Chapter 14

Testing and modeling users

14.1 Introduction
14.2 User testing
  14.2.1 Testing MEDLINEplus
14.3 Doing user testing
  14.3.1 Determine the goals and explore the questions
  14.3.2 Choose the paradigm and techniques
  14.3.3 Identify the practical issues: Design typical tasks
  14.3.4 Identify the practical issues: Select typical users
  14.3.5 Identify the practical issues: Prepare the testing conditions
  14.3.6 Identify the practical issues: Plan how to run the tests
  14.3.7 Deal with ethical issues
  14.3.8 Evaluate, analyze and present the data
14.4 Experiments
  14.4.1 Variables and conditions
  14.4.2 Allocation of participants to conditions
  14.4.3 Other practical issues
  14.4.4 Data collection and analysis
14.5 Predictive models
  14.5.1 The GOMS model
  14.5.2 The Keystroke level model
  14.5.3 Benefits and limitations of W M S
  14.5.4 Fitts' Law

14.1 Introduction

A central aspect of interaction design is user testing. User testing involves measuring the performance of typical users doing typical tasks in controlled laboratory-like conditions. Its goal is to obtain objective performance data to show how usable a system or product is in terms of usability goals, such as ease of use or learnability. More generally, usability testing relies on a combination of techniques including observation, questionnaires and interviews as well as user testing, but user testing is of central concern, and in this chapter we focus upon it. We also examine key issues in experimental design because user testing has developed from experimental practice, and although there are important differences between them there is also commonality.

The last part of the chapter considers how user behavior can be modeled to predict usability. Here we examine two modeling approaches (based on psychological theory) that have been used to predict user performance. Both come from the well-known GOMS family of approaches: the GOMS model and the Keystroke level model. We also discuss Fitts' Law.

The main aims of this chapter are to:

- Explain how to do user testing.
- Discuss how and why a user test differs from an experiment.
- Discuss the contribution of user testing to usability testing.
- Discuss how to design simple experiments.
- Describe the GOMS model, the Keystroke level model and Fitts' law and discuss when these techniques are useful.
- Explain how to do a simple keystroke level analysis.

14.2 User testing

User testing is an applied form of experimentation used by developers to test whether the product they develop is usable by the intended user population to achieve their tasks (Dumas and Redish, 1999). In user testing the time it takes typical users to complete clearly defined, typical tasks is measured and the number and type of errors they make are recorded. Often the routes that users take through tasks are also noted, particularly in web-searching tasks. Making sense of this data is helped by observational data, answers to user-satisfaction questionnaires and interviews, and key stroke logs, which is why these techniques are used along with user testing in usability studies.

The aim of an experiment is to answer a question or hypothesis to discover new knowledge. The simplest way that scientists do this is by investigating the relationship between two things, known as ***variables.*** This is done by changing one of them and observing what happens to the other. To eliminate any other influences that could distort the results of this manipulation, the scientist attempts to control the experimental environment as much as possible.

In the early days, experiments were the cornerstone of research and development in user-centered design. For example, the Xerox Star team did experiments to determine how many buttons to put on a mouse, as described in Box 14.1. Other early experimental research in HCI examined such things as how many items to put in a menu and how to design icons.

Because user testing has features in common with scientific experiments, it is sometimes confused with experiments done for research purposes. Both measure performance. However, user testing is a systematic approach to evaluating user performance in order to inform and improve usability design, whereas research aims to discover new knowledge.

Research requires that the experimental procedure be rigorous and carefully documented so that it can be replicated by other researchers. User testing should

BOX 14.1 The Origins of User Testing

Xerox's Star office workstation was a landmark in interaction design. It was based on several user-centered design principles that are now well accepted, but at the time were revolutionary. The following principles guided the Star's development (Bewley et al., 1990):

- There should be an explicit, consistent conceptual model that draws on objects and activities already familiar to the user—the origins of the now familiar desktop metaphor.
- Seeing and pointing are easier than recalling and typing—the origins of the mouse and GUI.
- Commands should be uniform across similar domains—the important principle of consistency.
- The screen should show the state of the object the user is working on—what you see is what you get (WYSIWYG, pronounced "whizee-wig").

EVEN WITH THESE PRINCIPLES, the design space was still enormous and many proposed designs turned out to be unsatisfactory. Various tools and techniques were tried to support its development, including the keystroke level model discussed later (Card et al., 1983), but one of the most important decisions was to experiment and test design ideas intensively—i.e., to design and evaluate iteratively.

These tests included controlled experiments in which the evaluators describe their methodology in the language of science. For example, they tested six mouse selection schemes "using a between-subject paradigm, in which each of six groups was assigned one of the six schemes" (Bewley et al., 1990, p. 371). In addition, they also did more informal tests as the questions to be settled became less well defined. "... experiments took on a flavor of 'fishing expeditions' to see what we came up with" (Bewley et al., 1990, p. 380).

The design effort required for the Star, without doubt a mammoth undertaking, took more than six years. The implementation involved from 20–45 programmers over 3.5 years producing over 250,000 lines of high-level code. Over 15 human factor tests were performed using over 200 users and lasting over 400 hours. Each provided invaluable information about design decisions that were being made.

Two other early pioneers in usability testing were John Bennett, from IBM in the US, who helped to define usability and Brian Shackel from HUSAT in the UK, who worked to operationalize Bennet's definition so that it could be tested and measured. This involved taking vague notions such as "easy to use" and specifying what was meant. All this work paved the way for the development of current user testing practices.

be carefully planned and executed, but real-world constraints must be taken into account and compromises made. It is rarely exactly replicable, though it should be possible to repeat the tests and obtain similar findings. Experiments are usually validated using statistical tests, whereas user testing rarely employs statistics other than means and standard deviations.

Typically 5–12 users are involved in user testing (Dumas and Redish, 1999), but often there are fewer and compromises are made to work within budget and schedule constraints. "Quick and dirty" tests involving just one or two users are frequently done to get quick feedback about a design idea. Research experiments generally involve more participants, more tightly controlled conditions, and more extensive data analysis in which statistical analysis is essential.

14.2.1 Testing MEDLINEplus

In Chapter 13 we described how heuristic evaluation was used to identify usability problems in the National Library of Medicine (NLM) MEDLINEplus website (Figure 14.1 Cogdill, 1999). We now return to that study and focus on how the user testing was done to evaluate changes made after heuristic evaluation. This case study exemplifies the kinds of issues to be considered in user testing, including developing tasks and test procedures, and approaches to data collection and analysis.

Goals and questions

The goal of the study was to identify usability problems in the revised interface. More specifically, the evaluators wanted to know if the revised way of categorizing information, suggested by the expert evaluators, worked. They also wanted to check that users could navigate the system to find the information they needed. Navigating around large websites can be a major usability problem, so it was important to check that the design of MEDLINEplus supported users' navigation strategies.

Selection of participants

MEDLINEplus was tested with nine participants selected from primary health care practices in the Washington, DC metropolitan area. This was accomplished by

