Behavior Research Methods
2005, 37 (2), 312-323

Psychology Experiment Authoring Kit (PEAK): Formal usability testing of an easy-to-use method for creating computerized experiments

WALTER SCHNEIDER
University of Pittsburgh, Pittsburgh, Pennsylvania
and Psychology Software Tools, Pittsburgh, Pennsylvania

D. J BOLGER
University of Pittsburgh, Pittsburgh, Pennsylvania

and

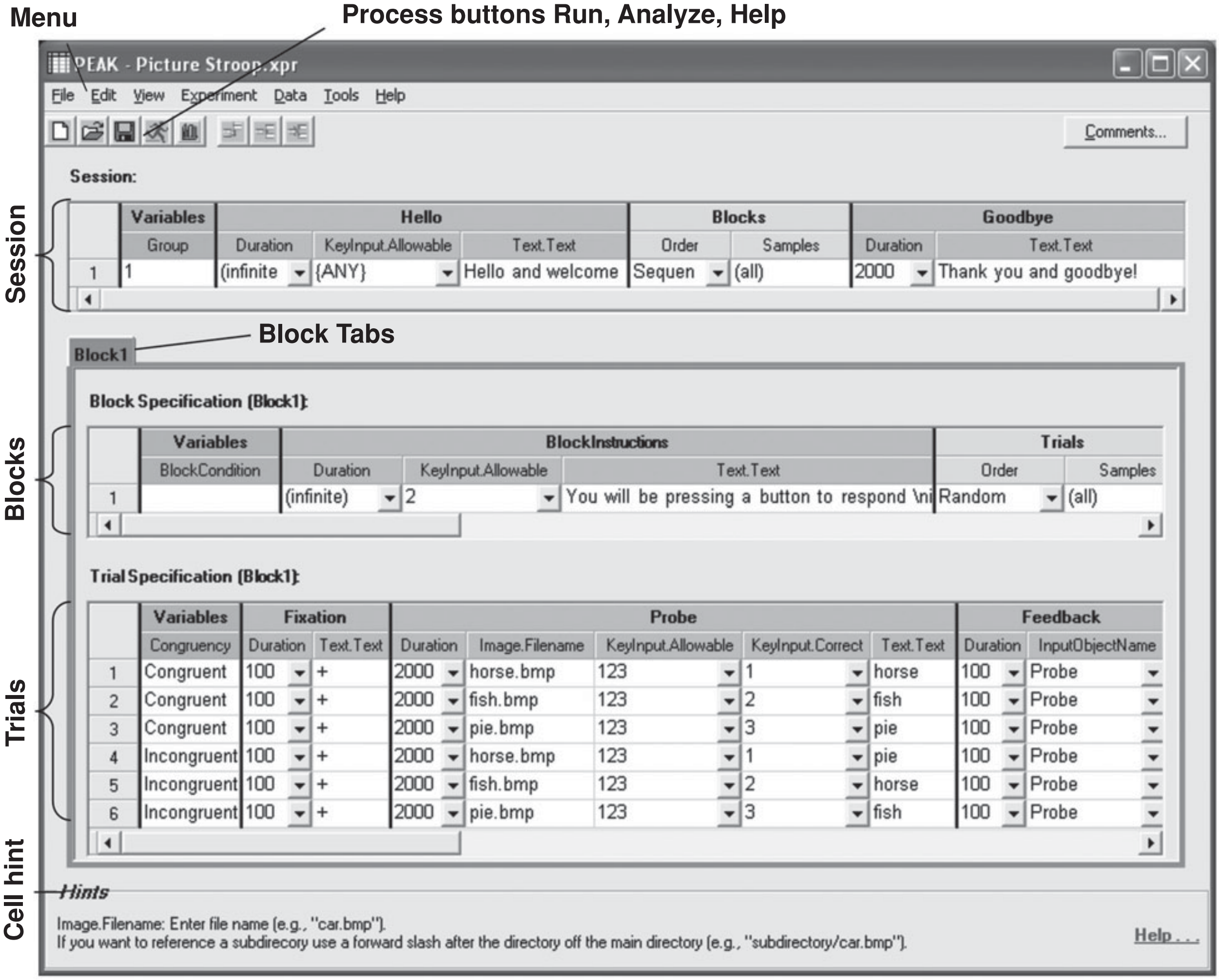
AMY ESCHMAN, CHRISTOPHER NEFF, and ANTHONY P. ZUCCOLOTTO
Psychology Software Tools, Pittsburgh, Pennsylvania

In academic courses in which one task for the students is to understand empirical methodology and the nature of scientific inquiry, the ability of students to create and implement their own experiments allows them to take intellectual ownership of, and greatly facilitates, the learning process. The Psychology Experiment Authoring Kit (PEAK) is a novel spreadsheet-based interface allowing students and researchers with rudimentary spreadsheet skills to create cognitive and cognitive neuroscience experiments in minutes. Students fill in a spreadsheet listing of independent variables and stimuli, insert columns that represent experimental objects such as slides (presenting text, pictures, and sounds) and feedback displays to create complete experiments, all within a single spreadsheet. The application then executes experiments with centisecond precision. Formal usability testing was done in two stages: (1) detailed coding of 10 individual subjects in one-on-one experimenter/subject videotaped sessions and (2) classroom testing of 64 undergraduates. In both individual and classroom testing, the students learned to effectively use PEAK within 2 h, and were able to create a lexical decision experiment in under 10 min. Findings from the individual testing in Stage 1 resulted in significant changes to documentation and training materials and identification of bugs to be corrected. Stage 2 testing identified additional bugs to be corrected and new features to be considered to facilitate student understanding of the experiment model. Such testing will improve the approach with each semester. The students were typically able to create their own projects in 2 h.

This report details a new approach to experiment generator software for undergraduate methodology courses and basic research experimentation and provides a novel approach to formal usability testing of the software in an ongoing classroom setting. We have developed the Psychology Experiment Authoring Kit (PEAK), a novel spreadsheet method of developing and communicating experiments that enables students with minimal computer knowledge (e.g., rudimentary spreadsheet skills) to learn to use the system in less than 2 h, and to create their own simple experiments within an hour.

The experiment specification provides a very transparent interface (see Figure 1) in which all the key variables, events, and specifications are illustrated in a well-structured single-page spreadsheet interface. This provides a pedagogical tool for understanding key experimental concepts, such as independent and dependent variables, counterbalancing, and precise procedural control. The system is designed to accommodate experimental paradigms covering much of the current computerized experimental literature. This interface is built on the *minimum learning step* approach that utilizes basic computer knowledge and skills (e.g., the Excel spreadsheet) to minimize learning time. For undergraduate research, it is priced at a modest level (less than the typical textbook) and is included in PsychMate (see Eschman, St. James, Schneider, & Zuccolotto, 2005), so each student can afford to own the system and use it on his or her personal computer.

This work was supported by NIH Grant 1R43MH072020-01, Managing Complexity in Psychology Experiment Generation. We thank Erik Reichle, Julie Fiez, and their students at the University of Pittsburgh for participating in the usability study. PEAK and PsychMate are products of Psychology Software Tools, Inc., 2050 Ardmore Boulevard, Suite 200, Pittsburgh, PA 15221 (www.pstnet.com). The materials used to assess PEAK are available to researchers upon request for testing ease of use and to illustrate the staged testing procedure described. Components of the PEAK interface are under patent application. One or more of the authors has a direct financial interest in some of the software or hardware described in this article. Correspondence should be addressed to W. Schneider, Learning, Research and Development Center, University of Pittsburgh, 3939 O'Hara Street, Pittsburgh, PA 15221 (e-mail: wws@pitt.edu).


