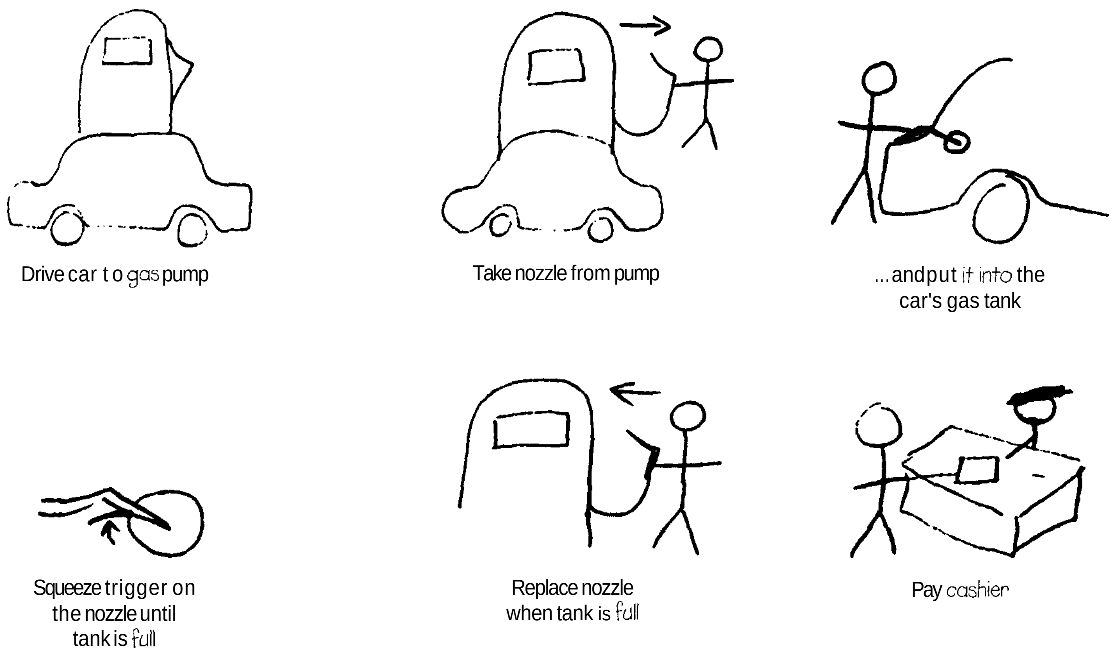


Figure 8.4 A storyboard depicting how to fill a car with gas.

Wizard **of** Oz Another low-fidelity prototyping method called Wizard of Oz assumes that you have a software-based prototype. In this technique, the user sits at a computer screen and interacts with the software as though interacting with the product. In fact, however, the computer is connected to another machine where a human operator sits and simulates the software's response to the user. The method takes its name from the classic story of the little girl who is swept away in a storm and finds herself in the Land of Oz (Baum and Denslow, 1900).

8.2.4 High-fidelity prototyping

High-fidelity prototyping uses materials that you would expect to be in the final product and produces a prototype that looks much more like the final thing. For example, a prototype of a software system developed in Visual Basic is higher fidelity than a paper-based mockup; a molded piece of plastic with a dummy keyboard is a higher-fidelity prototype of the PalmPilot than the lump of wood.

If you are to build a prototype in software, then clearly you need a software tool to support this. Common prototyping tools include Macromedia Director, Visual Basic, and Smalltalk. These are also full-fledged development environments, so they are powerful tools, but building prototypes using them can also be very straightforward.

Table 8.1 Relative effectiveness of low- vs. high-fidelity prototypes (Rudd et al., 1996)

Type	Advantages	Disadvantages
Low-fidelity prototype	• Lower development cost. • Evaluate multiple design concepts. • Useful communication device. • Address screen layout issues. • Useful for identifying market requirements. • Proof-of-concept.	• Limited error checking. • Poor detailed specification to code to. • Facilitator-driven. • Limited utility after requirements established. • Limited usefulness for usability tests. • Navigational and flow limitations.
High-fidelity prototype	• Complete functionality. • Fully interactive. • User-driven. • Clearly defines navigational scheme. • Use for exploration and test. • Look and feel of final product. • Serves as a living specification. • Marketing and sales tool.	• More expensive to develop. • Time-consuming to create. • Inefficient for proof-of-concept designs. • Not effective for requirements gathering.

Marc Rettig (1994) argues that more projects should use low-fidelity prototyping because of the inherent problems with high-fidelity prototyping. He identifies these problems as:

- They take too long to build.
- Reviewers and testers tend to comment on superficial aspects rather than content.
- Developers are reluctant to change something they have crafted for hours.
- A software prototype can set expectations too high.
- Just one bug in a high-fidelity prototype can bring the testing to a halt.

High-fidelity prototyping is useful for selling ideas to people and for testing out technical issues. However, the use of paper prototyping and other ideas should be actively encouraged for exploring issues of content and structure. Further advantages and disadvantages of the two types of prototyping are listed in Table 8.1.

8.2.5 Compromises in protoiyping

By their very nature, prototypes involve compromises: the intention is to produce something quickly to test an aspect of the product. The kind of questions or choices

BOX 8.1 Prototyping Cultures *(Schrage, 1996)*

"The culture of an organization has a strong influence on the quality of the innovations that the organization can produce." (Schrage, 1996, p. 193)

This observation is drawn mainly from product-related organizations, but also applies to software development. There are primarily two kinds of organizational culture for innovation: the specification culture and the prototyping culture. In the former, new products and development are driven by written specifications, i.e., by a collection of documented requirements. In the latter, understanding requirements and developing the new product are driven by prototyping. Large companies such as IBM or AT&T that have to gather and coordinate a large amount of information tend to be specification-driven, while smaller entrepreneurial companies tend to be prototype-driven. Both approaches have potential disadvantages. A carefully prepared specification may prove completely infeasible once prototyping begins. Similarly, a wonderful prototype may prove to be too expensive to produce on a large scale.

The medium used for developing the prototype affects the process itself. Schrage puts forward the example of General Motors, which used to produce clay prototypes of new cars and then try to capture these in CAD tools. On the other hand, Toyota designs its cars using CAD tools first and produces a clay prototype once the design has stabilized. The medium used also determines in part the questions that a prototype can answer. As a simple example, a horizontal software prototype will not be able to answer questions about the detailed operation of a function since it is not designed to model that level of detail.

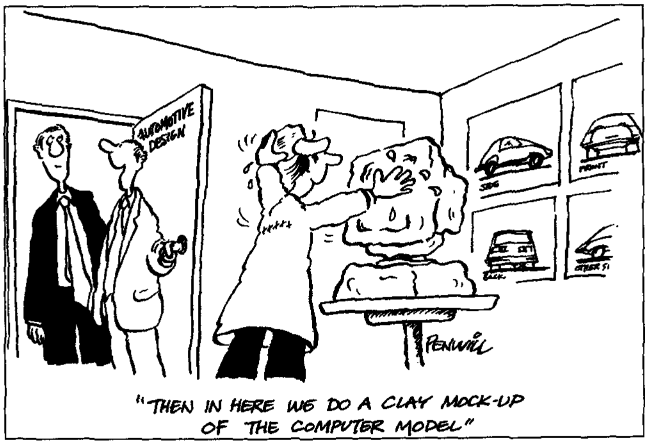
The speed of prototype development and the time between prototyping iterations is often a product of organizational culture and tradition. Some companies have a set number of prototypes embodied in their development method and use this number irrespective of the technical needs of any particular product. Generally speaking, the more prototyping cycles there are, the more polished the final product will be.

The corporate prototyping culture is most starkly revealed by who is involved in prototyping and when. For example, who owns the prototype? Is there a special prototyping department? Who gets to see and evaluate the prototype? Sometimes designers are happy to show emerging prototypes to their peers but not to managers, for fear of being misunderstood and either having the project cancelled or finding an order to ship the prototype when it is not ready. Prototype demonstrations to senior managers often happen too late in the development cycle to have any real impact because of these fears.

David Kelley (Schrage, p.195) claims that organizations wanting to be innovative need to move to a prototype-driven culture. Schrage sees that there are two cultural aspects to this shift. First, scheduled prototyping cycles that force designers to build many prototypes are more likely to lead to a prototype-driven culture than allowing designers to produce *ad hoc* prototypes when they think it appropriate. Second, rather than innovative teams being needed for innovative prototypes, it is now recognized that innovative prototypes lead to innovating teams! This can be especially significant when the teams are cross-functional, i.e., multidisciplinary.

that any one prototype allows the designer to answer is therefore limited, and the prototype must be designed and built with the key issues in mind. In low-fidelity prototyping, it is fairly clear that compromises have been made. For example, with a paper-based prototype an obvious compromise is that the device doesn't actually work! For software-based prototyping, some of the compromises will still be fairly clear; for example, the response speed may be slow, or the exact icons may be sketchy, or only a limited amount of functionality may be available.