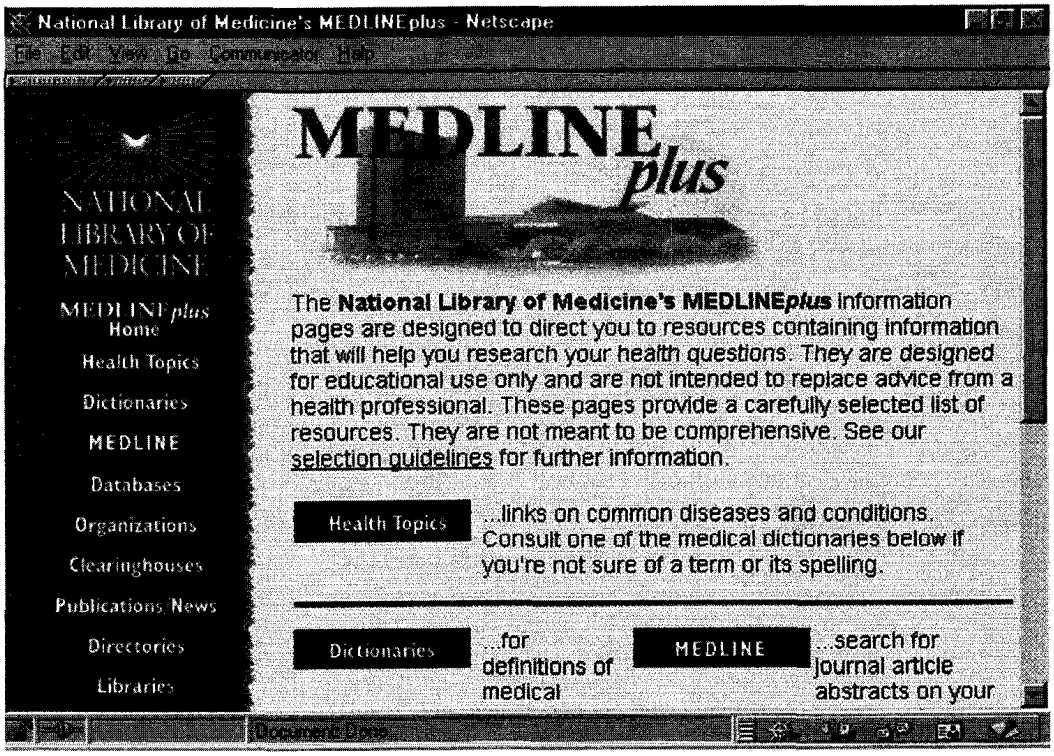


Figure 14.1 Home page of MEDLINEplus.

placing recruitment posters in the reception areas of two medical practices. People who wanted to participate were asked to complete a brief questionnaire, which asked about age, experience in using the web, and frequency of seeking health-related information. Dr. Cogdill, a usability specialist, then called all those who used the web more than twice a month. He explained that they would be involved in testing a product from the NLM, but did not mention MEDLINEplus so that potential testers would not review the site before doing the tests. Seven of the nine participants were women because balancing for gender was considered less important than web experience. It was important to find people in the Washington, DC region so that they could come to the test center and for the number of participants to fall within the range of 6–12 recommended by usability experts (Dumas and Redish, 1999).

Development of the tasks

The following five tasks were developed in collaboration with NLM staff to check the categorizing schemes suggested by the expert evaluators and navigation support. The topics chosen for the tasks were identified from questions most frequently asked by website users:

- Task 1: Find information about whether a dark bump on your shoulder might be skin cancer.

  Task 2: Find information about whether it's safe to use Prozac during pregnancy.
- Task 3: Find information about whether there is a vaccine for hepatitis C.

  Task 4: Find recommendations about the treatment of breast cancer, specifically the use of mastectomies.
- Task 5: Find information about the dangers associated with drinking alcohol during pregnancy.

The efficacy of each task was reviewed by colleagues and pilot tested.

The test procedure

The procedure involved five scripts that were prepared in advance and were used for each participant to ensure that all participants were given the same information and were treated in the same way. We present these scripts in figures to distinguish them from our own text. They are included here in their original form.

Testing was done in laboratory-like conditions. When the participants arrived they were greeted individually by the evaluator. He followed the script in Figure 14.2.

The participant was then asked to sit down at a monitor, and the goals of the study and test procedure were explained. Figure 14.3 shows the script used by the evaluator to explain the procedure to each participant (Cogdill, 1999), so that any performance differences that occurred among participants could not be attributed to different procedures.

Thank you very much for participating in this study.

The goal of this project is to evaluate the interface of MEDLINEplus. The results of our evaluation will be summarized and reported to the National Library of Medicine, the federal agency that has developed MEDLINEplus. Have you ever used MEDLINEplus before?

You will be asked to use MEDLINEplus to resolve a series of specific, health-related information needs. You will be asked to "think aloud" as you search for information with MEDLINEplus.

We will be videotaping only what appears on the computer screen. What you say as you search for information will also be recorded. Your face will not be videotaped, and your identity will remain confidential.

I'll need you to review and sign this statement of informed consent. Please let me know if you have any questions about it. ***(He hands an informed consent form similar to the one in Box 11.3 to the participant.)***

Figure 14.2 The script used to greet participants in the MEDLINEplus study.

We'll start with a general overview of MEDLINEplus. It's a web-based product developed by the National Library of Medicine. Its purpose is to link users with sources of authoritative health information on the web.

The purpose of our work today is to explore the MEDLINEplus interface to identify features that could be improved. We're also interested in finding out about features that are particularly helpful.

In a few minutes I'll give you five tasks. For each task you'll use MEDLINEplus to find health-related information.

As you use MEDLINEplus to find the information for each task, please keep in mind that it is MEDLINEplus that is the subject of this evaluation — not you.

You should feel free to work on each task at a pace that is normal and comfortable for you. We ***will*** be keeping track of how long it takes you to complete each task, but you should not feel rushed. Please work on each task at a pace that is normal and comfortable for you. If any task takes you longer than ***twenty*** minutes, we will ask you to move on to the next task. The Home button on the browser menu has been set to the MEDLINEplus homepage. We'll ask you to return to this page before starting a new task.

As you work on each task, I'd like you to imagine that it's something you or someone close to you needs to know.

All answers can be found on MEDLINEplus or on one of the sites it points to. But if you feel you are unable to complete a task and would like to stop, please say so and we'll move on to the next task.

Before we proceed, do you have any questions at this point?

Figure 14.3 The script used to explain the procedure.

Before starting the main tasks the participants were invited to explore the website for up to 10 minutes and to think aloud as they moved through the site. Figure 14.4 contains the script used to describe how to do this exploration task.

Each participant was then asked to work through the five tasks and was allowed up to 20 minutes for each task. If they did not finish a task they were asked to stop and if they forgot to think out loud or appeared to be stuck they were prompted. The evaluator used the script in Figure 14.5 to direct participants' behavior (Cogdill, 1999).

Before we begin the tasks, I'd like you to explore MEDLINEplus independently for as long as ten minutes.

As you explore, please "think aloud." That is, please tell us your thoughts as you encounter the different features of MEDLINEplus.

Feel free to explore any topics that are of interest to you.

If you complete your independent exploration before the ten minutes are up, please let me know and we'll proceed with the tasks. Again, please remember to tell us what you're thinking as you explore MEDLINEplus.

Figure 14.4 The script used to introduce and describe the initial exploration task.

Please read aloud this task before beginning your use of MEDLINEplus to find the information.

After completing each task, please return to the MEDLINEplus home page by clicking on the "home" button.

Prompts: "What are you thinking?"
"Are you stuck?"
"Please tell me what you're thinking."
[*If time exceeds 20 minutes:* "I need to ask you to stop working on this task and proceed to the next one."]

Figure 14.5 The script used to direct participants' behavior.

When all the tasks were completed, the participant was given a post-test questionnaire consisting of items derived from the QUIS user satisfaction questionnaire (Chin et al., 1988) described in Chapter 13. Finally, when the questionnaire was completed, there was a debriefing (Figure **14.6)** in which participants were asked for their opinions.

How did you feel about your performance on the tasks overall?

Tell me about what happened when [cite problem/error/excessive time].

What would you say was the best thing about the MEDLINEplus interface?

What would you say was the worst thing about the MEDLINEplus interface?

Figure 14.6 The debriefing script used in the MEDLINEplus study.

Data collection

Criteria for successfully completing each task were developed in advance. For example, participants had to find and access between 3–9 web page URLs. Each user's search moves were then recorded for each task. For example, the log revealed that Participant A visited the online resources shown in Table **14.1** while trying to complete the first task.

Completion times were automatically recorded and calculated from the video and interaction log data. The data from the questionnaire and the debriefing session

Table 14.1 The resources visited by participant A for the first task.

Databases
Home
MEDLINEIPubMed: "dark bump"
MEDLINEIPubMed: "bump"
Home
Dictionaries
External: Online Medical Dictionary
Home
Health Topics
Melanoma (HT)
External: American Cancer Society

were also used to help understand each participant's performance. The data collected contained the following:

- start time and completion time
- page count (i.e., pages accessed during the search task)
- external site count (i.e., number of external sites accessed during the search task)
- medical publications accessed during the search task
- the user's search path
- any negative comments or mannerisms observed during the search
- user satisfaction questionnaire data

ACTIVITY 14.1 What do you notice about how the user testing fits into the overall usability testing?

Comment

The user testing is closely integrated with the other techniques used in usability testing—questionnaires, interviews, thinkaloud, etc. In concert they provide a much broader picture of the user's interaction than any single technique would show.