Menu
Process buttons Run, Analyze, Help
PEAK - Picture Stroop.xpr
File Edit View Experiment Data Tools Help
Comments...
Session
Block Tabs
Block1
Blocks
Trials
Cell hint
Hints
Help . . .

Session:

	Variables	Hello			Blocks		Goodbye	
	Group	Duration	KeyInput.Allowable	Text.Text	Order	Samples	Duration	Text.Text
1	1	(infinite	{ANY}	Hello and welcome	Sequen	(all)	2000	Thank you and goodbye!

Block Specification (Block1):

	Variables	BlockInstructions			Trials	
	BlockCondition	Duration	KeyInput.Allowable	Text.Text	Order	Samples
1		(infinite)	2	You will be pressing a button to respond \n	Random	(all)

Trial Specification (Block1):

	Variables	Fixation		Probe					Feedback	
	Congruency	Duration	Text.Text	Duration	Image.Filename	KeyInput.Allowable	KeyInput.Correct	Text.Text	Duration	InputObjectName
1	Congruent	100	+	2000	horse.bmp	123	1	horse	100	Probe
2	Congruent	100	+	2000	fish.bmp	123	2	fish	100	Probe
3	Congruent	100	+	2000	pie.bmp	123	3	pie	100	Probe
4	Incongruent	100	+	2000	horse.bmp	123	1	pie	100	Probe
5	Incongruent	100	+	2000	fish.bmp	123	2	horse	100	Probe
6	Incongruent	100	+	2000	pie.bmp	123	3	fish	100	Probe

Image.Filename: Enter file name (e.g., "car.bmp").
If you want to reference a subdirecory use a forward slash after the directory off the main directory (e.g., "subdirectory/car.bmp").

Figure 1. PEAK interface illustrating the session, block, and trial levels of the experiment, with columns showing experiment specification and stimulus parameters.

There has been a long history of experiment generator programs. There are many approaches, ranging from script-based systems (Bates & D'Oliveiro, 2003; Dutta, 1995; Eberhardt, Neverov, & Haneef, 1997; Forster & Forster, 2003; Haussmann, 1992; Hawley, 1991; Hunt, 1994; Kessels, Postma, & de Hean, 1999; Pallier, Dupoux, & Jeannin, 1997; Palya & Walter, 1993; Palya, Walter, & Cho, 1995; Pulkin, 1996), hypercard (Chute, 1993; Cox, Hulme, & Brown, 1992), form-based systems (MEL; Schneider, 1989), experiment diagram approaches (PsyScope; Cohen, MacWhinney, Flatt, & Provost, 1993), cross-linked lists (SuperLab; Haxby, Parasuraman, Lalonde, & Abboud, 1993), and graphical interfaces (E-Prime; Schneider, Eschman, & Zuccolotto, 2002). These approaches fall at various points along a continuum in which *power* and *flexibility* are traded off with *speed of learning* and *ease of use*. For example, programming languages provide the greatest power for experiment generation at the greatest cost, whereas list templates provide modest learning time with very limited power. Graphical interfaces provide intermediate power at intermediate effort.

Almost all the experiment generators described in the journal *Behavior Research Methods, Instruments, & Computers* (see the experiment generation references above) make some claim that the packages are "easy to use." However, none of these articles have provided formal usability data on learning time to support the *ease-of-use* qualification. *Ease of use* is a vague concept that, at the very least, requires some operational definition of the term and quantitative data to support such a claim. For example, using Microsoft Excel is a typical standard of an "easy to use" product, yet the first author is still learning new ways to use spreadsheets after more than 25 years of experience. Simple time to learn to use the system needs to be defined. A good interface is often one in which the user quickly learns to perform basic actions and then continues to gain flexibility as he or she uses the product (Cooper, 1995).

In this article, we seek to set empirical criteria for *ease of use*. We suggest the following standard: An experiment generator package is *easy to use* if a *typical undergraduate psychology major can learn to use it in 2 h of laboratory time and can then create his or her own simple experiment in a median time of 2 h.*[1]

Although the present authors have created multiple experiment generators in the past 30 years (Schneider, 1989; Schneider, et al., 2002; Schneider & Scholz, 1973), we have in the past felt that none was sufficiently easy to use to the point that it was reasonable to teach in a standard undergraduate research methods course. The approaches used in experiment generators in the past have been too cryptic to be learnable by the bottom quartile of undergraduates without, typically, more than 10 h of instruction and, hence, do not fit into the typical 2 h class model and limited laboratory time available during a term (e.g., less than 8 h of lab time for a student to run and analyze his or her project).

The typical undergraduate methods class (in psychology) provides a heterogeneous mix of students, many with only rudimentary computer skills. Although many students are capable of surfing the Web or using a word processor, most have negligible programming skills. In an advanced laboratory course in cognitive psychology, in which basic research methods is a prerequisite (P0420 at the University of Pittsburgh), fewer than 15% could, on entry to the course, effectively use a spreadsheet to calculate the mean and standard deviation of a set of values within a single experimental condition. For many students, this course represents the first time ever in which they have used a spreadsheet program, and most have never performed a single calculation in such a program. Instructors must allocate limited class time to teach the basics of spreadsheet editing (e.g., copying, pasting, calculating means and standard deviations, creating tables) in the initial lectures.

Our department of psychology has come to view it as an obligation that the average psychology major should, by graduation, be able to use a spreadsheet to analyze results from an experiment with multiple independent variables. Getting the majority of students to a level of expertise at which they can use Excel to create pivot tables of dependent measures coding experimental variables typically takes repeated instruction across 3–4 h of class time and working through multiple examples.

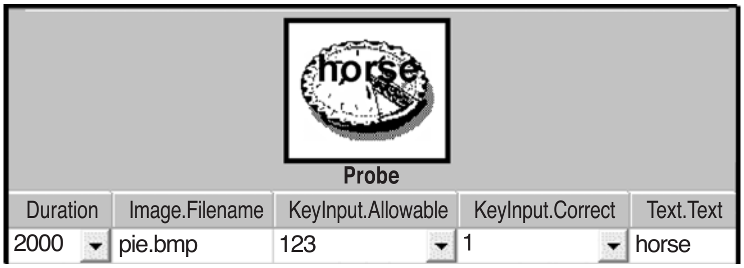
At the basic level, we assume that the spreadsheet-based computer skills of the typical researcher are (1) the ability to fill in values in a spreadsheet, (2) the ability to use menus to add/move/delete columns/rows, (3) the use of intelligent fill (sweeping out a set of cells and then extending the set to fill in more rows and columns on the basis of the data in selected rows), and (4) the ability to click on cells and buttons. Given these skills, our goal is to teach students to create novel experiments within 2 h.

We decided to develop a new interface approach to an experiment generator package that would build knowledge of experiment creation upon the foundation of basic spreadsheet skills we had already developed in our methods classes. The spreadsheet interface was the "killer application" that transformed budget calculations in business and accounting. This interface can have a similar impact on computer-based psychological experimentation. The current PEAK interface bootstraps onto the students' developing spreadsheet skills to provide a foundation for experiment creation (i.e., every spreadsheet row represents a trial, merged column headers represent objects such as slides, and columns set properties that determine what is presented and what responses are collected). Students are already experienced at looking at a table and expecting the information to be organized by rows and columns, providing affordances as to what actions can be taken.