Two common compromises that often must be traded against each other are breadth of functionality provided versus depth. These two kinds of prototyping



are called horizontal prototyping (providing a wide range of functions but with little detail) and vertical prototyping (providing a lot of detail for only a few functions).

Other compromises won't be obvious to a user of the system. For example, the internal structure of the system may not have been carefully designed, and the prototype may contain "spaghetti code" or may be badly partitioned. One of the dangers of producing running prototypes, i.e., ones that users can interact with automatically, is that they may believe that the prototype is the system. The danger for developers is that it may lead them to consider fewer alternatives because they have found one that works and that the users like. However, the compromises made in order to produce the prototype must not be ignored, particularly the ones that are less obvious from the outside. We still must produce a good-quality system and good engineering principles must be adhered to.

8.2.6 Construction: from design to implementation

When the design has been around the iteration cycle enough times to feel confident that it fits requirements, everything that has been learned through the iterated steps of prototyping and evaluation must be integrated to produce the final product.

Although prototypes will have undergone extensive user evaluation, they will not necessarily have been subjected to rigorous quality testing for other characteristics such as robustness and error-free operation. Constructing a product to be used by thousands or millions of people running on various platforms and under a wide range of circumstances requires a different testing regime than producing a quick prototype to answer specific questions.

The dilemma box below discusses two different development philosophies. One approach, called evolutionary prototyping, involves evolving a prototype into the final product. An alternative approach, called throwaway prototyping, uses the prototypes as stepping stones towards the final design. In this case, the

DILEMMA **Prototyping to Throw Away**

Low-fidelity prototypes are never intended to be kept and integrated into the final product. But when building a software-based system, developers can choose to do one of two things: either build a prototype with the intention of throwing it away after it has fulfilled its immediate purpose, or build a prototype with the intention of evolving it into the final product.

Above, we talked about the compromises made when producing a prototype, and we commented that the "invisible" compromises, concerned with the structure of the underlying software must not be ignored. However, when a project team is under pressure to produce the final product and a complex prototype exists that fulfills many of the requirements, or maybe a set of vertical prototypes exists that together fulfill the requirements, it can become very tempting to pull them together and issue the result as the final product. After all, many hours of development have probably gone into developing the prototypes, and evaluation with the client has gone well, so isn't it a waste to throw it all away? Basing the final product on prototypes in this way will simply store up testing and maintenance problems for later on: in short, this is likely to compromise the quality of the product.

Evolving the final prototype into the final product through a defined process of evolutionary prototyping can lead to a robust final product, but this must be clearly planned and designed for from the beginning. Building directly on prototypes that have been used to answer specific questions through the development process will not yield a robust product. As Constantine and Lockwood (1999) observe, "Software is the only engineering field that throws together prototypes and then attempts to sell them as delivered goods".

On the other hand, if your device is an innovation, then being first to market with a "good enough" product may be more important for securing your market position than having a very high-quality product that reaches the market two months after your competitors'.

prototypes are thrown away and the final product is built from scratch. If an evolutionary prototyping approach is to be taken, the prototypes should be subjected to rigorous testing along the way; for throw-away prototyping such testing is not necessary.

8.3 Conceptual design: moving from requirements to first design

Conceptual design is concerned with transforming the user requirements and needs into a conceptual model. Conceptual models were introduced in Chapter 2, and here we provide more detail and discuss how to go about developing one. We defined conceptual model as "a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave, and look like, that will be understandable by the users in the manner intended." The basis for designing this model is the set of user tasks the product will support. There is no easy transformation to apply to a set of requirements data that will produce "the best" or even a "good enough" conceptual model. Steeping yourself in the data and trying to empathize with the users while considering the issues raised in this section is one of the best ways to proceed. From the requirements and this experience, a picture of what you want the users' experience to be when using the new product will emerge.

Beyer and Holtzblatt (1998), in their method ***Contextual Design*** discussed in Chapter 9, recommend holding review meetings within the team to get different peoples' perspectives on the data and what they observed. This helps to deepen understanding and to expose the whole team to different aspects. Ideas will emerge as this extended understanding of the requirements is established, and these can be tested against other data and scenarios, discussed with other design team members and prototyped for testing with users. Other ways to understand the users' experience are described in Box 8.2.

Ideas for a conceptual model may emerge during data gathering, but remember what Suzanne Robertson said in her interview at the end of Chapter **7:** you must separate the real requirements from solution ideas.

Key guiding principles of conceptual design are:

- Keep an open mind but never forget the users and their context.
- Discuss ideas with other stakeholders as much as possible.
- Use low-fidelity prototyping to get rapid feedback.
- Iterate, iterate, and iterate. Remember Fudd's first law of creativity: "To get a good idea, get lots of ideas" (Rettig, 1994).

Considering alternatives and repeatedly thinking about different perspectives helps to expand the solution space and can help prompt insights. Prototyping (introduced in Section 8.2) and scenarios (introduced in Chapter **7)** are two techniques to help you explore ideas and make design decisions. But before explaining how these can help, we need to explore in more detail how to go about envisioning the product.

8.3.1 Three perspectives for developing a conceptual model

Chapter 2 introduced three ways of thinking about a conceptual model: Which interaction mode would best support the users' activities? Is there a suitable interface metaphor to help users understand the product? Which interaction paradigm will the product follow? In this section, we discuss each of these in more detail. In all the discussions that follow, we are not suggesting that one way of approaching a conceptual design is right for one situation and wrong for another; they all provide different ways of thinking about the product and hence aid in generating alternatives.

Which interaction mode? Which interaction mode is most suitable for the product depends on the activities the user will engage in while using it. This information is identified through the requirements activity. The interaction mode refers to how the user invokes actions when interacting with the device. In Chapter 2 we introduced two different types of interaction mode: those based on activities and those based on objects. For those based on activities, we introduced four general styles: instructing, conversing, manipulating and navigating, and exploring and browsing. Which is best suited to your current design depends on the application domain and the kind of system being developed. For example, a computer game is most likely to suit a manipulating and navigating style, while a drawing package has aspects of instructing and conversing.

BOX 8.2 How to Really Understand the Users' Experience

Some design teams go to great lengths to ensure that they come to empathize with, not just understand, the users' experience. We know from learning things ourselves that "learning by doing" is more effective than being told something or just seeing something. Buchenau and Suri (2000) describe an approach they call experience prototyping that is intended to give designers some of the insight into a user's experience that comes only from first-hand knowledge. For example, they describe a team designing a chest-implanted automatic defibrillator. A defibrillator is used with victims of cardiac arrest when their heart muscle goes into a chaotic arrythmia and fails to pump blood, a state called fibrillation. A defibrillator delivers an electric shock to the heart, often through paddle electrodes applied externally through the chest wall; an implanted defibrillator does this through leads that connect directly to the heart muscle. In either case, it's a big electric shock intended to restore the heart muscle to its regular rhythm that can be powerful enough to knock people off their feet.

This kind of event is completely outside most people's experience, and so it is difficult really to understand what the user's experience is likely to be for this kind of device. You can't fit a prototype pacemaker to each member of the design team and simulate fibrillation in them! This makes it difficult for designers to gain the insight they need. However, you can simulate some critical aspects of the experience, one of which is the random occurrence of a defibrillating shock. To achieve this, each team member was given a pager to take home over the weekend (elements of the pack are shown in Figure 8.5). The pager message simulated the occurrence of a defibrillating shock. Messages were sent at random, and team members were asked to record where they were, who they were with, what they were doing, and what they thought and felt knowing that this represented a shock. Experiences were shared the following week, and example insights ranged from anxiety around everyday happenings such as holding a child and operating power tools, to being in social situations at a loss how to communicate to onlookers what was happening. This first-hand experience brought new insights to the design effort.