Data analysis

Analysis of the data focused on such things as:

- website organization such as arrangement of topics, menu depth, organization of links, etc.
- browsing efficiency such as navigation menu location, text density, etc.
- the search features such as search interface consistency, feedback, terms, etc.

For example, Table 14.2 contains the performance data for the nine subjects for task 1. It shows the time to complete the task and the different kinds of searches undertaken. Similar tables were produced for each task. The exploration and questionnaire data was also analyzed to help explain the results.

Table 14.2 Performance data for task 1: Find information about whether a dark bump on your shoulder might be skin cancer. Mean (M) and standard deviation (SD) for all subjects are also shown.

Participant	Time to nearest minute	Reason for task termination	MEDLINEplus Pages	External sites accessed	MEDLINEplus searches	MEDLINE publication searches
A	12	Successful completion	5	2	0	2
B	12	Participant requested termination	3	2	3	0
C	14	Successful completion	2	1	0	0
D	13	Participant requested termination	5	2	1	0
E	10	Successful completion	5	3	1	0
F	9	Participant requested termination	3	1	0	0
G	5	Successful completion	2	1	0	0
H	12	Successful completion	3	1	0	6
I	6	Successful completion	3	1	0	0
M	10		3	2	1	1
SD	3		1	1	1	2

ACTIVITY 14.2 Examine Table 14.2.

(a) Why are letters used to indicate participants?

(b) What do you notice about the completion times when compared with the reasons for terminating tasks (i.e., completion records)?

(c) What does the rest of the data tell you?

Comment

(a) Participants' names should be kept confidentialin reports, so a coding scheme is used.

(b) Completion times are not closely associated with successful completion of this task. For example, completion times range from 5–14 minutes for successful completion and from 9–13 minutes for those who asked to terminate the task.

(c) From the data it appears that there may have been several ways to complete the task successfully. For example, participants A and C both completed the task successfully but their records of visiting the different resources differ considerably.

Conclusions and reporting the findings

The main finding was that reaching external sites was often difficult. Furthermore, analysis of the search moves revealed that several participants experienced difficulty finding the health topics pages devoted to different types of cancer. The post-test questionnaire showed that participants' opinions of MEDLINEplus were fairly neutral. They rated it well for ease of learning but poorly for ease of use because there were problems in going back to previous screens. These results were fed back to the developers in an oral presentation and in a written report.

ACTIVITY 14.3

(a) Was the way in which participants were selected appropriate and were there enough participants? Justify your comments.

(b) Why do you think participants were asked to read each new task aloud before starting it and to return to the home page?

(c) Was the briefing material adequate? Justify your comment.

Comments

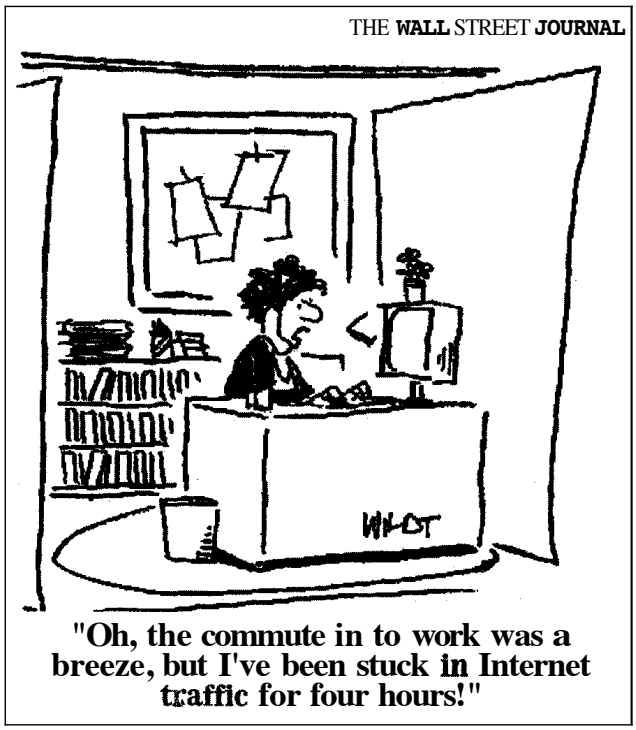
(a) This way of selecting participants was appropriate for user testing. The evaluator was careful to get a number of representative users across the user age range from both genders. Participants were screened to ensure that they were experienced web users. The evaluator decided to select from a local volunteer pool of participants, to ensure that he got people who wanted to be involved and who lived locally. Since using the web is voluntary, this is a reasonable approach. The number of participants was adequate for user testing.

(b) This was to make it easy for the evaluator to detect the beginning of a new task on the video log. Sending the participants back to the home page before starting each new task ensured that logging always started from the same place. It also helped to orient the participants.

(c) The briefing material was full and carefully prepared but not excessive. Participants were told what was expected of them and the prompts were preplanned to ensure that each participant was treated in the same way. An informed consent form was also included.

14.3 Doing user testing

There are many things to consider before doing user testing. Controlling the test conditions is central, so careful planning is necessary. This involves ensuring that the conditions are the same for each participant, that what is being measured is indicative of what is being tested and that assumptions are made explicit in the test design. Working through the DECIDE framework will help you identify the necessary steps for a successful study.



14.3.1 Determine the goals and Explore the questions

User testing is most suitable for testing prototypes and working systems. Although the goal of a test can be broad, such as determining how usable a product is, more specific questions are needed to focus the study, such as, "can users complete a certain task within a certain time, or find a particular item, or find the answer to a question" as in the MEDLINEplus study?

14.3.2 Choose the paradigm and techniques

User testing falls in the usability testing paradigm and sometimes the term "user testing" is used synonymously with usability testing. It involves recording data using a combination of video and interaction logging, user satisfaction questionnaires, and interviews.

14.3.3 Identify the practical issues: Design typical tasks

Deciding on which tasks to test users' performance is critical. Typically, a number of "completion" tasks are set, such as finding a website, writing a document or creating a spreadsheet. Quantitative performance measures are obtained during the tests that produce the following types of data (Wixon and Wilson, 1997):

- time to complete a task
- time to complete a task after a specified time away from the product

- number and type of errors per task
- number of errors per unit of time
- number of navigations to online help or manuals
- number of users making a particular error
- number of users completing a task successfully

As Deborah Mayhew (1999) reports, these measures slot neatly into usability engineering specifications which specify:

- current level of performance
- minimum acceptable level of performance
- target level of performance

The type of test prepared will depend on the type of prototype available for testing as well as study goals and questions. For example, whether testing a paper prototype, a simulation, or a limited part of a system's functionality will influence the breadth and complexity of the tasks set.

Generally, each task lasts between 5 and 20 minutes and is designed to probe a problem. Tasks are often straightforward and require the user to find this or do that, but occasionally they are more complex, such as create a design, join an online community or solve a problem, like those described in the MEDLINEplus and HutchWorld studies. Easy tasks at the beginning of each testing session will help build users' confidence.

14.3.4 Identify practical issues: Select typical users

Knowing users' characteristics will help to identify typical users for the user testing. But what is a typical user? Some products are targeted at specific types of users, for example, seniors, children, novices, or experienced people. HutchWorld, for example, has a specific user audience, cancer patients, but their experience with the web differs so a range of users with different experience was important. It is usually advisable to have equal numbers of males and females unless the product is specifically being developed for the male or female market. One of the most important characteristics is previous experience with similar systems. If the user population is large you can use a short questionnaire to help identify testers, as in the MEDLINEplus study.

ACTIVITY 14.4 Why is it important to select a representative sample of users whenever possible?

Comment It is important to have a representative sample to ensure that the findings of the user test can be generalized to the rest of the user population. Selecting participants according to clear objectives helps evaluators to avoid unwanted bias. For example, if 90% of the participants testing a product for 9–12 year-olds were 12, it would not be representative of the full age range. The results of the test would be distorted by the large group of users at the top-end of the age range.

DILEMMA How Many Users are Enough?

Deciding how many users to test is partly a logistical issue that depends on schedules, budgets, participants and facilities available. Many professionals recommend that 5–12 testers is enough (Dumas and Redish, 1999). Others say that as soon as the same kinds of problems start being revealed and there is nothing new, it is time to stop. However, the more testers there are, the more representative the findings will be across the user population.