The PEAK interface implementing a picture Stroop experiment (Rosinski, 1977) is shown in Figure 1. In the experiment, the subject is shown a picture with overlaid text (e.g., a line drawing of a horse with an overlaid text of "horse") and is asked to respond to the picture while ignoring the text. The PEAK spreadsheet in Figure 1 illustrates a complete specification of a picture Stroop experiment (Stroop, 1935) in which a fixation is presented for 1,000 msec, then the probe stimulus (e.g., the image of a "pie" and the text of "horse") is presented for up to 2,000 msec. The creation of a typical experiment involves the following steps.

1. Open PEAK, pick a model template experiment or an experiment from a list of existing experiments.
2. Add/remove the number of objects that make up an experimental trial (e.g., for trials including a fixation and a probe, add two slides containing text, bitmaps, and sounds).
3. Add/remove property columns as needed to the slides (e.g., if you need the property to specify a picture file to be displayed, add Image.Filename)
4. Fill in the *block* and *trial* (independent) variable names.
5. Fill in the first row of the spreadsheet for the first trial, typically adding values of the independent variables, text, picture filename, allowable/correct response keys, and stimulus durations.
6. Use the spreadsheet's intelligent fill feature to create the desired number of trials for the block.
7. Edit the cells of the block (e.g., word stimuli in a lexical decision experiment).
8. If graphics need to be created, use PowerPoint to create pictures.
9. Press the "Run" button in PEAK to output the experiment and pass it to the runtime engine to then execute the experiment and collect the data.
10. Press the "Analyze" button to analyze the results, providing data sheets for the experiment that was run. Use Tools to perform descriptive and inferential statistics, automatically create PowerPoint presentation slides or summary Web pages of the results, and transfer the results to other analysis programs.

To illustrate the construction of a probe object in the picture Stroop (see Figure 2), the student would define the



Probe

Duration	Image.Filename	KeyInput.Allowable	KeyInput.Correct	Text.Text
2000	pie.bmp	123	1	horse

Figure 2. Slide object specification illustrating an image of what a subject sees and the settings of properties.

first-row property of values to be: (1) Duration = “2000” (to set the duration of the stimulus), (2) Image.Filename = “pie.bmp” (to set the picture to be displayed), (3) KeyInput.Allowable = “123” (to set the group of response keys that the subject is permitted to press), (4) KeyInput.Correct = “1” (to set the key response that is considered to be correct for this trial), and (5) Text.Text = “horse” (to set the text that will be displayed).

They would also set the “Fixation” object text property to “ + ” and set a property on the “Feedback” input object to “Probe” (i.e., identifying the input on which the feedback is based) and to provide performance feedback on each trial. This entails only a handful of keystrokes (26 to be precise) for a complete specification of a trial. Next, they would select the first row and extend the rows to the desired number of trials, using intelligent fill. Then they would visit each row and edit the information that will change from one trial to the next (e.g., Image.Filename, Text.Text, KeyInput.Correct). The total picture Stroop experiment, presenting six trials with pictures and text, can be created by typing 50 keystrokes, thus requiring only the basic data entry skills of a spreadsheet.

The PEAK spreadsheet minimizes typing time dramatically, but more important, it allows the user to build a clear mental model of the experiment, using familiar tools and editing a spreadsheet on a single screen. All the user needs to know is how to type and add experimental objects, such as slides, questionnaires, text displays (e.g., moving window), movies, control objects (e.g., exit when performance criterion is reached), and feedback displays. Each row is a trial, and individual blocks are represented by tabs (like worksheets) in Excel. Experimenters do not have to think in terms of a script, variables, or experimental subroutines. Like Excel, PEAK provides context-specific help for every option, and from all cells and components of the spreadsheet.

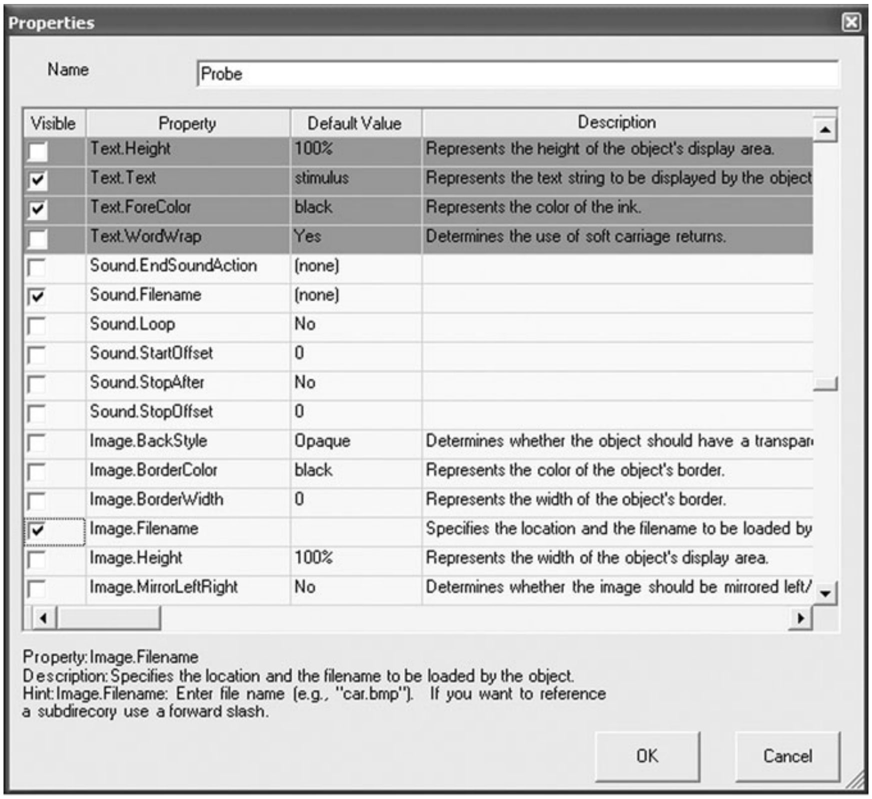
Addition Versus Removal

The PEAK spreadsheet provides extensive power while minimizing complexity. The two major methods of creating an experiment are illustrated with an analogy to sculpture. One can create a statue either by removing material from a base shape (e.g., stone carving) or from scratch (e.g., pottery). In an experiment, the advantage of removal is that the options are already visible and the experimenter needs only to examine the pop-up help and fill in the values. The disadvantage of removal is that, as the number of properties/columns gets large, the spreadsheet can become complex and daunting. The advantage of addition is that only those options that a researcher actually needs are visible.

An innovation in this project was to guide the student by using an experimentally sophisticated spreadsheet technology. The spreadsheet is aware of all the experimental object types and properties and the order/syntax of a well-formed experiment. The experimentally aware spreadsheet provides *extensive property options* to maximize flexibility while containing complexity. To get started with a new experiment, the researchers can load a template or existing experiment whose general design is similar to the expected experiment (i.e., template) and then remove or add columns/properties, using pop-up menus and/or a common properties dialogue box. The interface provides documentation of each option, and the researcher need only select and delete either an object (e.g., slide) or a property (e.g., Image.Stretch) in order to make modifications to the existing experiment.