Another instance in which designers tried hard to come to terms with the user experience is the Third Age suit, developed at ICE, Loughborough University (see Figure 8.6). This suit was designed so that car designers could experience what it

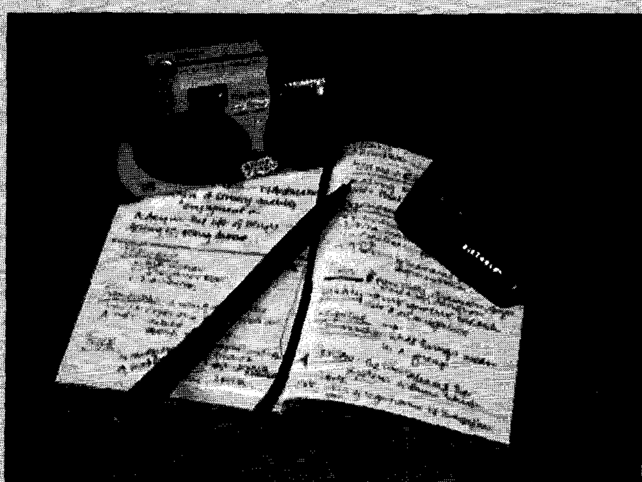


Figure 8.5 The patient kit for experience prototyping.

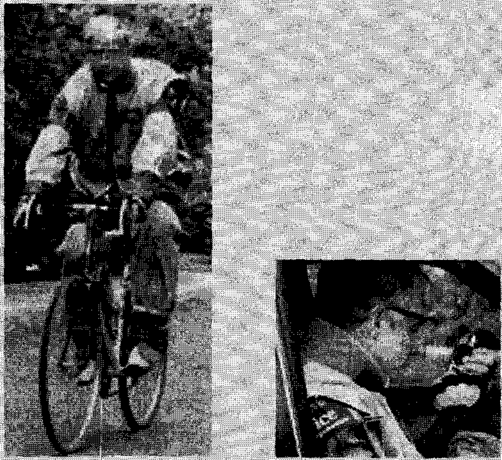


Figure 8.6 The Third Age suit: (a) riding a bike and (b) using a mobile phone.

might be like to be in an older body. The suit restricts movement in the neck, arms, legs, and ankles in a way that simulates the mobility problems typically experienced by someone over 55 years of age. For example, when operating the foot pedals in a car, many "third agers" (as they are called), lack the flexibility in their ankles to be able to rest their heel on the floor and operate the pedals by flexing their ankle. Thus they have to lift their whole foot up and push it down each time they operate the pedal, which puts much more stress on their leg muscles.

Most conceptual models will be a combination of modes, and it is necessary to associate different parts of the interaction with different modes. For example, consider the shared calendar example introduced in Chapter 7. One of the user tasks is finding out what is happening on a particular day. In this instance, instructing is an appropriate mode of interaction. No dialog is necessary for the system to show the required information. On the other hand, the user task of trying to arrange a meeting among a set of people may be conducted more like a conversation. We can imagine that the user begins by selecting the people for the meeting and setting some constraints on the arrangements such as time limit, urgency, length of meeting, etc. Then the system might respond with a set of possible times and dates for the user to select. This is much more like a conversation. (You may like to refer back to the scenario of this task in Chapter 7 and consider how well it matches this interaction mode.) For the task of planning, the user is likely to want to scan through pages and browse the days.

ACTIVITY 8.2 Consider the library catalog system introduced in Chapter 7. Identify tasks associated with this product that would be best supported by each of the interaction modes instructing, conversing, manipulating and navigating, and exploring and browsing.

Comment Here are some suggestions. You may have identified others:

(a) Instructing: the user wants to see details of a particular book, such as publisher and location.

(b) Conversing: the user wants to identify a book on a particular topic but doesn't know exactly what is required.

(c) Manipulating and navigating: the library books could be represented as icons that could be interrogated for information or manipulated to represent the book being reserved or borrowed.
(d) Exploring and browsing: the user is looking for interesting books, with no particular topic or author in mind.

Models based on objects provide a different perspective since they are structured around real-world objects. For example, the shared calendar system can be thought of as an electronic version of a paper calendar, which is a book kept by each person on their desk or in their bag. Alternatively, it could be thought of as a planner, a large flat piece of paper that is often pinned up on the wall in offices and is far more public. The choice of which objects to choose as a basis for the conceptual model is related to the choice of interface metaphor, which we consider below.

Mayhew (1999) identifies a similar distinction between conceptual models: process-oriented or product-oriented. The former kind of model best fits "an application in which there are no clearly identifiable primary work products. In these applications the main point is to support some work process." Examples of this might be software to control a chemical processing plant, a financial management package, or a customer care call-center. On the other hand, a product-oriented model "will best fit an application in which there are clear, identifiable work products that users individually create, modify and maintain." Examples of this are Microsoft products such as Excel, Powerpoint, Word, etc. More information about these kinds of conceptual model is given in Box 8.3.

Is there a suitable interface *metaphor?* Interface metaphors are another way to think about conceptual models. They are intended to combine familiar knowledge with new knowledge in a way that will help the user understand the system. Choosing suitable metaphors and combining new and familiar concepts requires a careful balance and is based on a sound understanding of the users and their context. For example, consider an educational system to teach six-year-olds mathematics. You could use the metaphor of a classroom with a teacher standing at the blackboard. But if you consider the users of the system and what is likely to engage them, you will be more likely to choose a metaphor that reminds the children of something they enjoy, such as a ball game, the circus, a playroom, etc.

Erickson (1990) suggests a three-step process for choosing a good interface metaphor. The first step is to understand what the system will do. Identifying functional requirements was discussed in Chapter 7. Developing partial conceptual models and trying them out may be part of the process. The second step is to understand which bits of the system are likely to cause users problems. Another way of looking at this is to identify which tasks or subtasks cause problems, are complicated, or are critical. A metaphor is only a partial mapping between the software and the real thing upon which the metaphor is based. Understanding areas in which users are likely to have difficulties means that the metaphor can be chosen to support those aspects. The third step is to generate metaphors. Looking for metaphors in the users' description of the tasks is a good starting point. Also, any

BOX 8.3 Process-Oriented versus Product-Oriented Conceptual Models *(Mayhew, 1999)*

Mayhew (1999) characterizes conceptual models in terms of their focus on products or on process. This is similar to our characterization of models that focus on objects or ones that focus on activities.

The difference between these two kinds of conceptual model is the drivers for the design activity. For a product-oriented system, the main products and the tools needed to create them form the main structure of the application. For a process-oriented application, it is the list of process steps that forms the system's basis. Mayhew suggests the following issues must be addressed during conceptual design, whether the application is primarily product-oriented or process-oriented:

- Products or processes must be clearly identified. For example, what documents are to be generated and what other tools are required to produce them? In a process-oriented model, what processes are to be supported?
- A set of presentation rules must be designed. For example, urgent tasks must be placed on the desktop, while less urgent tasks may be accessible through the menu bar. If designing for a GUI, design rules and guidelines come with the particular platform (see Box 8.5 below).
- Design a set of rules for how windows will be used.
- Identify how major information and functionality will be divided across displays.
- Define and design major navigational pathways. This will draw on the task analysis earlier, and leads to a structure for the tasks. Don't over-constrain users, make navigation easy, and provide facilities so that they always know where they are.
- Document alternative conceptual design models in sketches and explanatory notes.

An example conceptual model based on this approach is shown in Figure 8.7. This is a process-based model, and so it is structured around the processes and subprocesses the system is to support.

UTILITY CUSTOMER SERVICE

- Service Requests
  - Change Service
  - Cancel Service
  - Install New Service
    - Add Customer Info
    - Select Service
    - Check Credit
    - Quote Rate
    - Schedule Install
- Billing Questions
  - View Bill
  - Change Bill
  - Sales—Offer Service Options
- Maintenance Requests
  - Report Problem
  - Schedule Maintenance
  - View Maintenance History
- Info Requests
  - Show Services
  - Show Products
  - Show Installation Procedures

Figure 8.7 An example of a process-oriented conceptual model.

metaphors used in the application domain with which the users may be familiar may be suitable.

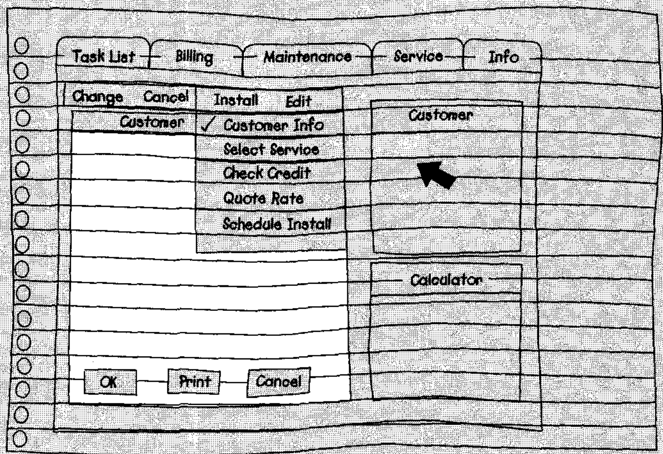
When suitable metaphors have been generated, they need to be evaluated. Again, Erickson (1990) suggests five questions to ask.