14.3.5 Identify practical issues: Prepare the testing conditions

User testing requires the testing environment to be controlled to prevent unwanted influences and noise that will distort the results. Many companies, such as Microsoft and IBM, test their products in specially designed usability laboratories to try to prevent this (Lund, 1994). These facilities often include a main testing laboratory, with recording equipment and the product being tested, and an observation room where the evaluators sit and subsequently analyze the data. There may also be a reception area for testers, a storage area, and a viewing room for observers. Such labs are very expensive and labor-intensive to run.

The space may be arranged to superficially mimic features of the real world. For example, if the product is an office product or for use in a hotel reception area, the laboratory can be set up to match. But in other respects it is artificial. Soundproofing and lack of windows, telephones, fax machines, co-workers, etc. eliminate most of the normal sources of distraction. Typically there are two to three wall-mounted video cameras that record the user's behavior, such as hand movements, facial expression, and general body language. Utterances are also recorded and often a keystroke log.

The observation room is usually separated from the main laboratory by a one-way mirror so that evaluators can watch testers but testers cannot see them. Figure 14.7 shows a typical arrangement. Video and other data is fed through to monitors

Figure 14.7 A usability laboratory in which evaluators watch participants on a monitor and through a one-way mirror.

in the recording room. While the test is going on, the evaluators observe and annotate the video stream, indicating events for later more detailed analysis.

The viewing room is like a small auditorium with rows of seats at different levels. It is designed so that managers and others can watch the tests. Video monitors display video and the managers overlook the observation room and into the laboratory through one-way mirrors. Generally only large companies can afford this extra room and it is becoming less common.

The reception area also has bathroom facilities so that testers do not have to go into the outside world during a session. Similarly, telephones in the laboratory do not connect with the outside world, so there are no distractions. The only communication occurs between the tester and the evaluators. The laboratory can be modified to include other features of the environment in which the product will be used if necessary, but it is always tightly controlled.

Many companies and researchers cannot afford to have a usability laboratory, or even to rent one. Instead, they buy mobile usability equipment (e.g., video, interaction logging system) and convert a nearby room into a makeshift laboratory. The mobile laboratory can also be taken into companies and packed away when not needed. This kind of makeshift laboratory is more amenable to the needs of user testing. Modifications may have to be made to test different types of applications. For example, Chris Nodder and his colleagues at Microsoft had to partition the space when they were testing early versions of NetMeeting, a videoconferencing product, in the mid-1990s, as Figure 14.8 shows (Nodder et al., 1999).

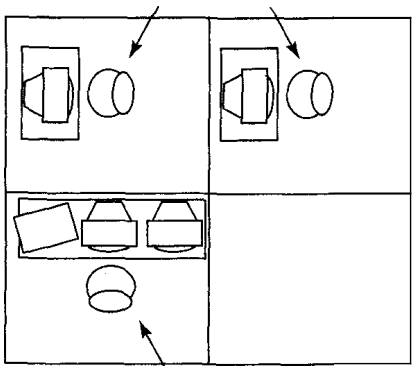


Figure 14.8 The testing arrangement used for NetMeeting videoconferencing system.

14.3.6 Identify practical issues: Plan how to run the tests

A schedule and scripts for running the tests, such as those used in MEDLINEplus, should be prepared beforehand. The equipment should be set up and a pilot test

performed to make sure that everything is working, the instructions are clear, and there are no unforeseen glitches.

It's a good idea to start the session with a familiarization task, such as browsing a website in a web usability study, so that participants can get used to the equipment before testing starts. An easy first task encourages confidence; ending with a fairly easy one makes participants go away feeling good. A contingency plan is needed for dealing with people who spend too long on a task, as in MEDLINEplus.

A query from the evaluator asking if the participant is all right can help. If the participant gets really stuck then the evaluator should tell him to move on to the next part of the task.

Long tasks and a long testing procedure should be avoided. It is a good idea to keep the session under one hour. Remember, all the data that is collected has to be analyzed and if you have nine participants who together generate nine hours of video, there is a lot to review and analyze.

14.3.7 Deal with ethical issues

As in all types of evaluation, you need to prepare and plan to administer an informed consent form. If the study is situated in a usability laboratory, it is also necessary to point out the presence of one-way mirrors, video cameras, and use of interaction logging.

14.3.8 Evaluate, analyze, and present the data

Typically performance measures (time to complete specified actions, number of errors, etc.) are recorded from video and interaction logs. Since most user tests involve a small number of participants, only simple descriptive statistics can be used to present findings: maximum, minimum, average for the group and sometimes standard deviation, which is a measure of the spread around the mean value. These basic measures enable evaluators to compare performance on different prototypes or systems or across different tasks. An increasing number of analysis tools are also available to support web usability analysis, particularly video analysis as mentioned in Chapter 12.

14.4 Experiments

Although classically performed scientific experimentation is usually too expensive or just not practical for most usability evaluations, there are a few occasions when it is used. For example, in a case study about the testing of a voice response system discussed later in Chapter 15 plenty of participants were available. The development schedule was flexible, and the evaluators knew that quantitative results would be well received by their clients, so they adopted a more experimental approach than usual. For this reason, and because the roots of user testing are in scientific experimentation and many undergraduate projects involve experiments, we will discuss experimental design.

The aim of an experiment is to answer a question or to test a ***hypothesis*** that predicts a relationship between two or more events known as ***variables.*** For example,

"Will the time to read a screen of text be different if 12-point Helvetica font is used instead of 12-point Times New Roman?" Such hypotheses are tested by manipulating one or some of the variables involved. The variable that the researcher manipulates is known as the *independent variable,* because the conditions to test this variable are set up independently before the experiment starts. In the example above, type font is the independent variable. The other variable, time to read the text, is called the *dependent variable* because the time to read the text *depends* on the way the experimenter manipulates the other variable, in this case which type font is used.

It is advisable to consult someone who is knowledgeable about relevant statistical tests before doing most experiments, rather than wondering afterwards what to do with the data that is collected.

14.4.1 Variables and conditions

Designs with one independent variable

In order to test a hypothesis, the experimenter has to set up the *experimental conditions* and find ways to control other variables that could influence the test result. So for example, in the experiment in which type font is the independent variable, there are two conditions:

Condition 1 = read screen of text in Helvetica font

Condition 2 = read screen of text in Times New Roman font

It is also helpful to have a *control condition* against which to compare the results of the experiment. For example, in the above test you could set up two control conditions: reading of the same text on printed paper, using Times font and reading of the same text on printed paper, using Helvetica font. The performance measures for both screen conditions could be compared with the paper versions.

Designs with two or more independent variables

Experiments are carried out in user testing usually to compare two or more conditions to see if users perform better in one condition than in the other. For example, we might wish to compare the existing design of a system (e.g., version 5.0) with a redesigned one (e.g., version 6.0). We would need to design a number of tasks that users would be tested on for both versions of the system and then compare their performance across these tasks. If their performance was statistically better in one condition compared with the other, we could say that the two versions were different. Supposing we were then interested in finding out whether the performance of different user groups was affected by the two versions of the system; how could we do this? We could split the users into two groups: those who are beginners and those who are expert users. We would then compare the performance of the two user groups across the two versions of the system. In so doing, we now have two independent variables each with two conditions: the version of the system and the experience of the user.

This gives us a 2 × 2 design as shown in the table.

Original design	Redesign
Beginners	Beginners
Experts	Experts

Deciding what it means to "perform better" involves determining what to measure; that is, what the dependent variables should be. Two commonly used dependent variables are the time that it takes to complete a task and the number of errors that users make doing the task.

Hypothesis testing can also be extended to include more variables. For example, three variables each with two conditions gives 2 X 2 × 2. In each condition the aim is to test the main effects of each combination and look for any interactions among them.

14.4.2 Allocation of participants to conditions

The discussion so far has assumed that different participants will be used for each condition but sometimes this is not possible because there are not enough participants and at other times it is preferable to have all participants take part in all conditions. Three well-known approaches are used: different participants for all conditions, the same participants for all conditions, and matched pairs of participants.