The sophisticated spreadsheet technology provides either a simple spreadsheet table of the heading of the objects and object properties or an iconic display of what the subject will see. In order to support and promote the storyboard concept within the PEAK interface, we intend to give a design time visualization for each object in a future enhancement. Figure 2 presents a rendering of this prototype feature. As the experimenter selects different rows of the trial spreadsheet, the upper displays will change to illustrate what the subject would experience, thus providing a close visual linkage between the storyboard, the spreadsheet, and what the subject will see.

Providing help for the adding or removing of objects/properties is a serious challenge, because the novice undergraduate user does not know about the existence of, or the formal names for, specific properties. There are potentially several hundreds of options that are difficult to find in a menu format. To meet this problem, PEAK has context-sensitive menus (available via right clicking) that pop up selectable options for objects/properties (see Figure 3). The properties are grouped into cat-



Name: Probe

Visible	Property	Default Value	Description
□	Text.Height	100%	Represents the height of the object's display area.
☑	Text.Text	stimulus	Represents the text string to be displayed by the object
☑	Text.ForeColor	black	Represents the color of the ink.
□	Text.WordWrap	Yes	Determines the use of soft carriage returns.
□	Sound.EndSoundAction	(none)	
☑	Sound.Filename	(none)	
□	Sound.Loop	No	
□	Sound.StartOffset	0	
□	Sound.StopAfter	No	
□	Sound.StopOffset	0	
□	Image.BackStyle	Opaque	Determines whether the object should have a transpar
□	Image.BorderColor	black	Represents the color of the object's border.
□	Image.BorderWidth	0	Represents the width of the object's border.
☑	Image.Filename		Specifies the location and the filename to be loaded by
□	Image.Height	100%	Represents the width of the object's display area.
□	Image.MirrorLeftRight	No	Determines whether the image should be mirrored left/

Property: Image.Filename
Description: Specifies the location and the filename to be loaded by the object.
Hint: Image.Filename: Enter file name (e.g., "car.bmp"). If you want to reference a subdirecory use a forward slash.

Figure 3. Properties page slide object providing a complete list of all potential properties. The current specifications are visibility, default value, and description.

egories (e.g., general, response, text, images, sound) to aid location, and the full list of properties can be examined, sorted, and colored by category.

To deal with issues of randomization and number of trials, we created a selector object (see "TrialSelector" object in Figure 1). By right clicking on the TrialSelector object, property options appear that set counterbalancing, randomization, timing, and termination conditions.

A challenge when dealing with the needs of a the novice user is that one must be cautious of giving users capabilities that, in the hands of a novice, can produce more harm than good. In PEAK, groups of columns/properties are routinely added. In a spreadsheet, one can add sets of cells as needed. However, in a complex table that has merged cells, this raises problems for the novice user (e.g., in Excel, when the cells are added under a merged cell, the top, merged column will not be extended). Repairing misshifted rows and columns can be very difficult for the novice. To limit such problems in PEAK, adding, hiding, and deleting columns and rows is available only by clicking buttons or selecting menu options that never permit misalignment of the columns/cells.

A second challenge is to visually provide the user *affordances* that intuitively inform the user of options (Norman, 2002). For example, how would a novice become aware of the need to specify "{ALPHA}" in the allowed keys field in order to accept any alphabetic character? PEAK informs the user of options by providing several types of cues. First, there is a hint at the bottom of the screen for each cell, detailing how to use it, with a hyperlink to more information (see Figure 1). Second, icons within cells indicate the presence of a dropdown menu that lists options where appropriate. Third, users are instructed to "right click the mouse" when they are confused, in order to get a list of all the actions that would be appropriate for the part of the spreadsheet on which they are working. Reminding students in class about these three methods provides them with problem-solving skills to learn more about the interface as needed.

Software Implementation

We implemented the system, providing the standard spreadsheet editing features of Excel, to support transfer of students' spreadsheet skills. We enhanced the interface in multiple ways to keep users out of trouble (e.g., no insertion of cells) and provide them affordances. We provide intuitive graphical displays (e.g., thumbnail versions of the image the subject would see). To provide flexibility, we built the system on top of the E-Prime system (Schneider et.al., 2002), providing a set of features of objects that have evolved in the package as it has grown to support over 10,000 researchers over the past 5 years. For example, the slide currently has 120 properties. All of these properties have commonly used defaults, so experimenters can ignore them until they have a need to modify them (e.g., there is no need to be concerned with the image-related property that controls mirror image reflection until it would be needed to implement a paradigm such as a mental rotation experiment). We implemented the system to support an E-Prime slide object, which is capable of simultaneously outputting text, pictures, and sound files with millisecond accuracy. The output experiment specification is generated as an XML file, preprocessed by an E-Prime–based interpreter/translator application, and executed using the E-Prime runtime engine, which has been directly integrated into PEAK. The resulting data can be easily merged and exported to various statistical packages to support analysis. Most of the code in the system was written using C# and the Microsoft .NET Framework 2003. The runtime interpreter application was created as a customized E-Prime application (using extensive E-Basic code) that reads the XML experiment specification created by PEAK. The experiment runs with millisecond timing accuracy and centisecond response accuracy.

Usability Testing

The central goal of formal usability testing is the determination of the primary areas of difficulty in the current approach to experiment generation and to assess the overall level of usability of the system. If we can identify the areas of user difficulty, we can build systems that target precisely those areas, thereby improving the overall quality of psychological research. Also, usability testing can identify bugs and other design flaws, so they can be corrected. There are multiple views on usability testing, but the norm is that only a relatively small number of subjects are needed to detect a significant proportion of design problems within a specified area of testing. Faulkner (2003) reports needing only 5 subjects to detect 85% of the bugs in a simple calendar time entry task. Faulkner points out that there is high variability in low-end designs and that 20 users are needed in order to detect 90% of the bugs.

We developed a novel usability methodology that provides strong benefits in a university setting. We refer to this as a *staged individual and classroom user testing methodology*. We tested an initial set of 10 subjects, utilizing detailed single-subject user testing, followed by in-classroom group testing. The single-subject usability testing involved having individuals in a one-on-one instructional setting. The task was to learn the PEAK interface and to create a lexical decision experiment. The task session was videotaped, actions of the subject and the experimenter were event coded by an observer, and the accuracy of the experiments was scored by an independent rater. In the second (classroom) phase, we used the instructional materials in a standard classroom setting. The materials, available from the first author,[2] included (1) a PowerPoint minilecture on the use of PEAK, (2) help system documentation, (3) a step-by-step getting-started guide, (4) questionnaires, and (5) three experimental templates: number Stroop (Fox, Shor, & Steinman, 1971), color Stroop (Stroop, 1935), and picture Stroop (Toma & Tsao, 1985). The PEAK program was augmented to include an extensive usability-monitoring code to record every unified action that was taken by the user (e.g., each cell modification or menu option selected, what functions were utilized, and the duration of each action). Student subjects received questionnaires asking them to rate each of the instructional materials after each stage.

This staged approach has several benefits over traditional user testing. On the basis of Faulkner (2003), running 10 subjects would be expected to find a minimum of 82% and a mean of 95% of the problems in a simple interface. Correcting the most serious problems before a full classroom study would lead to a more effective and pleasant pedagogical experience for the classroom students. Second, the one-on-one format in which think-aloud protocols are used with individual students provides a rich data set with which to more deeply assess the underlying nature of problems that users have. This can be the basis for hypothesis testing to better understand and remedy problems. Unfortunately, single-user testing is very expensive and difficult to run on large samples, particularly since few new problems are detected. To catch low-frequency errors, we ran 64 subjects in a classroom setting. The second stage of running a large group of subjects should identify nearly all the problems (98% in Faulkner, 2003), plus test the software in the intended environment (i.e., a large classroom setting).