1. How much structure does the metaphor provide? A good metaphor will require structure, and preferrably familiar structure.

Here is a possible design presenting these work processes through a process-oriented Conceptual Model:

Conceptual Model—Utility Customer Service

Application Window



The presentation rules followed in this design are as follows (this design is not complete; it simply provides some examples of components of a process-oriented model).

The overall application is represented as a tab metaphor. *Highest-level processes* (e.g., billing questions, maintenance requests) are represented by tabs, and each tab represents a work space for that process. The tabbed work space includes a main window where that process is carried out, plus two "common windows," where tools common across all highest-level processes are maintained.

Second-level subprocesses are represented by selections in the menu bar within each tab, and *third-level subprocesses* by selections in pull-downs from the menu bar.

Structured subprocesses are controlled through dimming of subprocesses in pull-downs (until earlier subprocesses are completed, later subprocesses are dimmed out and unselectable).

Completed subprocesses are designated with a check mark.

Common activities available across highest-level processes (i.e., Customer, Calculator) are presented as separate, dedicated windows within the tabbed work spaces.

All windows are dialog boxes—that is, they cannot be minimized. They are all unresizable and unscrollable, but are movable and modeless. The main dialog box represents subprocesses, and it changes contents as the user moves through the subprocesses in sequence to complete a given subprocess and process.

Different windows have different background colors. Note that the active tabbed workspace itself is dark gray, the main subprocess dialog box is white, and all common dialog boxes are light gray.

Figure 8.7 (*continued*).

2. How much of the metaphor is relevant to the problem? One of the difficulties of using metaphors is that users may think they understand more than they do and start applying inappropriate elements of the metaphor to the system, leading to confusion or false expectations.
3. Is the interface metaphor easy to represent? **A** good metaphor will be associated with particular visual and audio elements, as well as words.

4. Will your audience understand the metaphor?
5. How extensible is the metaphor? Does it have extra aspects that may be useful later on?

In the calendar system, one obvious metaphor we could use is the individual's paper-based calendar. This is familiar to everyone, and we could combine that familiarity with facilities suitable for an electronic document such as hyperlinks and searching. Having thought of this metaphor, we need to apply the five questions listed above.

1. Does it supply structure? Yes, it supplies structure based on the familiar paper-based calendar. However, it does not supply structure for the notion of sharing information, i.e., other people looking in the calendar, because of two issues: first, an individual's calendar is very personal, and second, even if there is a paper-based calendar for a set of people, it can be closed and the information hidden from casual observers.
2. How much of the metaphor is relevant i.e., how many properties of the paper-based calendar are applicable to the electronic version? Well, in the electronic version it isn't appropriate to think of physically turning pages, but then a facility for looking at one "page" after another is required. The individual's calendar can be carried around from place to place. Whether or not we want to encourage that aspect of the metaphor depends on the kind of interaction paradigm we might consider. Finally, this is a shared calendar, and normally our personal calendars are not shared.
3. Is the metaphor easy to represent? Yes.
4. Will your audience understand the metaphor? Yes.
5. How extensible is the metaphor? The functionality of a paper-based calendar is fairly limited. However, it is also a book, and we could borrow facilities from electronic books (which are also familiar objects to most of our audience), so yes, it can be extended.

ACTIVITY 8.3 Another possible interface metaphor for the shared calendar system is the wall planner. Ask the five questions above of this metaphor.

Comment

(a) Does it supply structure? Yes, it supplies structure based on the wall-planner. This metaphor embodies the notion of public access more than the paper-based calendar. In particular, the wall planner is never "closed" to those who are near it.

(b) How much of the metaphor is relevant? Most of this metaphor is relevant. Individuals don't walk around with the wall planner, though, so the answer depends on how the calendar is to be used.

(c) Is the metaphor easy to represent? Yes, it could be represented as a spreadsheet.

(d) Will your audience understand the metaphor? Yes.

(e) How extensible is the metaphor? The functionality of a wall planner is also fairly limited. There are no obvious ways in which to extend the metaphor to help with this application.

Which interaction paradigm? Interaction paradigms are design philosophies that help you think about the product being developed. Interaction paradigms include the now traditional desktop paradigm, with WIMP interface (windows, icons, menus and pointers), ubiquitous computing, pervasive computing, wearable computing, tangible bits, attentive environments, and the Workaday World. Thinking about the user tasks with these different paradigms in mind can help provide insight both to choose the interaction paradigm and to inspire a different perspective on the problem.

Thinking about environmental requirements is particularly relevant when considering interaction paradigms. For example, consider the shared calendar in the context of the following paradigms:

- ***Ubiquitous computing.*** Combining some of our earlier discussions, we could perhaps imagine the shared calendar as being like a planner on the wall, but in an electronic form with which people could interact.

  Pervasive computing. Carrying around our own copy of the shared calendar builds directly upon current expectations and experience of personal calendars. We can imagine a system that allows individuals to keep a copy of the system on their own palmtop computers or PDAs, while also being linked to a central server somewhere that allows access to other information that is shared.
- ***Wearable computing.*** Imagine having an earring or a tie pin telling you that you have an appointment in an hour's time at a client's office and that you need to book a taxi? Or maybe asking you whether it is all right to book a meeting with your colleague on a particular date. What other possibilities can this model conjure up?

ACTIVITY 8.4 Consider the library catalog system and think about each of the paradigms listed above. Choose two of them and suggest different kinds of interaction that these paradigms imply.

Comment We had the following thoughts, but you may have others. The library catalog is likely to be used only in certain places, such as the library or perhaps in an office. The idea of wearable computers is not as attractive in this situation as pervasive computing would be, since people would have to put on the wearable when they arrived at the library. Alternatively, the library system might be designed to "cut in" on an existing wearable. Both of these solutions seem a little intrusive. Pervasive computing, on the other hand, would allow users to interact with the catalog wherever in the library they were, rather than having to go to a place where the PC or card catalog sits. You could possibly have digital books at the end of each library shelf that gave access to the catalog.

8.3.2 Expanding the conceptual model

Considering the issues in the previous section helps the designer to envision a product. These ideas must be thought through in more detail before being prototyped or tested with users. One aspect that will need to be decided is what technologies to use, e.g., mutimedia, virtual reality, or web-based materials, and what input and output devices best suit the situation, e.g., pen-based, touch screen, speech, keyboard, and so on. These decisions will depend on the constraints on the system, arising from the requirements you have established. For example, input and output devices will be influenced particularly by user and environmental requirements.

You also have to decide what concepts need to be communicated between the user and the product and how they are to be structured, related, and presented. This means deciding which functions the product will support, how those functions are related, and what information is required to support them. Although these decisions must be made, remember that they are made only tentatively to begin with and may change after prototyping and evaluation.

What functions will the product perform? Understanding the tasks the product will support is a fundamental aspect of developing the conceptual model, but it is also important to consider more specifically what functions the product will perform, i.e., how the task will be divided up between the human and the machine. For example, in the shared calendar example, the system may suggest dates when a set of people are able to meet, but is that as far as it should go? Should it automatically book the dates, or should it email the people concerned informing them of the meeting or asking if this is acceptable? Or is the human user or the meeting attendee responsible for checking this out? Developing scenarios, essential use cases, and use cases for the system will help clarify the answers to these questions. Deciding what the system will do and what must be left for the user is sometimes called ***task*** allocation. The trade-off between what to hand over to the device and what to keep in the control of the user has cognitive implications (see Chapter 3), and is linked to social aspects of collaboration (see Chapter 4). An example relating to our shared calendar system was discussed in Box 4.2 of Chapter 4: should the system allow users to book meetings in others' calendars without asking their consent first? In addition, if the cognitive load is too high for the user, then the device may be too stressful to use. On the other hand, if the device takes on too much and is too inflexible, then it may not be used at all.

Another aspect concerns the functions the hardware will perform, i.e., what functions will be hard-wired into the device and what will be left under software control, and thereby possibly indirectly in the control of the human dser? This leads to considerations of the architecture of the device, although you Would riot expect necessarily to have a clear architectural design at this stage of development.