Different participants

In different participant design a single group of participants is allocated randomly to each of the experimental conditions, so that ***different*** participants perform in ***different*** conditions. There are two major drawbacks with this arrangement. The first is making sure that you have enough participants. The second is that if small groups are used for each condition, then the effect of any individual differences among participants, such as differences in experience and expertise, becomes a problem. Randomly allocating the participants and pre-testing to identify any participants that differ strongly from the others helps. An advantage is that there are no ***ordering effects,*** caused by the influence of participants' experience of one set of tasks on performance on the next, as each participant only ever performs in one condition.

Same participants

In same-participant design, all participants perform in all conditions so only ***half*** the number of participants is needed; the main reason for this design is to lessen the impact of individual differences and to see how performance varies across conditions for each participant. However, it is important to ensure that the ***order*** in which participants perform tasks does not bias the results. For example, if there are two tasks, A and B, half the participants should do task A followed by task B and the other half should do task B followed by task A. This is known as ***counterbalancing.***

Counterbalancing neutralizes possible unfair effects of learning from the first task, i.e., the order effect.

Matched participants

In matched-participants design, participants are matched in pairs based on certain user characteristics such as expertise and gender. Each pair is then randomly allocated to each experimental condition. This design is used when participants cannot perform in both conditions. The problem with this arrangement is that other important variables that haven't been taken into account may influence the results. For example, experience in using the web could influence the results of tests to evaluate the navigability of a website. So web expertise would be a good criterion for matching participants.

The advantages and disadvantages of using different experimental designs are summarized in Table 14.3.

Table 14.3 The advantages and disadvantages of different experimental designs

Design	Advantages	Disadvantages
Different participants	No order effects	Many participants needed. Individual differences among participants are a problem. Can be offset to some extent by randomly assigning to groups.
Same participants	Eliminates individual differences between experimental conditions.	Need to counterbalance to avoid ordering effects.
Matched participants	Same as different participants, but the effects of individual differences are reduced.	Can never be sure that subjects are matched across variables.

14.4.3 Other practical issues

Just as in user testing, there are many practical issues to consider and plan, for example where will the experiment be conducted, how will the equipment be setup, how will participants be introduced to the experiment, and what scripts are needed to standardize the procedure? Pilot studies are particularly valuable in identifying potential problems with the equipment or the experimental design.

14.4.4 Data collection and analysis

Data should be collected that measures user performance on the tasks set. These usually include response times, number of errors, and times to complete a task.

Analyzing the data involves knowing what to look for. Do the data sets from the two conditions look different or similar? Are there any extreme atypical values? If so, what do they reflect? Displaying the results on a graph will also help reveal differences.

The response times, errors, etc. should be averaged across conditions to see if there are any marked differences. Simple statistical tests like t-tests can reveal if these are significant. For example, a t-test could reveal whether Helvetica or Times font is slower to read on a screen. If there was no significance then the hypothesis would have to be refuted, i.e., the claim that Helvetica font is easier to read is not true.

Box 14.2 describes an experiment to test whether broad, shallow menu design is preferable to deep menus on the web.

BOX 14.2 An Experiment to Evaluate Structure in Web Page Design

A huge amount of work has been done on exploring the optimal number of items in a menu design, and most studies conclude that breadth is preferable to depth in organizing menu content. By this it is meant having a large number of top level menu items with few levels rather than a small number of top level items with many levels. Around 1997, when the web was still a relatively new phenomenon, there was an assumption that the number of links from a home page to other items should be fewer than 10. Their assumption was based on misapplying Miller's magic number, 7 ± 2. This assumption fails to recognize, however, that users do not need to remember the items, they need only to be able to identify them, which is far easier. A contrary position was that because recognition is easier than recall, it would be better to have a much larger number of links on the home page. This goes against a rule of thumb for information display on paper that advocates the use of white space to prevent confusion and an unpleasing, cluttered design. To solve this controversy Kevin Larson and Mary Czerwinski (1998) from Microsoft Research carried out an experiment and user satisfaction study. The following account outlines the main points of their study.

The goal of the study was to find the optimal depth versus breadth structure of hyperlinks for expertly categorized web content. Three conditions were tested using different link designs for the same web content. Each design had 512 bottom-level nodes.

Condition 1: $8 \times 8 \times 8$ (8 top-level categories, each with 8 sublevels, with 8 content-levels under each)

Condition 2: 16×32 (16 top-level categories, each with 32 content-level categories)

Condition 3: 32×16 (32 top-level categories, each with 16 content-levels categories)

These conditions were tested by 19 experienced web users, who each performed eight search tasks for each condition, making a total of 24 searches. The eight searches were selected for each participant at random from a bank of 128 possible target items, that were categorized according to content and complexity. Participants were given the same number of items from each category and no one searched for the same item more than once (i.e., there was no duplication of items across conditions).

Reaction times (RT) to complete each search were recorded and the average (Avg.) and standard deviation (SD) for each condition was computed. The results showed that on average participants completed search tasks fastest in the 16×32 hierarchy (Avg. RT = 36 seconds, SD = 16), second fastest in the 32×16 hierarchy (Avg. RT = 46 seconds, SD = 26), and slowest in the $8 \times 8 \times 8$ hierarchy (Avg. RT = 58 seconds, SD = 23). These results suggest that breadth is preferable to depth for searching web content. However, very large numbers of links on one page may be detrimental to searching performance.

ACTIVITY 14.5

(a) What were the independent and dependent variables in this study?

(b) Write two possible hypothesis statements.

(c) How would you categorize the experimental design?

(d) The participants are all described as "experts." Is this adequate? What else do you want to know about them?

(e) Comment on the description of the tasks. What else do you want to know?

(f) If you know some statistics, suggest what further analysis of the results should be done.

(g) Three other analyses were done on issues that were not mentioned in this description, but that anyone doing this experiment might have looked at. From your knowledge of interaction design, suggest what these analyses might be and say why.

(h) What are the implications of this study for web design?

Comment

(a) The independent variable is menu link structure. The dependent variable is reaction time to complete a search successfully.

(b) Web search performance is better with broad shallow link structures. There is no difference in search performance with different link structures.

(c) All the participants did all the tasks, so this is a same-participant design.

(d) "Expert" could refer to a broad range of expertise. The evaluators could have used a screening questionnaire to make sure that all the participants had reached a basic level of expertise and there were no super-experts in the group. However, given that all the participants did all the conditions, differences in expertise had less impact than in other experimental designs.

(e) Our excerpt contains very little description of the tasks. It would be good to see examples of typical tasks in each task category. How was the similarity and complexity of the tasks tested?

(f) A one-way analysis of variance was used to validate the significance of the main finding. Other tests are also discussed in the full paper.

(g) Participants could be asked to rate their preferences using a subjective rating questionnaire, which is similar to a user satisfaction questionnaire. The researchers also analyzed the paths the participants took to see if any of the conditions caused less optimal searching. They found that the condition with 32 items on the top-level caused a feeling of "lost in hyperspace," though this was not statistically significant. A less obvious analysis examined memory and scanning ability and found that better memory and scanning ability was associated with faster reaction time in the 16 × 32 hierarchy.

(h) Implications for web design are to avoid deep narrow link hierarchies and very broad shallow ones. However, as the authors emphasize, this is only one study and more research is needed before any generalizations can be made.

14.5 Predictive models

In contrast to the other forms of evaluation we have discussed, predictive models provide various measures of user performance without actually testing users.

This is especially useful in situations where it is difficult to do any user testing. For example, consider companies who want to upgrade their computer support for their employees. How do they decide which of the many possibilities is going to be the most effective and efficient for their needs? One way of helping them make their decision is to provide estimates about how different systems will fare for various kinds of task. Predictive modeling techniques have been designed to enable this.

The most well-known predictive modeling technique in human-computer interaction is GOMS. This is a generic term used to refer to a family of models, that vary in their granularity as to what aspects of a user's performance they model and make predictions about. These include the time it takes to perform tasks and the most effective strategies to use when performing tasks. The models have been used mainly to predict user performance when comparing different applications and devices. Below we describe two of the most well-known members of the GOMS family: the GOMS model and its "daughter," the keystroke level model.