For the videotaped sessions, an event-coding program (written in E-Prime) was utilized to record the duration of each task by a human coder, as well as to keep track of any requests for help, indications of confusion, experimenter actions, and subject actions.

In 2 h, the students worked through a 49-page workbook, created a lexical decision experiment, and described how they would implement the Posner attention-cuing experiment (Posner, Snyder, & Davidson, 1980). Most of the materials were step-by-step guides with little text and with pictures of what needed to be altered in the interface. For the lexical decision experiment, the students were given the task description, pictures of the screens the subject would see, and the subjects' instructions: "Respond by pressing the key indicating whether the stimulus is (1) a word or (2) a nonword in a lexical decision experiment (e.g., for stimulus 'sep' press '2'). There will be a fixation slide presented for 0.5 seconds, the probe slide presented until response or a maximum of 2 seconds, and a feedback display stating if correct or incorrect." For the attention task, the experiment should present pictures. The subject instructions included the following: "Respond by pressing the key indicating if whether the stimulus is on the left or right. You will see two boxes, one will blink and then one will fill in (e.g., for a stimulus with the filled in box on the right you would press '2'). There will be a fixation slide presented for 0.5 seconds, a cue slide for 0.75 seconds, a delay slide for 0.5 seconds, and a probe slide presented until response or a maximum of 2 seconds."

Subjects. We ran the study in conjunction with the University of Pittsburgh Psychology 0420 advanced methods laboratory. The course meets twice a week for lecture and once a week for a 2 h laboratory. For the final paper assignment, students are required to generate their own experiments. The students were upper level juniors and seniors, and 46% were psychology majors.

Experiment 1: Individual videotaped users. We ran 10 videotaped subjects. In addition, 2 users were tested to shake down the procedures. The subjects included the faculty course instructor (assistant professor), 2 graduate student teaching assistants (advanced cognitive psychology graduate students in their 5th and 6th years of graduate school), and 7 students from the Psychology 0420 class. The timing data from the students are presented in Table 1. The time for the instructor was shorter (45.1 min) and that for the teaching assistants longer (58.1 min) than that for the students (51.0 min). The teaching assistants, who would be required to teach the material a week later, spent more time asking detailed questions about the interface than did the other subjects.

The experiment timing included a 5-min PowerPoint slideshow, exercises to familiarize the subjects with the interface, and two exercises in which experiments were created, using the interface in following a step-by-step guide. Thereafter, the subjects created a new text experiment, the lexical decision experiment, having been given a table of the independent variables and the trial stimuli, sample pictures of the displays, and the subject instructions.

The results show a dramatic success in exceeding our time criteria of 2 h to learn and create the experiment. It took the students just a median of 54 min of instruction and less than 5 min to create the lexical decision experiment. Although the experimenter was always available for questions, little time was spent dealing with questions (average, 20 sec). All of the experiments were created accurately and ran successfully. Overall, effective use of the experiment generator by the student subjects occurred in less than one fifth of the expected time it would take to perform a similar task in currently available interfaces (e.g., E-Prime). However, there is little data on similar products with which to make further comparisons. The results show that the subjects could very easily move from a list of stimuli to a PEAK specification of the experiment for a simple lexical decision task. The transfer of skills from existing spreadsheet knowledge occurred as we had predicted. That is, the subjects had little trouble with the interface and discovered new features as they progressed.

Table 1
Performance of Videotaped Students

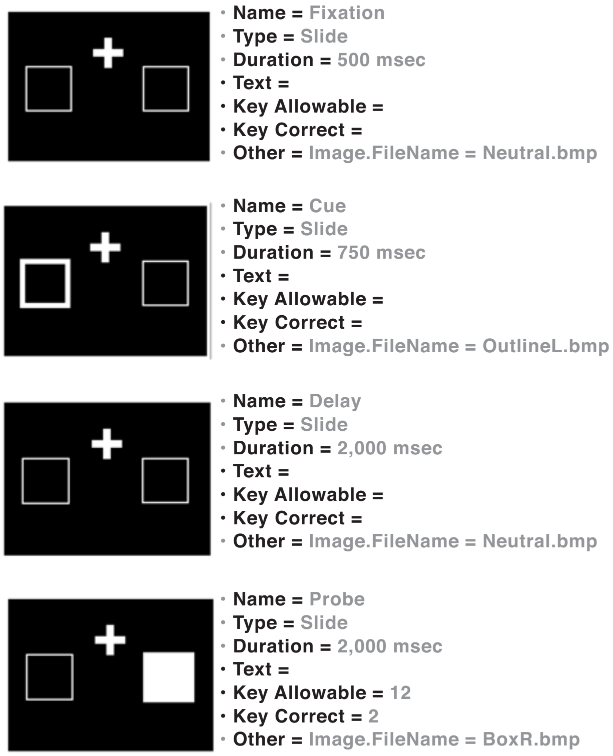
Tasks and Results	Minutes
Time in lecture and exercises	51.0
Median	54.2
Slideshow	4.8
Exercise 1: Familiarize with interface	9.5
Exercise 2: Add properties to create color Stroop	20.2
Exercise 3: Convert number Stroop to picture	6.5
Mean time to create lexical decision experiment	4.6
SD for lexical decision experiment	1.3
Median	4.7
Average time getting help	0.3

In general, the subjects found the documentation very useful (4.7 on a 5-point scale: 1, *not at all useful*; 5, *very useful*). They reported that they were able to complete the task quickly and easily (4.1 on a 5-point scale: 1, *not at all*; 5, *completely*). If given the opportunity, they would definitely use PEAK to create their own experiments (4.7 on a 5-point scale; 1, *not at all*; 5, *definitely*).

However, there were also surprises and reasons for concern. We ended the videotape session with asking them how they would implement a very different experiment, a Posner attentional-cuing experiment (Posner et al., 1980; see Figure 4). The subjects were given storyboard figures to show them the sequence of displays in a trial and were asked to explain (not implement) how they would go about implementing the experiment, using PEAK. Our very early student subjects were not capable of taking a novel experiment and conceptualizing how to implement it. For example, the instructions showed the four images of the display and asked them how many objects would be needed to implement the experiment (answer, four). Yet both verbal and nonverbal cues indicated a lack of knowledge. For example, the transcript between one of the subjects (S) and the experimenter (E) was

> E: Start out at the trial level. What objects would you add?
> S: Allowable keys are just gonna be "1" and "2." So, just double click on input.allowable and do "1" and "2" for all of those. KeyInput.Correct is going to refer to which side the light ends up being on, so "1" is for left and "2" is for right. You're not going to need text at all.
> E: Look at the example trial procedure. There is a fixation, a cue, a delay, and a probe. Right now you have a fixation and a probe. So what would you have to do to add in the cue and the delay?
> S: So, I can just double click on here (*double clicks Fixation object to display Properties*), and go down and find . . . . I'm looking for "cue."
> E: Actually, cancel out of here. We already have a Fixation, with a duration of . . . .
> . . . *Subject is focusing on the mechanics of entering properties rather than focusing on the definition of the objects for the trial procedure. Experimenter goes through various attempts to try to explain that events are objects, and an object has specific properties. Subject does not understand the object structure of the experiment.*
> . . . *Subject continues to focus on property settings. Experimenter explains that he must add objects and set property values for the missing events (Cue and Delay).*