How are the functions related to each other? Functions may be related temporally, e.g., one must be performed before another, or two can be performed in parallel. They may also be related through any number of possible categorizations, e.g., all functions relating to telephone memory storage in a cell phone, or all options for accessing files in a word processor. The relationships between tasks may constrain use or may indicate suitable task structures within the device. For example, if a task is dependent on completion of another task, then you may want to restrict the user to performing the tasks in strict order. An instance in which this has been put into practice is in some CASE (Computer-Aided Software Engineering) tools designed to support a specific development approach. Often these tools will insist that certain diagrams must be drawn before others. For example, in object-oriented software development you normally draw class diagrams before sequence diagrams, and some tools do not allow you to draw a sequence diagram until the relevant class diagram is in place. If you're working on a small project that doesn't require this kind of discipline, this can be very frustrating, but from the perspective of a manager in charge of a large project, having these restrictions in place may be advantageous.

If task analysis has been performed on relevant tasks, the breakdown will support these kinds of decisions. For example, in the shared calendar example, the task analysis performed in Section 7.1 shows the subtasks involved and the order in which the subtasks can be performed. Thus, the system could allow meeting constraints to be found before or after the list of people, and the potential dates could be identified in the individuals' calendars before checking with the departmental calendar. It is, however, important to get both the list of attendees and meeting constraints before looking for potential dates.

What information needs to be available? What data is required to perform a task? How is this data to be transformed by the system? Data is one of the categories of requirements we aim to identify and capture through the requirements activity. During conceptual design, we need to consider the information requirements and ensure that our model caters for the necessary data and that information is available as required to perform the task. Detailed issues of structure and display, such as whether to use an analog display or a digital display, will more likely be dealt with in the later, physical design activity, but implications arising from the type of data to be displayed may impact conceptual design issues.

For example, in the task of booking a meeting among a set of people using the shared calendar, the system needs to be told who is to be at the meeting, how long the meeting is to take, what its location should be, and what is the latest date on which the meeting should be booked, e.g., in the next week, next two weeks, etc. In order to perform the function, the system must have this information and also must have calendar information for each of the people in the meeting, the set of locations where the meeting may take place, and ideally some way of knowing how long a person would have to travel to the location.

8.3.3 Using scenarios in conceptual design

In Chapter 7, we introduced scenarios as informal stories about user tasks and activities. They are a powerful mechanism for communicating among team members and with users. We stated in Chapter 7 that scenarios could be used and refined through different data-gathering sessions, and they can indeed be used to check out potential conceptual models.

Scenarios can be used to explicate existing work situations, but they are more commonly used for expressing proposed or imagined situations to help in conceptual design. Often, stakeholders are actively involved in producing and checking through scenarios for a product. Bødker identifies four roles that have been suggested for scenarios (Bødker, 2000, p. 63):

- as a basis for the overall design
- for technical implementation
- as a means of cooperation within design teams
- as a means of cooperation across professional boundaries, i.e., as a basis of communication in a multidisciplinary team

In any one project, scenarios may be used for any or all of these. Box 8.4 details how different scenarios were used throughout the development of a speech-

Scenario 3: Hyper-wonderland

This scenario addresses the positive aspects of how a hypermedia solution will work.

The setting is the Lindholm construction site sometime in the future.

Kurt has access to a portable PC. The portables are hooked up to the computer at the site office via a wireless modem connection, through which the supervisors run the hypermedia application.

Action: During inspection of one of the caissons[1] Kurt takes his portable PC, switches it on and places the cursor on the required information. He clicks the mouse button and gets the master file index together with an overview of links. He chooses the links of relevance for the caisson he is inspecting.

Kurt is pleased that he no longer needs to plan his inspections in advance. This is a great help because due to the 'event-driven' nature of inspection, constructors never know where and when an inspection is taking place. Moreover, it has become much easier to keep track of personal notes, reports etc. because they can be entered directly on the spot.

The access via the construction site interface does not force him to deal with complicated keywords either. Instead, he can access the relevant information right away, literally from where he is standing.

A positive side effect concerns his reachability. As long as he has logged in on the computer, he is within reach of the secretaries and can be contacted when guests arrive or when he is needed somewhere else on the site. Moreover, he can see at a glance where his colleagues are working and get in touch with them when he needs theii help or advice.

All in all, Kurt feels that the new computer application has put him more in control of things.

Scenario 4: Panopticon

This scenario addresses the negative aspects of how a hypermedia solution will work.

The setting is the Lindholm construction site sometime in the future.

Kurt has access to a portable PC. The portables are hooked up to the computer at the site office via a wireless modem connection, through which the supervisors run the hypermedia application.

Action: During inspecting one of the caissons Kurt starts talking to one of the builde n about some reinforcement problem. They argue about the recent lab tests. and he takes out his portable PC in order to provide some data which justify his arguments. It takes quite a while before he finds a spot where he can place the PC. either there is too much light, or there is no level surface at a suitable height. Finally, he puts the laptop on a big box and switches it on. He positions the cursor on the caisson he is currently inspecting and clicks the mouse to get into the master file. The table of contents pops up and from the overview of links he chooses those of relevance - but no lab test appears on the screen. Obviously, the file has not been updated as planned.

Kurt is rather upset. This loss of prestige in front of a contractor engineer would not have happened if he had planned his inspection as he had in the old days.

Sometimes, he feels like a hunted fox especially in situations where he is drifting around thinking about what kind of action to take in a particular case. If he has forgotten to log out, he suddenly has a secretary on the phone: "I see you are right at caisson 39. so could you not just drop by and take a message?"

All in all Kurt feels that the new computer application has put him under control.

[1]Used in building to hold water back during construction.

Figure 8.8 Example plus and minus scenarios.

recognition system. More specifically, scenarios have been used as scripts for user evaluation of prototypes, providing a concrete example of a task the user will perform with the product. Scenarios can also be used to build a shared understanding among team members of the kind of system being developed. Scenarios are good at selling ideas to users, managers, and potential customers. For example the scenario presented in Figure 7.7 was designed to sell ideas to potential customers on how a product might enhance their lifestyles.

An interesting idea also proposed by Bødker is the notion of *plus* and *minus scenarios*. These attempt to capture the most positive and the most negative consequences of a particular proposed design solution (see Figure 8.8) thereby helping designers to gain a more comprehensive view of the proposal.

ACTIVITY 8.5 Consider an in-car navigation device for planning routes, and suggest one plus and one minus scenario. For the plus scenario, try to think of all the possible benefits of the device. For the minus scenario, try to imagine everything that could go wrong.

Comment *Scenario 1* This plus scenario shows some potential positive aspects of an in-car navigation system.

"Beth is in a hurry to get to her friend's house. She jumps into the car and switches on her in-car navigation system. The display appears quickly, showing her local area and indicating the current location of her car with a bright white dot. She calls up the memory function of the device and chooses her friend's address. A number of her frequent destinations are stored like this in the device, ready for her to pick the one she wants. She chooses the "shortest route" option and the device thinks for a few seconds before showing her a bird's-eye view of her route. This feature is very useful because she can get an overall view of where she is going.

Once the engine is started, the display reverts to a close-up view to show the details of her journey. As she pulls away from the pavement, a calm voice tells her to "drive straight on for half a mile, then turn left." After half a mile, the voice says again "turn left at the next junction." As Beth has traveled this route many times before, she doesn't need to be told when to turn left or right, so she turns off the voice output and relies only on the display, which shows sufficient detail for her to see the location of her car, her destination and the roads she needs to use."

Scenario 2 This minus scenario shows some potential negative aspects of an in-car navigation system.

"Beth is in a hurry to get to her friend's house. She gets in her car and turns on the in-car navigation system. The car's battery is faulty so all the information she had entered into the device has been lost. She has to tell the device her destination by choosing from a long list of towns and roads. Eventually, she finds the right address and asks for the quickest route. The device takes ages to respond, but after a couple of minutes displays an overall view of the route it has found. To Beth's dismay, the route chosen includes one of the main roads that is being dug up over this weekend, so she cannot use the route. She needs to find another route, so she presses the cancel button and tries again to search for her friend's address through the long list of towns and roads. By this time, she is very late."

BOX 8.4 Using Scenarios throughout Design

Scenarios were used throughout the design of a speech-recognition system (Karat, 1995). The goal of the project was to produce a product that used speech-recognition technology, so there was no defined set of user requirements to start with. The system offered speech-to-text dictation capabilities and also speech command capabilities for an application running on the same platform.

