., "female & age 20-40").

Step 9 is used to define the number of digits for a participant ID that will be randomly generated during each run of the experiment and helps to identify data points

originating from the same participant when analyzing data from a server logfile. (This is a useful feature in Web experimenting, because dynamic IP addressing often makes it difficult to identify Internet participants.) The user will usually just click "Proceed" within a second, causing WEXTOR to generate its helpful code.

WEXTOR then generates a visual display of the experimental design, a list of the factors and levels, and a descrip-

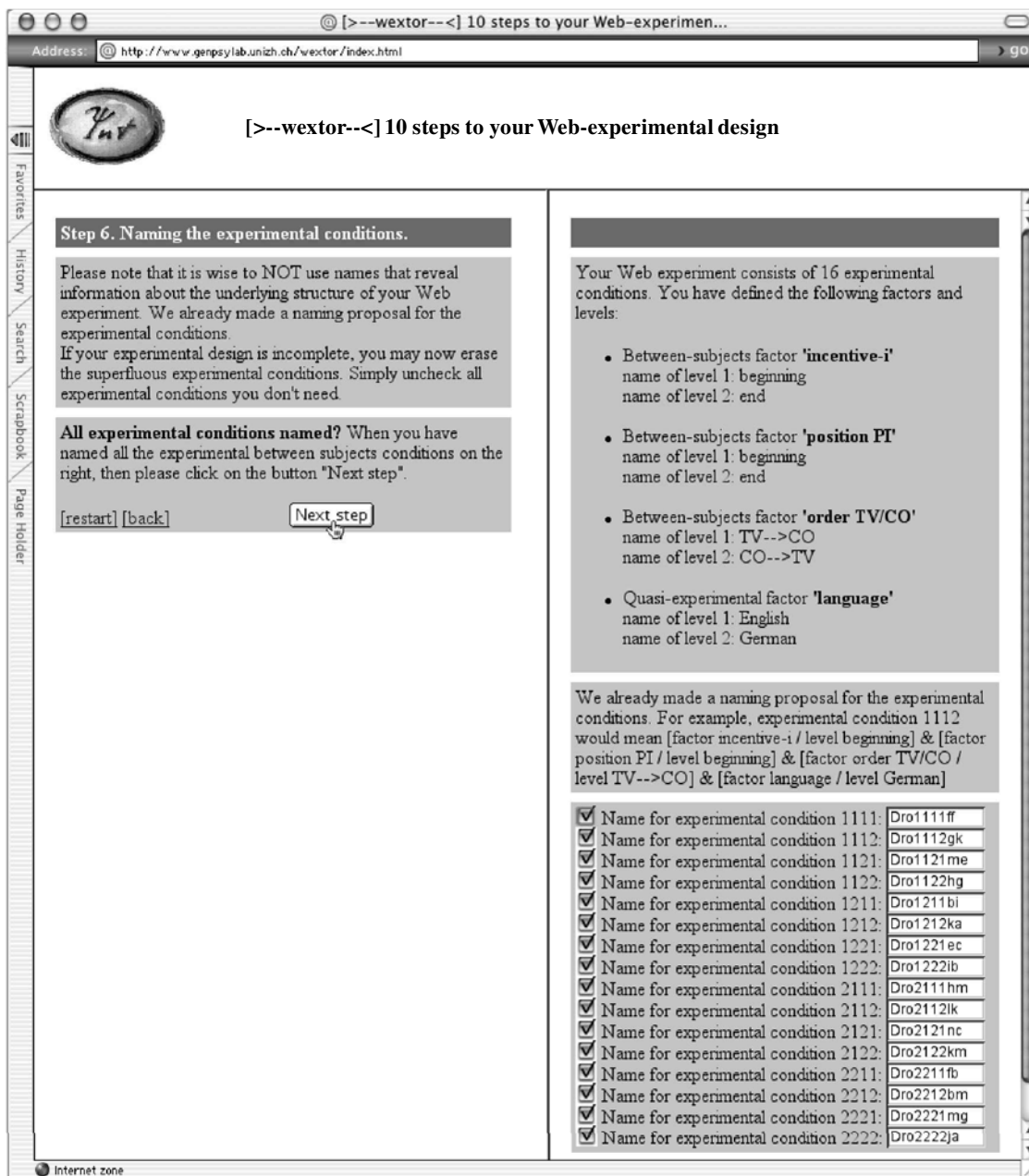


Figure 2. An example display in Step 6: Experimental conditions are named and can be chosen for complete and incomplete designs.

tion of the distribution rules implemented in Step 8 (see Figure 3 for a visual display of the example we used in this article). This information can then be printed and saved on disk. WEXTOR then generates the full HTML and Javascript code for the experiment on a second Web page. The experimenter merely needs to copy the code for each Web page into a new text file and save it with an ".htm" or ".html" extension as *text only* to the appropriate folders that were created earlier.