The early subjects could not conceptualize the experiment as a set of slide events, even though they had to add a slide in one experiment and modify slides in two others. If given a slide, they could have set the properties of the slide, but their conceptualization of the experi-



- Name = Fixation
- Type = Slide
- Duration = 500 msec
- Text =
- Key Allowable =
- Key Correct =
- Other = Image.FileName = Neutral.bmp

- Name = Cue
- Type = Slide
- Duration = 750 msec
- Text =
- Key Allowable =
- Key Correct =
- Other = Image.FileName = OutlineL.bmp

- Name = Delay
- Type = Slide
- Duration = 2,000 msec
- Text =
- Key Allowable =
- Key Correct =
- Other = Image.FileName = Neutral.bmp

- Name = Probe
- Type = Slide
- Duration = 2,000 msec
- Text =
- Key Allowable = 12
- Key Correct = 2
- Other = Image.FileName = BoxR.bmp

Figure 4. Posner experiment storyboard with a diagram of what the slide stimulus looks like and the property settings of the slide object. Students are encouraged to create the storyboard on paper before implementing the experiment on the computer.

ment was too vague to conceptualize a new procedure with multiple objects. They seemed to struggle with not having a clear dissection of the trial to work with. These students lacked the capability to formally describe how an experiment would be implemented through slide objects and their properties. They required substantial hints from the experimenter to conceptualize the model experiment and to realize that a trial requires four slide objects to which particular properties must be set. Note that this conceptual failure occurred even though all of them had successfully added objects in the earlier exercises.

In review sessions of the videotapes, the experimenters and the design team worked to identify the problem and evaluated changes in the interface and instructions that could correct it. We felt there were two core problems that needed to be addressed. First, the users had to conceptualize the experiment before trying to implement it. The subjects immediately attended to the specific features (e.g., "what is the text that must be entered"), without having recognized that the overall design structure (four, not two, slide objects) needed to be implemented. Second, the users did not know the order of steps in which they should proceed. They would start changing columns in the spreadsheet often in nonproductive ways.

Two pedagogical tools were employed to remediate the students' deficiencies in achieving the goal of designing the experiment. The first was a conceptual model, the *storyboard*, by which the students could identify the design of an experimental trial, and the second, the *quick reference card*, was a simple outline provided to the student subjects to enable explicit goal/subgoal processing. We decided that users need to be trained in parsing an experiment into the events that make up the experiment and that this needs to be done before they become immersed in setting properties within the interface. It appears that the spreadsheet interface has a cost; the students interpreted adding columns as building an experiment and did not recognize that they had to add objects (not a spreadsheet concept) before working at the property level. We decided that this issue might best be addressed through instructional/documentation changes.

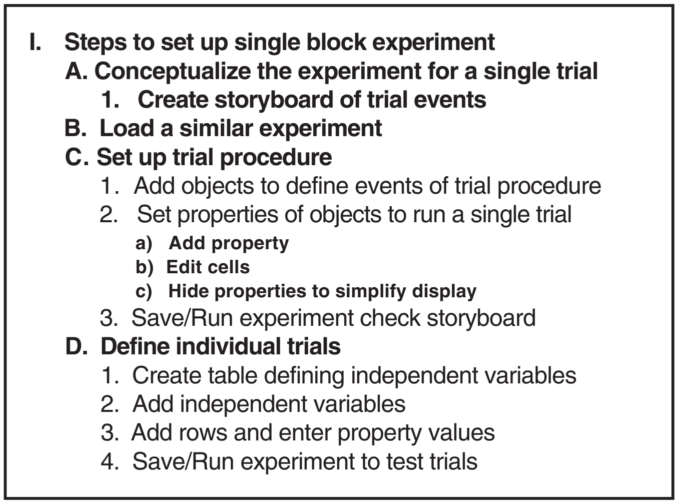
For several of our initial student subjects (4–7) in Stage 1, we first had them do an explicit storyboard of the experiment before they were to try to implement it. A storyboard for the Posner experiment is shown in Figure 4. We gave the students blank storyboard sheets containing rectangles and a list of the major properties (e.g., *name*, *duration*, and *text*, as in Figure 4) and had them draw the images, filling in the specifications (Figure 4, text after the "="). We found that they needed to do this exercise several times before they seemed to understand that experiments are created through the use of multiple slidelike objects.

The last subject during our individual testing was seen by the experimenters to be our most naive in terms of both computer skill and experimental knowledge, but she received the best-developed storyboard training. She, in contrast to the previous 6 subjects, did very well conceptualizing and describing how she would implement the Posner experiment. She described a coherent implementation plan:

> E: How many slides would you have in a Posner task on the trial procedure?
> S: Four
> E: Right. And they are the . . .?
> S: The Fixation, Cue, Delay, and the Probe
> E: And how would you add those?
> . . . *After being told she was allowed to work in the interface* . . .
> S: Well, I clicked on the "right" thing (*right-clicks to display context menu*) and I would probably add two objects.
> . . . *Continues to explain how she would position and set the properties for those objects.*

In order to address the poor goal-setting ability of student subjects, we substantially revamped the documentation and developed a very concrete set of steps describing how to generate an experiment—the first of which was to create the storyboard. We also provided a quick-reference card containing the steps (Figure 5 lists the steps for creating a trial taken from the quick-reference card).

In addition to the major conceptual problems, the initial user testing identified over 174 bugs and/or additional features that we should consider implementing. In



I. Steps to set up single block experiment
   A. Conceptualize the experiment for a single trial
      1. Create storyboard of trial events
   B. Load a similar experiment
   C. Set up trial procedure
      1. Add objects to define events of trial procedure
      2. Set properties of objects to run a single trial
         a) Add property
         b) Edit cells
         c) Hide properties to simplify display
      3. Save/Run experiment check storyboard
   D. Define individual trials
      1. Create table defining independent variables
      2. Add independent variables
      3. Add rows and enter property values
      4. Save/Run experiment to test trials

Figure 5. PEAK experiment creation steps provide a list of outline steps for creating an experiment. Each step can be clicked on to expose detailed help.

the 2 weeks between the videotape testing and the time of the in-class testing, 32 features were added, and 50 bugs were addressed. In addition, event tracking was added to the program to provide automatic tracking of users' operations in PEAK.

Experiment 2: Classroom test of PEAK. The goal and challenge of the classroom tests was to determine whether advanced undergraduate students could learn the basics of experiment creation and how to use the PEAK interface in 2 h of lab time. The classroom setting was quite different from the one-on-one instruction of the videotaped sessions. In this case, we presented the materials in a group setting. There were 64 students in three laboratory sections (18, 22, and 24) that were present in class on the day of testing. All the students had to use PEAK for class assignments that consisted of running pregenerated experiments. In accordance with human subject procedures, we asked for and obtained permission to record data from 46 of the class members.[3] These were advanced undergraduates who needed to learn PEAK because they would have to use it in performing their final psychology course project. They had little training in statistics, a basic knowledge of research methodology, and minimal computer skills (knowledge of Word and Excel at a basic level).

Table 2
PEAK Classroom Results: Median Task Durations

Session	Task	Minutes
1	PowerPoint lecture	30
	On screen demo of PEAK	10
	Exercise 1: Familiarize interface	12
	Exercise 2: A single block number Stroop task	23
	Exercise 3: A multiple block design color Stroop task	15
2	Exercise 4: Picture Stroop task	6
	Exercise 5: Lexical decision	8
	Exercise 6: Posner experiment	22
Total		126

In the classroom setting, the tasks were done in a group format with a group lecture followed by students working at their own pace. Notebooks were coded for the exercises, with page numbers in a large font so an experimenter could walk around the room and determine the median page number (and hence, the exercise) at which the students were working. We lengthened the lecture to 30 min, and it was presented by a faculty member. During that lecture, the students were encouraged to fill in storyboards on paper and coding sheets for the color Stroop and picture Stroop experiments. Thereafter, the correct answers were shown on the screen for students to compare with their own.