Initially, scenarios were used to set the direction of the project: discussions revolved around whether the scenario was correct or not, i.e., whether people would want to use the device to achieve the suggested task. Then scenarios were used to sketch out screens and an early user guide. Discussions at this point included checking what information was needed on the screen at what time, and also deciding what components needed to be built. Use-oriented scenarios, i.e., scenarios suggesting how the device might be used, formed the basis of early design meetings that resulted in a shared understanding of what facilities the system might include. An example scenario from basic direction setting was, "Imagine taking away the keyboard and mouse from your current workstation and describe doing everything through voice commands."

Once the basic direction was agreed, further scenarios were generated to discuss the components of the system. These scenarios focused on typical use of speech commands so that vocabulary could be tracked. An example scenario for discussing vocabulary and system components was:

> *Overall task*: Open system editor, find file REPORT.TXT, change font to Times 16, save changes, and exit the editor.

This scenario was then broken down into a specific word list as follows:

> *Voice scenario steps*: "system_editor" "open" "open" "file" "find" "r" "e" "p" "open" "font" "times" "16" "ok" "save" "close"

A short user guide was developed early on, in parallel with the initial scenario development. User guide scenarios were generated by thinking about the kinds of questions a user might need to answer, for example, What is a speech manager? How do I know what I can say?

Once early prototypes were developed, scenarios together with additional tasks were used as a basis for user testing. One of the problems was that people were unsure of what they could say, and although the system included a "What can I say?" module, this itself proved difficult to use. An example scenario used in testing was "Change the background color of the icon for the communications folder to red."

Scenarios in the form of video prototypes were taken to potential customers later in the project for feedback. The feedback they received was mostly in scenario form too, and the scenarios extracted were fed back into the design process. For example, one of the scenarios collected was, "I would like to walk around while I dictate." This could be accommodated by making mobility a factor when selecting the microphone.

Collecting feedback in the form of scenarios continued later in the project, and these informed both the design of the product and the associated documentation.

8.3.4 Using prototypes in conceptual design

The whole point of producing a prototype is to allow some evaluation of the emerging ideas to take place. As pointed out above, prototypes are built in order to answer questions. Producing anything concrete requires some consideration of the details of the design. If the prototype is to be evaluated seriously by users, then they must be able to see how their tasks might be supported by the product, and this will require consideration of more detailed aspects.

Prototyping is used to get feedback on emerging designs. This feedback may be from users, or from colleagues, or it may be feedback telling you that the idea is not technically feasible. Different kinds of prototype are therefore used at different points in the development iterations and with different people. Generally speaking, low-fidelity prototypes (such as paper-based scenarios) are used earlier in design and higher-fidelity prototypes (such as limited software implementations) are used later in design. However, low-fidelity prototypes are not very impressive to look at, so if the feedback you're looking for is approval from people who will be basing their judgment on first impressions, then a horizontal, high-fidelity prototype might suit the job better than one based on post-its or cards.

Figure 8.9 shows a card-based prototype for the shared calendar system created for a user testing session to check that the task flow and the information requirements were correct for the task of arranging a meeting. The first card shows the screen that asks the user for relevant information to find a suitable meeting date. The second card shows the screen after the system has found some potentially suitable dates and displays the results. Finally, the third screen depicts the situation

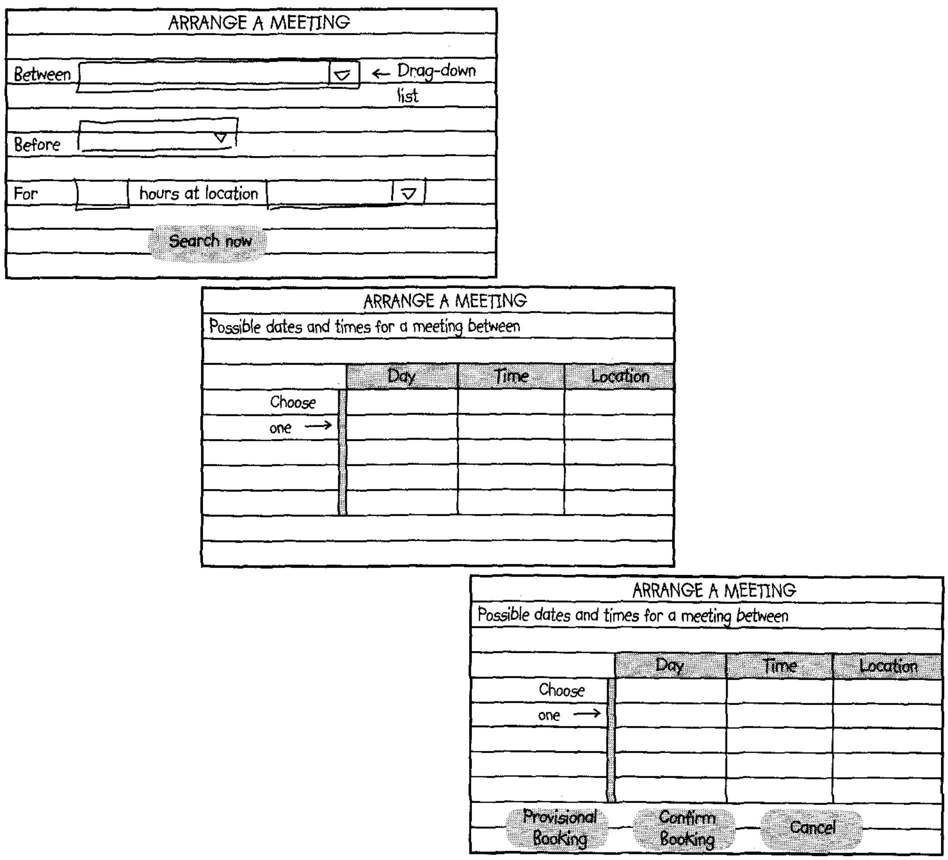


Figure 8.9 A card-based prototype for booking a meeting in the shared calendar system.

after a user has chosen one of the dates and is asked to provisionally book the chosen option, to confirm that this should be booked, or to cancel.

Note that at this point we have not decided how the navigation will work, i.e., whether there will be a tool bar, menus, etc. But we have included some detailed aspects of the design, in order to provide enough detail for users to interact with the prototype.

To illustrate how these cards can be used and the kind of information they can yield, we held a prototyping session with a potential user of the calendar. The session was informal (a kind of "quick and dirty" evaluation that you'll learn more about in Chapter 11) and lasted about 20 minutes. The user was walked through the task to see if the work flow was appropriate for the task of booking a meeting. Generally, the work flow agreed with the user's model of the task, but the session also highlighted some further considerations that did not arise in the original data gathering. Some of these had to do with work flow, but others were concerned with more detailed design. For example, the user suggested that it should be possible to state a range of dates rather than just a "before" date; he also thought that the people attending the meeting should have a chance to confirm the date through the system, and then when everyone had confirmed, the booking could be confirmed and placed in the calendar. On the detailed design, he thought that date entry through a matrix rather than a drop-down list would be more comfortable, and he asked how the possible meeting dates would be ordered. There were many more comments, all of which would be food for thought in the design. We considered only the one task, and yet it yielded a lot of very useful information.

ACTIVITY 8.6 Produce a card-based prototype for the library catalog system and the task of borrowing a book as described by the scenario, use case, and HTA in Chapter 7. You may also like to ask one of your peers to act as a user and step through the task using the prototype.

Comment Our version of the prototype is shown in Figure 8.10.

8.4 Physical design: getting concrete

Physical design involves considering more concrete, detailed issuer; of designing the interface, such as screen or keypad design, which icons to use, how to structure menus, etc.

There is no rigid border between conceptual design and physical design. As you saw above, producing a prototype inevitably means making some detailed decisions, albeit tentatively. Interaction design is inherently iterative, and so some detailed issues will come up during conceptual design; similarly., during physical design it will be necessary to revisit decisions made during conceptual design. Exactly where the border lies is not relevant. What is relevant is that the conceptual design should be allowed to develop freely without being tied to physical constraints too early, as this might inhibit creativity.