14.5.1 The GOMS model

The GOMS model was developed in the early eighties by Stu Card, Tom Moran and Alan Newell (Card et al., 1983). As mentioned in Chapter 3, it was an attempt to model the knowledge and cognitive processes involved when users interact with systems. The term GOMS is an acronym which stands for ***goals, operators, methods and selection rules:***

- ***Goals*** refer to a particular state the user wants to achieve (e.g., find a website on interaction design).
- ***Operators*** refer to the cognitive processes and physical actions that need to be performed in order to attain those goals (e.g., decide on which search engine to use, think up and then enter keywords in search engine). The difference between a goal and an operator is that a goal is obtained and an operator is executed.
- ***Methods*** are learned procedures for accomplishing the goals. They consist of the exact sequence of steps required (e.g., drag mouse over entry field, type in keywords, press the "go" button).
- ***Selection rules*** are used to determine which method to select when there is more than one available for a given stage of a task. For example, once keywords have been entered into a search engine entry field, many search engines allow users to press the return key on the keyboard or click the "go" button using the mouse to progress the search. A selection rule would determine which of these two methods to use in the particular instance. Below is a detailed example of a GOMS model for deleting a word in a sentence using Microsoft Word.

Goal: delete a word in a sentence

Method for accomplishing goal of deleting a word using menu option:

Step 1. Recall that word to be deleted has to be highlighted
Step 2. Recall that command is "cut"
Step 3. Recall that command "cut" is in edit menu
Step 4. Accomplish goal of selecting and executing the "cut" command
Step 5. Return with goal accomplished

Method for accomplishing goal of deleting a word using delete key:

Step 1. Recall where to position cursor in relation to word to be deleted
Step 2. Recall which key is delete key
Step 3. Press "delete" key to delete each letter
Step 4. Return with goal accomplished

Operators to use in above methods:

Click mouse
Drag cursor over text
Select menu
Move cursor to command
Press keyboard key

Selection Rules to decide which method to use:

1: Delete text using mouse and selecting from menu if large amount of text is to be deleted
2: Delete text using delete key if small number of letters is to be deleted

14.5.2 The Keystroke level model

The keystroke level model differs from the GOMS model in that it provides actual numerical predictions of user performance. Tasks can be compared in terms of the time it takes to perform them when using different strategies. The main benefit of making these kinds of quantitative predictions is that different features of systems and applications can be easily compared to see which might be the most effective for performing specific kinds of tasks.

When developing the keystroke level model, Card et al. (1983) analyzed the findings of many empirical studies of actual user performance in order to derive a standard set of approximate times for the main kinds of operators used during a task. In so doing, they were able to come up with the average time it takes to carry out common physical actions (e.g., press a key, click on a mouse button) together with other aspects of user–computer interaction (e.g., the time it takes to decide what to do, the system response rate). Below are the core times they proposed for

these (note how much variability there is in the time it takes to press a key for users with different typing skills).

Operator name	Description	Time (sec)
K	Pressing a single key or button	0.35 (average)
	Skilled typist (55 wpm)	0.22
	Average typist (40 wpm)	0.28
	User unfamiliar with the keyboard	1.20
	Pressing shift or control key	*0.08*
P	Pointing with a mouse or other device to a target on a display	1.10
P_1	Clicking the mouse or similar device	0.20
H	Homing hands on the keyboard or other device	0.40
D	Draw a line using a mouse	Variable depending on the length of line
M	Mentally prepare to do something (e.g., make a decision)	1.35
R(t)	System response time--counted only if it causes the user to wait when carrying out their task	t

The predicted time it takes to execute a given task is then calculated by describing the sequence of actions involved and then summing together the approximate times that each one will take:

$$T_{execute} = T_K + T_P + T_H + T_D + T_M + T_R$$

For example, consider how long it would take to insert the word ***not*** into the following sentence, using a word processor like Microsoft Word:

Running through the streets naked is normal.
So that it becomes:
Running through the streets naked is not normal.

First we need to decide what the user will do. We are assuming that he will have read the sentences beforehand and so start our calculation at the point where he is about to carry out the requested task. To begin he will need to think what method to select. So we first note a mental event (M operator). Next he will need to move the cursor into the appropriate point of the sentence. So we note an H operator (i.e., reach for the mouse). The remaining sequence of operators are then: position the mouse before the word normal (P), click the mouse button (P_1), move hand from mouse over the keyboard ready to type (H), think about which letters to type (M), type the letters ***n, o*** and ***t*** (3K) and finally press the spacebar (K).

The times for each of these operators can then be worked out:

Mentally prepare (M)	1.35
Reach for the mouse (H)	0.40
Position mouse before the word "normal" (P)	1.10
Click mouse (P_1)	0.20
Move hands to home position on keys (H)	0.40
Mentally prepare (M)	1.35
Type "n" (good typist) (K)	0.22
Type "*o*" (K)	0.22
Type "*t*" (K)	0.22
Type "space" (K)	0.22
Total predicted time:	5.68 seconds

When there are many components to add up, it is often easier to put together all the same kinds of operators. For example, the above can be rewritten as:
$2(M) + 2(H) + 1(P) + 1(P_1) + 4\ (K) = 2.70 + 0.88 + 1.10 + 0.2 + 0.80 = 5.68$ seconds.

Over 5 seconds seems a long time to insert a word into a sentence, especially for a good typist. Having made our calculation it is useful to look back at the various decisions made. For example, we may want to think why we included a mental operator before typing the letters n, o and *t* but not one before any of the other physical actions. Was this necessary? Perhaps we don't need to include it. The decision when to include a time for mentally preparing for a physical action is one of the main difficulties with using the keystroke level model. Sometimes it is obvious when to include one (especially if the task requires making a decision) but for other times it can seem quite arbitrary. Another problem is that, just like typing skills vary between individuals, so too do the mental preparation times people spend thinking about what to do. Mental preparation can vary from under 0.5 of a second to well over a minute. Practice at modeling similar kinds of tasks together with comparing them with actual times taken can help overcome these problems. Ensuring that decisions are applied consistently also helps. For example, if comparisons between two prototypes are made, apply the same decisions to each.

ACTIVITY 14.6 As described in the GOMS model above there are two main ways words can be deleted in a sentence when using a word processor like Word. These are:

(a) deleting each letter of the word individually by using the delete key

(b) highlighting the word using the mouse and then deleting the highlighted section in one go

Which of the two methods do you think is quickest for deleting the word "not" from the following sentence:

I do not like using the keystroke level model.

Comment

(a) Our analysis for method 1 is:

Mentally prepare	M	1.35
Reach for mouse	H	0.40
Move cursor one space after the word "not"	P	1.10
Click mouse	P_1	0.20
Home in on delete key	H	0.40
Press delete key 4 times to remove word plus a space (using value for good typist value)	4(K)	0.88
Total predicted time = 4.33 seconds		

(b) Our analysis for method 2 is:

Mentally prepare	M	1.35
Reach for mouse	H	0.40
Move cursor to just before the word "not"	P	1.10
Click and hold mouse button down (half a P_1)	P_1	0.10
Drag the mouse across "not" and one space	P	1.10
Release the mouse button (half a P_1)	P_1	0.10
Home in on delete key	H	0.40
Press delete key (Using value for good typist rate)	K	0.22
Total predicted time = **4.77** seconds		

The result seems counter-intuitive. Why do you think this is? The reason is that the amount of time required to select the letters to be deleted is longer for the second method than pressing the delete key three times in the first method. If the word had been any longer, for example, "keystroke" then the keystroke analysis would have predicted the opposite. There are also other ways of deleting words, such as double clicking on the word (to select it) and then either pressing the delete key or the combination of ctrl+X keys. What do you think the keystroke level model would predict for either of these two methods?

14.5.3 Benefits and limitations of GOMS

One of the main attractions of the GOMS approach is that it allows comparative analyses to be performed for different interfaces or computer systems relatively easily. Since its inception, a number of researchers have used the method, reporting on its success for comparing the efficacy of different computer-based systems. The most well-known is Project Ernestine (Gray et al., 1993). This study was carried out to determine if a proposed new workstation, that was ergonomically designed, would improve telephone call operators' performance. Empirical data collected for a range of operator tasks using the existing system was compared with hypothetical data deduced from doing a GOMS analysis for the same set of tasks for the proposed new system.