As an introduction to each task, the subjects were presented with an overview for each task in the getting-started guide. The overview provided a general description of the experiments they would be creating and the skills they would be learning in the process, as well as an explanation of the spreadsheet interface (Figure 1). The subjects were then asked to run a demo of the completed experiment, in order to illustrate the trial procedure, and to follow the steps outlined in the getting-started guide to complete each experiment. The students worked independently and at their own pace. At any point, they could click the "Comment" button and send comments to the developers (we received 108 such comments). The range of time to execute the exercises was between half and twice the median times. The time to complete each of the first four exercises was determined by tracking when half the class had completed each exercise (on the basis of identifying the page in the workbook at which they were). For Exercises 5 and 6, which involve the creation of new experiments, the total time spent working on the related files was recorded.

Exercise 1 served as an introduction to the PEAK interface and the Help system and to the concept of levels in an experiment (i.e., session, block, and trial). Exercise 2 led the students through conceptualization of a color Stroop experiment, using a storyboard process, translation of the storyboard to a PEAK spreadsheet, and defining trial-level variables. Exercises 3 and 4 introduced additional concepts, including multiblock experiments, image presentation, and randomization. Exercise 5 asked the students to modify an existing number Stroop experiment to create a lexical decision experiment (i.e., modifying the stimuli from color words to words or nonwords, and setting the appropriate values for the WordType independent variable). In Exercise 6, the students were asked to use what they had learned in previous tasks to create a Posner attention-cuing experiment on their own.

Between two and four Psychology Software Tools staff members, along with the individual instructors (graduate student teaching assistants, or TAs), were on hand to observe and assist the students as they worked through the six exercises in the getting-started guide. The types of questions the instructors encountered multiple times were the following: How to add an object? How to add an image? How to locate the properties of an object? There were very few such questions, however, averaging about one question per student during the class.

The classroom format provided some challenges in terms of classroom management and compliance, likely representing different group dynamics in the computerized laboratory classroom. For example, in the first lab, the students drifted in, and class began 20 min late; informal surveys from the back of the room suggested that half the students were surfing the Web or reading/writing e-mails during the lecture. In contrast, the other sections started on time, and nearly everyone was attentive. These differences in attention impacted the final performance. In the first section, only 5 students completed the Posner task (with a mean score of 8.0, *SD* = 2.12), based on a total of 16 points for getting every object and property correct. In the second section, 12 students completed the Posner task (with a mean score of 11.4, *SD* = 3.96), and in the third, 16 completed it (with a mean score of 12.88, *SD* = 3).

Overall, many students reported that the PEAK software was very straightforward and easy to use. On the survey item "Overall, I found the experiment software easy to use," the scores revealed a mean rating of 3.9 (*SD* = 0.8) on a 5-point scale (1, *strongly disagree*; 5, *strongly agree*). The survey item "Overall, I found the experiment software interface (mouse, pull-down menus, and dialog boxes) an easy method for performing experimental functions" received a mean rating of 4.2 (*SD* = 0.7) on the same scale. Since PEAK resembles Excel, which all the students were familiar with, due to instruction earlier in the course, they had few problems (and complained about some useful or expected Excel features not being implemented, such as the "undo" button).

Key data on successful implementation indicated that the subjects, on average, could convert the color Stroop experiment into a lexical decision task in roughly 8 min. All the subjects completed the lexical decision experiment accurately. Creating the Posner task was more challenging, since it had a different format than all the practice exercises and required the addition of two new slide objects and new concepts of a prime and a delay slide between the fixation and the probe. The average time to create the Posner experiment was 22 min. Accuracies varied, with a mean of 11.1 out of 16 for setting all the parameters of all the trials correctly. Most subjects (88%) were able to create a functional experiment with some errant parameters (e.g., stimulus duration, correct response, etc.). If the subjects had more carefully checked these parameters and tested the experiment (i.e., by running it on themselves), we expect that they would have produced fully accurate experiments. (Note that the exercise was not part of the course grading and that, after the students had implemented the goal experiments in some form, they went on to work on their personal class projects.)

In addition, the PEAK application time stamped and recorded user activities and operations during their use of the application (of the 46 students that gave permission for recording); we collected 5,509 events characterized on the dimensions of User, Date, Exercise, End-State, Time, Load File, CurrentFile, CumulativeTime, TaskTime, CurrentTab, Level, LevelID, Object, Property, Row, Action, InterfaceElement, BeginningValue, EndValue. General quantitative data indicate that 86% of the time (75% of the action counts), the users worked at the trial level of the experiment, 13% at the block level, and 6% at the session level. The users spent 70% of their time working on slide properties, 15% on independent variables, and 10% on block/trial randomization. These data support the view that the key areas of efficiency we should address are how users configure slides, particularly at the trial level of the experiment.

Real-world classroom testing exposed problems that needed to be addressed. For any given class, an average of 15% of the students were absent. When these students appeared in the second class, the other students were further ahead. With the very detailed instructional materials we had available, new students could learn the material on their own after receiving a short (5 min) overview by the instructor, while the other students proceeded with other exercises in the workbook. Students who were not attentive to the lectures (e.g., surfed the Web) took longer to go through the exercises and had greater difficulty with the implementation of the experiments. We do not feel it is productive to fault documentation that is only minimally attended to by the learner. Hence, we think it is appropriate to encourage students to attend to and to consider, for usability purposes, the data primarily from students who (1) are present for the presentation of the materials, (2) are fairly attentive (e.g., attend to the lecture 75% of the time), and (3) do the exercises, with 75% of the in-class time dedicated to the exercises. In addition, when class schedules shift and the introduction of new material is delayed until late in the laboratory period, time should be viewed as cumulative across periods, rather than as time in a single laboratory session.

Students were able, in later laboratory sessions not involving specific instruction, to create their own first drafts of experiments—typically, in about 2 h.

Discussion. User testing is a cyclical process, not a single event. The major results of the first cycle include the following. (1) The PEAK software provides a markedly faster learning time than that expected from previous experiment generator approaches. (2) PEAK was found to be *easy to use* in formal user testing, in that the median subject learned to use the interface effectively in a 2 h laboratory session (median, 96 min), agreed with the statement that it was "easy to use" ($M = 3.9$, $SD = 0.8$, with 4 being *agree* on a 5-point scale), could create a simple experiment in 8 min, and could create an experiment with modest complexity in 22 min. (3) Staged individual and in-classroom user-testing methodology provided a rich data set at modest cost that allowed debugging and targeted improvement of the software, lectures, and documentation and identified problems to fix and new features to add. (4) The students were responsive, providing useful suggestions for new features and reporting problems. (5) User testing exposed a range of problems (e.g., difficulty in conceptualization, documentation bugs, dealing with class management in presenting new software). (6) Addressing these problems on the basis of experimenter/developer review of individual testing sessions resulted in positive payoffs (most students in the class study group successfully implemented the Posner experiment, which was not true of the initial 7 subjects before the revamping of the documentation).