Design is about making choices and .decisions, and the designer must strive to balance environmental, user, data and usability requirements with functional

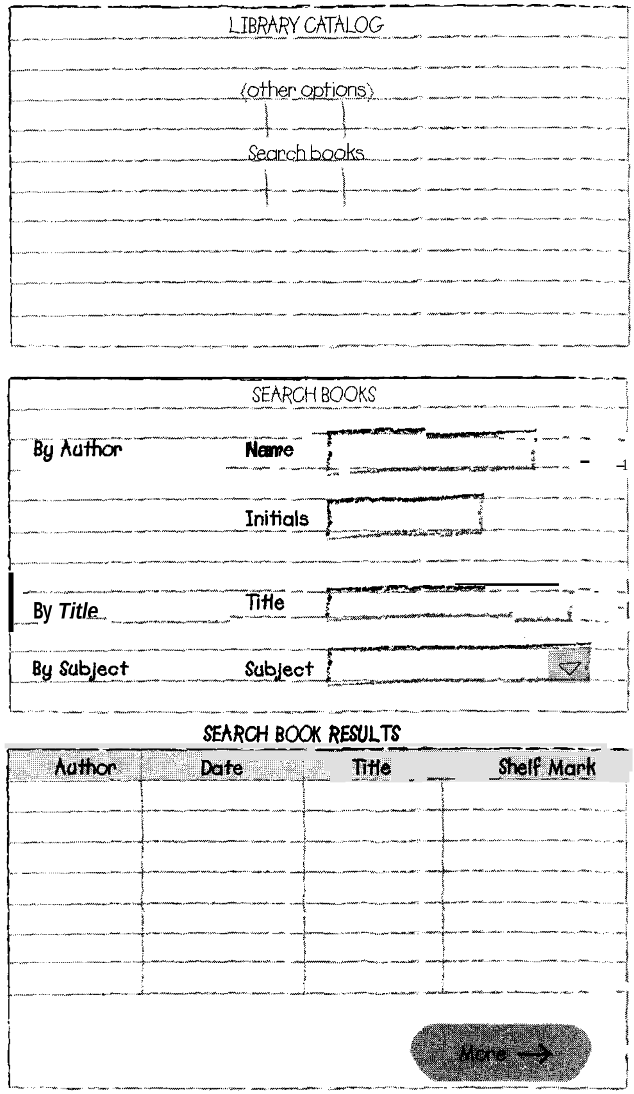


Figure 8.10 A card-based prototype for borrowing a book in the library catalog system.

requirements. These are often in conflict. For example, a cell phone must provide a lot of functionality but is constrained by having only a small screen and a small keyboard. This means that the display of information is limited and the number of unique function keys is also limited, resulting in restricted views of information and the need to associate multiple functions with function keys. Figure 8.11 shows the number of words it can display.

There are many aspects to the physical design of interactive products, and we can't cover them all in this book. Instead, we introduce some principles of

Figure 8.11 An average cell phone screen can display only a short message legibly.

good design in the context of some common interface elements. On our website (www.ID-book.com), you will find more activities and concrete examples of physical design.

8.4.1 Guidelines for physical design

The way we design the physical interface of the interactive product must not conflict with the user's cognitive processes involved in achieving the task. In Chapter 3, we introduced a number of these processes, such as attention, perception, memory, and so on, and we must design the physical form with these human characteristics very much in mind. For example, to help avoid memory overload, the interface should list options for us instead of making us remember a long list of possibilities. A wide range of guidelines, principles, and rules has been developed to help designers ensure that their products are usable, many of which are embodied in style guides and standards (see Box 8.5 for more information on this). Nielsen's set of guidelines were introduced in Chapter 1 in the form of heuristics. Another well-known set intended for informing design is Shneiderman's eight golden rules of interface design (Shneiderman, 1998):

1. *Strive for consistency.* For example, in every screen have a 'File' menu in the top left-hand corner. For every action that results in the loss of data, ask for confirmation of the action to give users a chance to change their minds.
2. *Enable frequent users to use shortcuts.* For example, in most word-processing packages, users may move around the functions using menus or shortcut "quick keys," or function buttons.
3. *Offer informative feedback.* Instead of simply saying "Error 404," make it clear what the error means: "The URL is unknown." This feedback is also influenced by the kinds of users, since what is meaningful to a scientist may not be meaningful to a manager or an architect.
4. *Design dialogs to yield closure.* For example, make it clear when an action has completed successfully: "printing completed."
5. *Offer errorprevention and simple error handling.* It is better for the user not to make any errors, i.e., for the interface to prevent users from making mistakes. However, mistakes are inevitable and the system should be forgiving about the errors made and support the user in getting back on track.
6. *Permit easy reversal of actions.* For example, provide an "undo" key where possible.
7. *Support internal locus of control.* Users feel more comfortable if they feel in control of the interaction rather than the device being in control.

8. ***Reduce short-term memory load.*** For example, wherever possible, offer users options rather than ask them to remember information from one screen to another.

Other guidelines that have been suggested include keeping the interaction simple and clear, organizing interface elements to aid understanding and use through suitable groupings, and designing images to be immediate and generalizable. All of

BOX 8.5 Design Guidelines and Standards

Design guidelines and standards exist to help designers create better designs by learning from others' experience. Some guidelines are at a detailed level and are called design rules, while others are more abstract, require interpretation before being applied, and are called design principles. For example, one very general but very pertinent guideline for website design is, "Keep it simple." This is relevant throughout design but must be interpreted within the specific context before it can be applied. On the other hand, a design rule for web design might be, "Don't offer an option to search the whole web from your own website." This is a very specific rule that requires no interpretation to apply. These terms were introduced in Box 1.5 of Chapter 1, together with some others commonly used in this context.

Design Principles

Principles often embody design-related information derived from theory, and so this is one way in which the cognitive models and processes introduced in Chapter 3 can be put to practical use in your designs. For example, the guideline to use "recognition rather than recall" is based on theories of memory that say people find it is easier to recognize things than to remember them with no prompting. However carefully names are chosen to reflect function, it is easier to choose the right option from a list of options than to remember the name of a command. Shneiderman's (1998) eight golden rules of interface design are an example set of principles.

Rules

Rules are more specific versions of design guidelines and provide more detailed guidance. A classic example of design rules is the collection by Smith and Mosier (1986). These rules are quite detailed and prescriptive. For example, one dealing with consistent format states, "Adopt a consistent format for the location of various display features from one display to another." Each rule is accompanied by explanatory notes such as examples, exceptions, comments that provide detailed guidance, and any useful references to their own or others' work.

Style guides

A style guide is a collection of specific design rules and principles from which the rules are derived. They are used to ensure a consistent look and feel across a set of applications. The most widely known style guides are those for Windows development (Microsoft Corporation, 1992) and for Macintosh development (Apple Computer Inc., 1993). An example from the Macintosh human interface guidelines concerned with designing color icons states that "When you design an icon, you should start by creating the black-and-white version first, then the color should be added."

Style guides tailored to a specific company can be used to deliver a particular corporate image. Such style guides are called corporate style guides.

Standards

Some international standards govern the development of interactive systems. These are also collections of principles and rules to provide designers with a framework based on others' experience. The most pertinent standards are:

- ISO9241 Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)
- ISO 13407 Human-centered Design Processes for Interactive Systems.
- ISO 14915 Design of the User Interface of Multimedia Applications.

these focus on making the communication between user and product as clear as possible.

Extensive experience in the art of communication (through posters, text, books, images, advertising, etc.) is relevant to interaction design. In her interview at the end of Chapter 6, Gillian Crampton Smith identifies the roles that traditional designers can play in interaction design; one of them she highlights is the fact that designers are trained to produce a coherent design that delivers the desired message to the intended audience. Including such designers on the team can bring this experience to bear. Mullet and Sano (1995) identify a number of useful design principles arising from the visual arts.

To see how these can be translated into the context of interaction design, we consider their application to different widgets, i.e., screen elements, in the next section.

8.4.2 Different kinds of widget

Interfaces are made up of widgets, elements such as dialog boxes, menus, icons, toolbars, etc. Each element must be designed or chosen from a predesigned set of widgets. Sometimes these decisions are made for you through the use of a style guide. Style guides may be commercially produced, such as the Windows style guide (called commercial style guides), or they may be internal to a company (called corporate style guides). A style guide dictates the look and feel of the interface, i.e., which widgets should be used for which purpose and what they look like. For example, study your favorite Windows applications. Which menu is always on the right-hand side of the toolbar? What icon is used to represent "close" or "print"? Which typeface is used in menus and dialog boxes? Each Windows product has the same look and feel, and this is specified in the Windows style guide. If you go to a commercial website, you may find that each screen also has the same look and feel to it. This kind of corporate identity can be captured in a corporate style guide. More information about standards and style guides is in Box 8.5.

We consider here briefly three main aspects of interface design: menu design, icon design, and screen layout. These are applicable to a wide range of interactive products, from standard desktop interfaces for PC software, to mobile communicator functions and microwave ovens.