In Step 10, the content wizard is evoked on a separate Web page. The HTML code generated in Step 9 can be pasted into a text field. Without needing to know anything about HTML, the user can use the content wizard to create interactive materials, as well as dependent measures, on the experiment pages. Currently, the content wizard allows the implementation of text fields, horizontal and vertical radio-button scales of customized length, graphical scales, multiple-selection response items, images and other

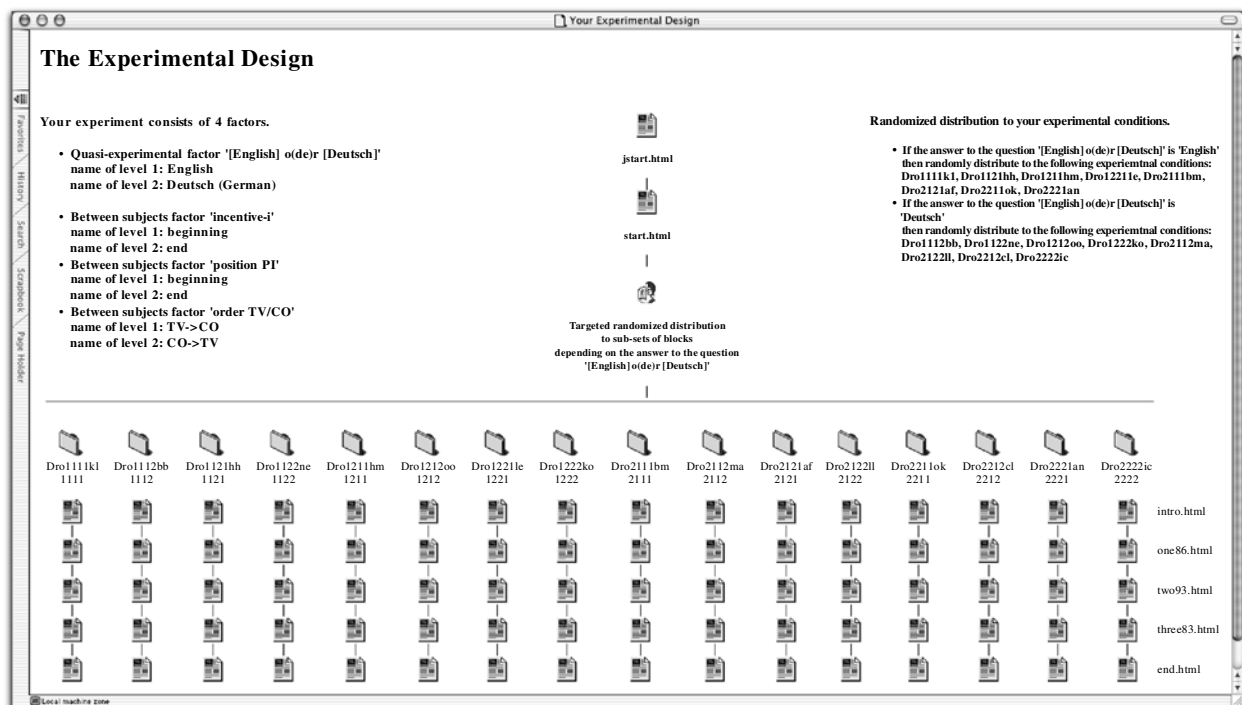


Figure 3. A visual display of an experimental design created by WEXTOR.

media content, and any combination thereof. Depending on their ambition and experimental idea, experienced users will then load the whole experiment Web site into a WYSIWYG (What you see is what you get) HTML generator, such as Adobe Golive or Macromedia Dreamweaver, for further refinement of the graphical layout.

Version History

Since October 2000, WEXTOR has been freely available on the World-Wide Web as *betaware*. As of this writing, WEXTOR is in its fourth major development stage—namely, Version 1.3.

Version 1.1 included the option of defining quasi-experimental factors. From that version, random distribution to selected experimental conditions can also be based on specified combinations of levels of quasi-experimental factors.

In Version 1.2, a number of features, including within-subjects factorial designs and a layout differing from previous versions, were implemented to enhance usability.