This staged individual and classroom methodology can be applied to better formally represent *ease of use*, and to produce better instructional materials and more effective products. We believe that there is a complementary benefit of individual and in-class testing. We do not think we would have realized, in the group-testing format, that the key problem in the initial version was not the interface but, rather, the need to provide a conceptual framework (provided through the storyboard) and to link it directly to the steps users should execute. It was examination of the think-aloud protocols that made apparent the nature of the problem. Thereafter, the design/documentation team could consider the problem and propose solutions. The instructional materials were altered from subject to subject until we felt we had a solution. The classroom testing had the benefit that it efficiently exposed many bugs and suggested features. We anticipate using this method on an ongoing basis in classes as we add new features to PEAK each semester over the next 2 years.

It is surprising to find that the field of software development in psychological experimentation has operated in the last 30 years without a formal assessment of *ease of use*. The testing in this project yielded substantial benefits (e.g., identified 174 bugs or features to be addressed, collected 108 suggestions/comments, identified serious problems in the user knowledge/documentation that blocked successful completion of experiments by initial users, and provided the foundation to alter the documentation so that nearly all the students could create experiments successfully). Such tests are expensive (we estimate three person months to carry out the test). This is a significant cost, although only a small proportion (<5%) of the total software development cost. This project had the benefits of grant support for the testing and a large and cooperative development team (three programmers, a documentation writer, two user testers, a teaching assistant, and a supervising scientist). We believe that formal user testing will provide a high return in terms of substantial benefit to tens of thousands of students using the package in the years ahead. Furthermore, we believe that the result of our effort with PEAK as a pedagogical tool will substantially advance the training of students in empirical methodology and in understanding the nature of scientific inquiry.

REFERENCES

Bates, T. C., & D'Oliveiro, L. (2003). PsyScript: A Macintosh application for scripting experiments. *Behavior Research Methods, Instruments, & Computers*, **35**, 565-576.

Chute, D. L. (1993). MacLaboratory for Psychology: Successes, failures, economics, and outcomes over its decade of development. *Behavior Research Methods, Instruments, & Computers*, **25**, 180-188.

Cohen, J., MacWhinney, B., Flatt, M., & Provost, J. (1993). PsyScope: An interactive graphic system for designing and controlling experiments in the psychology laboratory using Macintosh computers. *Behavior Research Methods, Instruments, & Computers*, **25**, 257-271.

Cooper, A. (1995). *About face: The essentials of user interface design*. New York: Wiley.

Cox, R., Hulme, C., & Brown, G. D. A. (1992). STM Experimenter: Using HyperCard and MacRecorder in short-term memory experiments. *Behavior Research Methods, Instruments, & Computers*, **24**, 575-579.

Dutta, A. (1995). Experimental RunTime System: Software for developing and running reaction time experiments on IBM-compatible PCs. *Behavior Research Methods, Instruments, & Computers*, **27**, 516-519.

Eberhardt, S. P., Neverov, M., & Haneef, O. (1997). RunScript: An extendable object-oriented program for computer-controlled psychology experiments. *Behavior Research Methods, Instruments, & Computers*, **29**, 313-321.

Eschman, A., St. James, J., Schneider, W., & Zuccolotto, A. (2005). PsychMate: Providing psychology majors the tools to do real experiments and learn empirical methods. *Behavior Research Methods, Instruments, & Computers*, **37**, 301-311.

Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, & Computers*, **35**, 379-383.

Forster, K. I., & Forster, J. C. (2003). DMDX: A Windows display program with millisecond accuracy. *Behavior Research Methods, Instruments, & Computers*, **35**, 116-124.

Fox, L. A., Shor, R. E., & Steinman, R. J. (1971). Semantic gradients and interference in naming color, spatial direction, and numerosity. *Journal of Experimental Psychology*, **91**, 59-65.

Haussmann, R. E. (1992). Tachistoscopic presentation and millisecond timing on the IBM PC/XT/AT and PS/2: A Turbo Pascal unit to provide general-purpose routines for CGA, Hercules, EGA, and VGA monitors. *Behavior Research Methods, Instruments, & Computers*, **24**, 303-310.

Hawley, K. J. (1991). PsyExper: Another experimental generation system for the IBM PC. *Behavior Research Methods, Instruments, & Computers*, **23**, 155-159.

Haxby, J. V., Parasuraman, R., Lalonde, F., & Abboud, H. (1993). SuperLab: General-purpose Macintosh software for human experimental psychology and psychological testing. *Behavior Research Methods, Instruments, & Computers*, **25**, 400-405.

Hunt, S. M. J. (1994). MacProbe: A Macintosh-based experimenter's workstation for the cognitive sciences. *Behavior Research Methods, Instruments, & Computers*, **26**, 345-351.

Kessels, R. P. C., Postma, A., & deHaan, E. H. F. (1999). Object Relocation: A program for setting up, running, and analyzing experiments on memory for object locations. *Behavior Research Methods, Instruments, & Computers*, **31**, 423-428.

Norman, D. A. (2002). *The design of everyday things*. New York: Basic Books.

Pallier, C., Dupoux, E., & Jeannin, X. (1997). EXPE: An expandable programming language for on-line psychological experiments. *Behavior Research Methods, Instruments, & Computers*, **29**, 322-327.

Palya, W. L., & Walter, D. E. (1993). A powerful, inexpensive experiment controller or IBM PC interface and experiment control language. *Behavior Research Methods, Instruments, & Computers*, **25**, 127-136.

Palya, W. L., Walter, D. E., & Chu, J. Y. M. (1995). An inexpensive 1-millisecond experiment control interface for IBM PCs and its user-friendly control language. *Behavior Research Methods, Instruments, & Computers*, **27**, 129-130.

Posner, M. I., Snyder, C. R. R., & Davidson, B. J. (1980). Attention and the detection of signals. *Journal of Experimental Psychology: General*, **109**, 160-174.

Pulkin, B. V. (1996). Programming without programming: The system Butterfly for professional psychologists. *Behavior Research Methods, Instruments, & Computers*, **28**, 577-583.

Rosinski, R. R. (1977). Picture–word interference is semantically based. *Child Development*, **48**, 643-647.

Schneider, W. (1989). Enhancing a standard experimental delivery system (MEL) for advanced psychological experimentation. *Behavior Research Methods, Instruments, & Computers*, **21**, 240-244.

Schneider, W., Eschman, A., & Zuccolotto, A. (2002). *E-Prime: A user's guide*. Pittsburgh: Psychology Software Tools.

Schneider, W., & Scholz, K. W. (1973). Requirements for minicomputer operating systems for human experimentation and an implementation on a 4K PDP-8 computer. *Behavior Research Methods, Instruments, & Instrumentation*, **5**, 173-177.

Stroop, J. R. (1935). Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, **18**, 643-662.

Toma, R. J., & Tsao, Y.-C. (1985). Interference effects in the picture–word Stroop task. *Perceptual & Motor Skills*, **61**, 223-228.

NOTES

1. Given the skewed distribution and the fact that, in some classes, some students had great difficulty or may not have turned in assignments, we feel the median is a better measure than the mean.

2. These are included as a separate file for review.

3. Each student was required to fill in a consent form on which they could select to have data collected or not. On the basis of their selection, they were assigned a subject number that had encrypted the consent level. If they chose not to contribute data, the data were not written to the data file. All statistics are based on the 46 consenting subjects. The instructors were not informed about the consent status. Help was given to all the subjects, without awareness of the consent status.

(Manuscript received December 8, 2004;
accepted for publication February 15, 2005.)