Similar to the activity above, the outcome of the study was counter-intuitive. When comparing the GOMS predictions for the proposed system with the empirical data collected for the existing system, the researchers discovered that several tasks would take longer to accomplish. Moreover, their analysis was able to show why

this might be the case: certain keystrokes would need to be performed at critical times during a task rather than during slack periods (as was the case with the existing system). Thus, rather than carrying out these keystrokes in parallel when talking with a customer (as they did with the existing system) they would need to do them sequentially—hence the predicted increase in time spent on the overall task. This suggested to the researchers that, overall, the proposed system would actually slow down the operators rather than improve their performance. On the basis of this study, they were able to advise the phone company against purchasing the new workstations, saving them from investing in a potentially inefficient technology.

While this study has shown that GOMS can be useful in helping make decisions about the effectiveness of new products, it is not often used for evaluation purposes. Part of the problem is its highly limited scope: it can only really model computer-based tasks that involve a small set of highly routine data-entry type tasks. Furthermore, it is intended to be used only to predict expert performance, and does not allow for errors to be modeled. This makes it much more difficult (and sometimes impossible) to predict how an average user will carry out their tasks when using a range of systems, especially those that have been designed to be very flexible in the way they can be used. In most situations, it isn't possible to predict how users will perform. Many ***unpredictable*** factors come into play including individual differences among users, fatigue, mental workload, learning effects, and social and organizational factors. For example, most people do not carry out their tasks sequentially but will be constantly multi-tasking, dealing with interruptions and talking to others.

A dilemma with predictive models, therefore, is that they can only really make predictions about predictable behavior. Given that most people are unpredictable in the way they behave, it makes it difficult to use them as a way of evaluating how systems will be used in real-world contexts. They can, however, provide useful estimates for comparing the efficiency of different methods of completing tasks, particularly if the tasks are short and clearly defined.

14.5.4 Fitts' Law

Fitts' Law (1954) predicts the time it takes to reach a target using a pointing device. It was originally used in human factors research to model the relationship between speed and accuracy when moving towards a target on a display. In interaction design it has been used to describe the time it takes to point at a target, based on the size of the object and the distance to the object. Specifically, it is used to model the time it takes to use a mouse and other input devices to click on objects on a screen. One of its main benefits is that it can help designers decide where to locate buttons, what size they should be and how close together they should be on a screen display. The law states that:

$$T = k \log_2(D/S + 0.5), k = 100 \text{ msec}.$$

where

T = time to move the hand to a target

D = distance between hand and target

S = size of target

In a nutshell the bigger the target the easier and quicker it is to reach it. This is why interfaces that have big buttons are easier to use than interfaces that present lots of tiny buttons crammed together. Fitts' law also predicts that the most quickly accessed targets on any computer display are the four corners of the screen. This is because of their "pinning" action, i.e., the sides of the display constrain the user from over-stepping the target. However, as pointed out by Tog on his AskTog website, corners seem strangely to be avoided at all costs by designers.

Fitts' Law, therefore, can be useful for evaluating systems where the time to physically locate an object is critical to the task at hand. In particular it can help designers think about where to locate objects on the screen in relation to each other. This is especially useful for mobile devices, where there is limited space for placing icons and buttons on the screen. For example, in a recent study carried out by Nokia, Fitts' Law was used to predict expert text entry rates for several input methods on a 12-key mobile phone keypad. The study helped the designers make decisions about the size of keys, their positioning and the sequences of presses to perform common tasks for the mobile device. Trade-offs between the size of a device, and accuracy of using it were made with the help of calculations from this model.

ACTIVITY 14.7 Microsoft toolbars provide the user with the option of displaying a label below each tool. Give a reason why labeled tools may be accessed faster. (Assume that the user knows the tool and does not need the label to identify it.)

Comment The label becomes part of the target and hence the target gets bigger. As we mentioned earlier bigger targets can be accessed faster.

Furthermore, tool icons that don't have labels are likely to be placed closer together so they are more crowded. Spreading the icons further apart creates buffer zones of space around the icons so that if users accidentally go past the target they will be less likely to select the wrong icon. When the icons are crowded together the user is at greater risk of accidentally overshooting and selecting the wrong icon. The same is true of menus, where the items are closely bunched together.

Assignment

This assignment continues the work you did on the web-based ticketing system at the end of Chapters 7, 8, and 13. The aim of this assignment is again to evaluate the prototypes produced, but this time using user testing. You will then be able to compare the kind of results you got from the heuristic evaluation with those from the user testing. Even though you will be using different prototypes for each evaluation, you should be able to compare the types of problems that each technique reveals.

(a) Based on your knowledge of the requirements for this system, develop a standard task, e.g., booking two seats for a particular performance.

(b) Prepare a short informed consent form, and write an introduction that explains why you are testing this prototype.

(c) Select three typical users, who can be friends or colleagues, and ask them to do the task using your prototype.

(d) 'Note the problems that each user encounters. If you can, time their performance. (If you happen to have a video camera you could film each participant.)

(e) Did the kinds of problems that user testing revealed differ from those obtained from a heuristic evaluation? If so, in what ways?

(f) What are the main advantages and disadvantages of each technique?

Summary

This chapter described user testing, which is the core of usability testing. The various aspects of user testing were discussed, including setting up tests, collecting data, controlling conditions and analyzing findings. Experimental design and how experiments differ from user testing was also discussed.

Predicting user performance using the GOMS model, the keystroke level model, and Fitts' Law was presented. These techniques can be useful for determining whether a proposed interface, system or keypad layout will be optimal.

Key points

- User testing is a central component of usability testing which typically also includes observation, user satisfaction questionnaires and interviews.
- Testing is commonly done in controlled laboratory-like conditions, in contrast to field studies that focus on how the product is used in its natural context.
- Experiments aim to answer a question or hypothesis by manipulating certain variables while keeping others constant.
- The experimenter controls independent variable(s) in order to measure dependent variable(s).
- There are three types of experimental design: different participants, same participants, and matched pair participants.
- The GOMS model, keystroke-level model and Fitts' law can be used to predict expert, error-free performance for certain kinds of tasks.
- Predictive models require neither users nor experts, but the evaluators must be skilled in applying the models.
- Predictive models are used to evaluate systems with limited, clearly defined functionality such as data entry applications.

Further reading

DUMAS, J. S., AND REDISH, J. C. **(1999)** ***A Practical Guide to Usability Testing.*** Exeter, UK: Intellect. Many books have been written about user testing and usability, but this one is particularly useful because it describes the process in detail and provides many examples.

RUBIN, J. **(1994)** ***Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests.*** New York: John Wiley & Sons. This book also provides good practical advice about preparing and conducting user tests, analyzing and reporting the results.

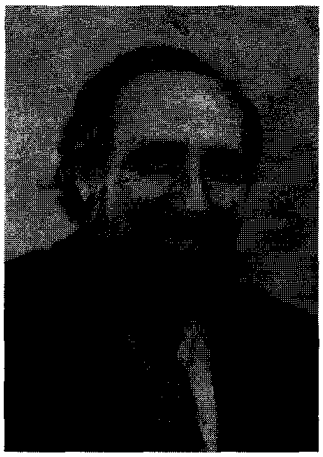
ROBSON, C. **(1994)** ***Experimental Design and Statistics in Psychology.*** Aylesbury, UK. Penguin Psychology. This book provides an introduction to experimental design and basic statistics.

LARSON, K., AND CZERWINSKI, M. **(1998)** ***Web page design: Implications of memory, structure and scent for information retrieval.*** Paper presented at CHI **98,** Los Angeles. This paper describes the breadth-versus-depth web study outlined in Box **14.2.**

CARD, S. K., MORAN, T. P., AND NEWELL, A. (1983) ***The Psychology of Human Computer Interaction.*** Hillsdale, *NJ:* Lawrence Erlbaum Associates. This seminal book describes GOMS and the keystroke level model.

MACKENZIE, I. S. **(1992)** Fitts' law as a research and design tool in human-computer interaction. ***Human-Computer Interaction, 7,*** 91–139. This early paper by Scott Mackenzie provides a detailed discussion of how Fitts' law can be used in HCI.

INTERVIEW with Ben Shneiderman



Ben Shneiderman is professor of computer science at University of Maryland, where he was founder and director of the Human-Computer Interaction Laboratory from 1983 to 2000. He is author of the highly acclaimed book *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, now in its third edition. He developed the concept of direct manipulation and created the user interface for the selectable text link that makes the web so easy to use.