The current public Version 1.3 contains a content wizard for creating interactive materials, as well as dependent measures (graphical scales, multiple-choice items, etc.), on the experiment pages.

Outlook

WEXTOR will be further integrated with the Web Experimental Psychology Lab and the *web experiment list*³ on which Web experiments can be linked for recruiting Internet participants. Because WEXTOR still needs extended

help functions, we are planning extensive networking with the materials developed in the Swiss Virtual Campus⁴ project entitled “Experimental Design and Web-based Experimentation” in order to enable *learning by doing* in (Web) experimental design.

Other improvements will include easier handling of within-subjects designs, implementation of nested factors, and automatic storage of the generated HTML and JavaScript code in text files on an FTP server. Templates for page layout and frequently used pages, such as informed consent forms and general instructions, will also be offered. In a recent effort to find an attractive logo for WEXTOR, we arranged an on-line vote on 20 different proposed logos for participation by every WEXTOR user. We also opened a mailing list⁵ for beta testers and users of WEXTOR. We hope that WEXTOR will act as a useful tool for Web and off-line experimenters, as well as for learning and teaching methodological concepts in psychology.

REFERENCES

- BIRNBAUM, M. H. (2000). SurveyWiz and FactorWiz: JavaScript Web pages that make HTML forms for research on the Internet. *Behavior Research Methods, Instruments, & Computers*, *32*, 339-346.
- BUCHANAN, T., & REIPS, U.-D. (2001, October 10). Platform-dependent biases in online research: Do Mac users really think different? In K. J. Jonas, P. Breuer, B. Schauenburg, & M. Boos (Eds.), *Perspectives on Internet research: Concepts and methods*. Retrieved December 27, 2001, from <http://server3.uni-psych.gwdg.de/gor/contrib/buchanan-tom>.
- FRICK, A., BÄCHTIGER, M. T., & REIPS, U.-D. (2001). Financial incen-

- tives, personal information, and drop out in online studies. In U.-D. Reips & M. Bosnjak (Eds.), *Dimensions of Internet science* (pp. 209-219). Lengerich: Pabst.
- KIRK, R. E. (1995). *Experimental design: Procedures for the behavioral sciences* (3rd ed.). Pacific Grove, CA: Brooks/Cole.
- MARTIN, D. W. (2000). *Doing psychology experiments* (5th ed.). Pacific Grove, CA: Brooks/Cole.
- REIPS, U.-D. (1997). Das psychologische Experimentieren im Internet [Psychological experimenting on the Internet]. In B. Batinic (Ed.), *Internet für Psychologen* (pp. 245-265). Göttingen: Hogrefe.
- REIPS, U.-D. (2000). The Web experiment method: Advantages, disadvantages, and solutions. In M. H. Birnbaum (Ed.), *Psychological experiments on the Internet* (pp. 89-114). San Diego: Academic Press.
- REIPS, U.-D. (2001). The Web Experimental Psychology Lab: Five years of data collection on the Internet. *Behavior Research Methods, Instruments, & Computers*, **33**, 201-211.
- REIPS, U.-D. (in press). Internet-based psychological experimenting: Five dos and five don'ts." *Social Science Computer Review*.
- SCHWARZ, S., & REIPS, U.-D. (2001). CGI versus JavaScript: A Web experiment on the reversed hindsight bias. In U.-D. Reips & M. Bosnjak (Eds.), *Dimensions of Internet science* (pp. 75-90). Lengerich: Pabst.
- SULLIVAN, L. (1947). The tall office building artistically considered. In I. Athey (Ed.), *Kindergarten chats (revised 1918) and other writings* (pp. 202-213). New York: George Wittenborn. (Original work published 1896)

NOTES

1. The term *Web experiment* is largely synonymous with the widely used terms *Internet experiment*, *on(-)line experiment*, *Web-based experiment*, *WWW experiment*, and *Internet-based experiment*, because experiments using Internet services other than the World-Wide Web (such as e-mail, ICQ, Telnet, Gopher, FTP, etc.) are rarely conducted. Because WEXTOR generates code for experiments based on Web pages, the term *Web experiment* is used in this article.

2. <http://www.genpsy.unizh.ch/Ulf/Lab/WebExpPsyLab.html>

3. <http://www.genpsy.unizh.ch/Ulf/Lab/webexplist.html>

4. <http://www.virtualcampus.ch/>

5. <http://groups.yahoo.com/group/wextor>

(Manuscript received November 20, 2001;
revision accepted for publication April 2, 2002.)