JP: Ben you've been a strong advocate of measuring user performance and user satisfaction. Why is just watching users not enough?

BS: Watching users is a great way to begin, but if we are to develop a scientific foundation for HCI that promotes theory and supports prediction, measurement will be important. The purpose of measurement is not statistics but insight.

JP: OK can you give me an example?

BS: Watching users traverse a menu tree may reveal some problems they have, but only when you start to measure the time and number of branches taken can you discover that broader and shallower trees are almost always the winning strategy. This conflict between broader and shallower trees emerged in a conference panel discussion with a leading researcher for a major corporation. She and her colleagues followed up by testing users' speed of performance on searching tasks with two-level and three-level trees.

(Editor's note: You can read about this experiment in Box 14.2).

JP: But is speed of performance always the important measure?

BS: Measuring speed of performance, rate of errors, and user satisfaction separately is important because sometimes users may be satisfied by an elaborate graphical interface even if it slows them down substantially. Finding the right balance among performance, error rates, and user satisfaction depends on whether you are building a repetitive data-entry system, an air-traffic control system, or a game.

JP: Experiments are an important part of your undergraduate classes. Why?

BS: Most computer science and information systems students have had little exposure to experiments. I want to make sure that my students can form lucid and testable hypotheses that can be experimentally tested with groups of real users. They should understand about choosing a small number of independent variables to modify and dependent variables to measure. I believe that students benefit by understanding how to control for biases and perform statistical tests that confirm or refute the hypotheses. My students conduct experimental projects in teams and prepare their reports on the web. For example, one team did a project in which they varied the display size and demonstrated that web surfers found what they needed faster with larger screens. Another group found that bigger mouse pads do not increase speed of performance (www.otal.umd.edu/SHORE2000). Even if students never conduct an experiment professionally, the process of designing experiments helps them to become more effective analysts. I also want my students to be able to read scientific papers that report on experiments.

JP: What "take-away messages" do you want your students to get from taking an HCI class?

BS: I want my students to know about rigorous and replicable scientific results that form the foundation for this emerging discipline of human-computer interaction. Just as physics provides a scientific foundation for mechanical engineering, HCI provides a rigorous foundation for usability engineering.

JP: How do you distinguish between an experiment and usability testing?

BS: The best controlled experiments start with a hypothesis that has practical implications and theoretical results of widespread importance. A controlled experiment has at least two conditions and applies statistical tests such as t-test and analysis of variance (ANOVA) to verify statistically significant differences. The results confirm or refute the hypothesis

and the procedure is carefully described so that others can replicate it. I tell my students that experiments have two parents and three children. The parents are "a practical problem" and "a theoretical foundation" and the three children are "help in resolving the practical problem," "refinements to the theory," and "advice to future experimenters who work on the same problem."

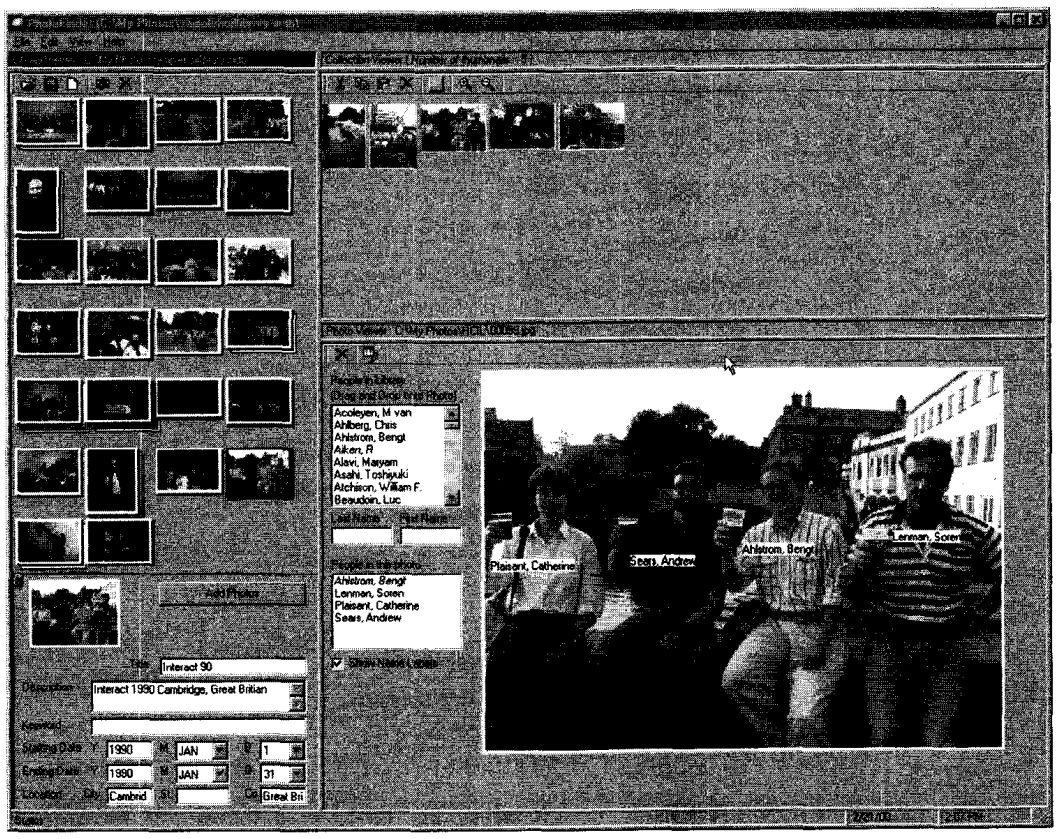
By contrast, a usability test studies a small number of users who carry out required tasks. Statistical results are less important. The goal is to refine a product as quickly as possible. The outcome of a usability test is a report to developers that identifies frequent problems and possibly suggests improvements, maybe ranked from high to low priority and from low to high developer effort.

JP: What do you see as the important usability issues for the next five years?

BS: I see three directions for the next five years. The first is the shift from emphasizing the technology to focusing on user needs. I like to say "the old computing is about what computers can do, the new computing is about what users can do."

JP: But hasn't HCI always been about what users can do?

BS: Yes, but HCI and usability engineering have been more evaluative than generative. To clarify, I believe that deeper theories about human needs will contribute to innovations in mobility, ubiquity, and community. Information and communication tools will become pervasive and enable higher levels of social interaction. For example, museum visitors to the Louvre, white-water rafters in Colorado, or family travelers to Hawaii's Haleakala volcano will be able to point at a sculpture, rock, or flower and find out about it. They'll be able to see photos at different seasons taken by previous visitors and send their own pictures back to friends and grandparents. One of our projects allows people to accumulate, organize, and retrieve the many photos that they will take and receive. Users of our PhotoFinder software tool can organize their photos and annotate them by dragging



and dropping name labels. Then they can find photos of people and events to tell stories and reminisce (see figure).

HCI researchers who understand human needs are likely to come up with innovations that help physicians to make better diagnoses, enable shoppers to find what they want at fair prices, and allow educators to create more compelling experiences for students.

JP: What are the other two directions?

BS: The second opportunity is to support universal usability, thereby bringing the benefits of information and communications technology to the widest possible set of users. website designers will need to learn how to attract and retain a broad set of users with divergent needs and differing skills. They will have to understand how to accommodate users efficiently with slow and fast network connections, new and old computers, and various software platforms. System designers who invent strategies to accommodate young and old, novice and expert, and users with varying disabilities will earn the appreciation of users and the respect of their colleagues. Evidence is accumulating that designs that facilitate multiple natural-language versions of a website also make it easy to accommodate end-user customization, convert to wireless applications, support disabled users and speed modifications. The good news is that satisfying these multiple requirements also produces interfaces that are better for all users. Diversity promotes quality.

The third direction is the development of tools to let more people be more creative more of the time. Word processors, painting tools and music-composition software are a good starting point, but creative people need more powerful tools so that they can explore alternative solutions rapidly. Creativity-support tools will speed search of existing solutions, facilitate consultations with peers and mentors, and record the users' history of activity so that they can review or revise their work.

But remember that every positive development also has a potential dark side. One of the formidable challenges for HCI students is to think carefully about how to cope with the unexpected and unintended. Powerful tools can have dangerous consequences.