Menu design Menus provide users with a choice that can be a choice of commands or a choice of options related to a command. They provide the means by which the user can perform actions related to the task in hand and therefore are based on task structure and the information required to perform a task.

Menus may be designed as drop-down, pop-up or single-dialog menus. It may seem obvious how to design a menu, but if you want to make the application easy to use and provide user satisfaction, some important points must be taken into account. For example, for pull-down and pop-up menus, the most commonly used functions should be at the top, to avoid frequent long scans and scrolls. The principle of grouping can be used to good effect in menu design. For example, the menu can be divided into collections of items that are related, with each collection being

5.2 Grouping options in a menu

Menu options should be grouped within a menu to reflect user expectations and facilitate option search.

5.2.1 Logical groups

If the menu option contains a large number of options (eight or more) and these options can be logically grouped, options should be grouped by function or into other logical categories which are meaningful to users.

EXAMPLE: Grouping the commands in a word-processing system into such categories as customise, compose, edit, print.

5.2.2 Arbitrary groups

If 8 or more options are arranged arbitrarily in a menu panel, they should be arranged into equally distributed groups utilising the following equation:

$$g = \sqrt{n}$$

where

g is the number of groups
n is the number of options on the panel.

EXAMPLE: Given **19** options in a menu panel, arrange them into 4 groups of about **5** options each.

Figure 8.12 An excerpt from ISO 9241 concerning how to group items in a menu.

separated from others. Opposite operations such as "quit" and "save" should be clearly separated to avoid accidentally losing work instead of saving it (See Figure 1.6 in Chapter 1).

An excerpt from ISO 9241, a major international standard for interaction design, considers grouping in menu design, as shown in Figure 8.12.

To show how the design of menus may proceed, we return to the shared calendar. In our initial data gathering, we identified a number of possible tasks that the user might want to perform using the calendar. These included making an entry, arranging a meeting among a number of people, entering contact details, and finding out other people's engagements. Tied to these would also be a number of administrative and housekeeping actions such as deleting entries, moving entries, editing entries, and so on. Suppose we stick with just this list. The first question is what to call the menu entries. Menu names need to be short, clear, and unambiguous. The space for listing them will be restricted, so they must be short, and you want them to be distinguishable, i.e., not easily confused with one another so that the user won't choose the wrong one by mistake. Our current descriptions are really too long. For example, instead of "find out other people's engagements" we could have ***Query entry*** as a menu option, following through to a dialog box that asks for relevant details.

We need to consider logical groupings. In this case, we could group according to user goal, i.e., have ***Query entry, Add entry, Edit entry, Move entry,*** and ***Delete entry*** grouped together (see Figure 8.13). Similarly, we could group Add ***contact,***

Calendar Entry	Contacts	Arrange Meeting
Add Entry	Add Contact	
Edit Entry	Edit Contact	
Move Entry	Delete Contact	
Delete Entry		

Figure 8.13 Possible menu groupings for the shared calendar system.

Edit contact and ***Delete contact*** together. Finding other people's engagements could be generalized to a simple ***Search*** option that led to a dialog box in which the search parameters are specified. Arranging a meeting is also an option that doesn't clearly group with other commands. This and the ***Search*** option may be better represented as options on a toolbar than as menu items on their own.

Icon design Designing a good icon takes more than a few minutes. You may be able to think up good icons in a matter of seconds, but such examples are unlikely to be widely acceptable to your user group. When symbols for representing ladies' and gents' toilets first appeared in the UK, a number of confused tourists did not understand the culturally specific icons of a woman wearing a skirt and a man wearing trousers. For example, some people protested that they thought the male icon was a woman wearing a trouser suit. We are now all used to these symbols, and indeed internationally recognized symbols for how to wash clothes, fire exits, road signs, etc. now exist. However, icons are cultural and context-specific. Designing a good icon takes time.

At a simple level, designers should always draw on existing traditions or standards, and certainly should not contradict them. Concrete objects or things are easier to represent as an icon since they can be just a picture of the item. Actions are harder but can sometimes be captured. For example, using a picture of a pair of scissors to represent "cut" in a word-processing application provides sufficient clues as long as the user understands the convention of "cut" for deleting text.

In our shared calendar, if we are going to have the ***Search*** and ***Arrange a Meeting*** commands on a tool bar, we need to identify a suitable icon for each of them. A number of possible icons spring to mind for the ***Search*** option, mainly because searching is a fairly common action in many interactive products: a magnifying glass or a pair of binoculars are commonly used for such options. Arranging a meeting is a little difficult, though. It's probably easier to focus on the meeting itself than the act of arranging the meeting, but how do you capture a meeting? You want the icon to be immediately recognizable, yet it must be small and simple. What characteristic(s) of a meeting might you capture? One of the things that comes to mind is a group of people, so maybe we could consider a collection of stick people? Another element of a meeting is usually a table, but a table on its own isn't enough, so maybe having a table with a number of people around it would work?

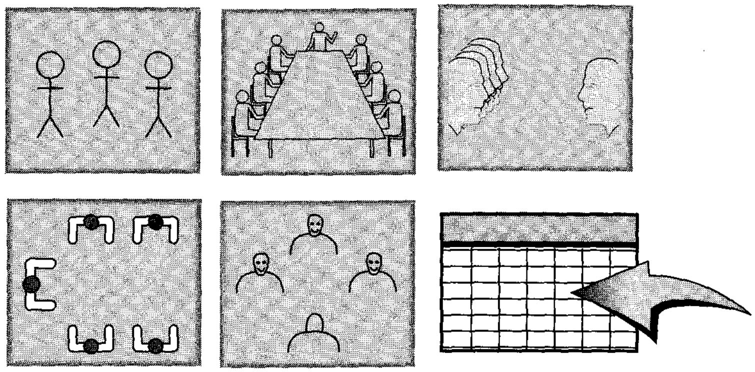


Figure 8.14 **A** variety of possible icons to represent the "arrange a meeting" function.

ACTIVITY 8.7 Sketch a simple, small icon to represent a set of people around the table, or suggest an icon of your own. Show it to your peers or friends, tell them that it's an icon for a shared calendar application, and see if they can understand what it represents.

Comment **A** variety of attempts are shown in Figure 8.14. The last icon is the icon that paim.net uses for arranging meetings. This is a different possibility that tries to capture the fact that you're entering data into the planner.

We discussed some cognitive aspects relevant to icon design in Chapter **3.** For example, icons must be designed so that users can readily perceive their meaning and so that they are distinguishable one from another. Since the size of icons on the screen is often very small, this can be difficult to achieve, but users must be able to tell them apart. Look back again at Figure 3.4 and the activity associated with it. How easy do you think it would be to tell some of these icons apart if they were just a little smaller, or the screen resolution was lower?

Screen *design.* There are two aspects to screen design: how the task is split across a number of screens, and how the individual screens are designed.

The first aspect can be supported by reference to the task analysis, which broke down the user's task into subtasks and plans of action. One starting point for screen design is to translate the task analysis into screens, so that each task or subtask has its own screen. This will require redesign and adjustment, but it is a starting point. The interaction could be divided into simple steps, each involving a decision or simple data entry. However, this can become idiotic, and having too many simple screens can become just as frustrating as having information all crammed into one screen. This is one of the balances to be drawn in screen design. Tasks that are more complicated than this (and are usually unsuited to simple task analysis) may require a different model of interaction in which a number of screens are open at the same time and the user is allowed to switch among them.

Another issue affecting the division of a task across screens is that all pertinent information must be easily available at relevant times.

Guidelines for the second aspect, individual screen design, draw more clearly from some of the visual communication principles we mentioned above: for example, designing the screen so that users' attention is drawn immediately to the salient points, and using color, motion, boxing and grouping to aid understanding and clarity. Each screen should be designed so that when users first see it, their attention is focused on something that is appropriate and useful to the task at hand. Animations can be very distracting if they are not relevant to the task, but are effective if used judiciously.

Good organization helps users to make sense of an interaction and to interpret it within their own context (as discussed in Chapter **3).** This is another example where principles of good grouping can be applied, for example, grouping similar things together or providing separation between dissimilar or unrelated items. Grouping can be achieved in different ways: by placing things close together, using colors, boxes, or frames to segregate items, or using shapes to indicate relationships among elements. There is a trade-off between sparsely populated screens with a lot of open space and overcrowded screens with too many and too complicated sets of icons. If the screen is overcrowded, then users

BOX 8.6 Design Patterns for HCI

Design patterns have become popular in software engineering since the early 1990s. Patterns capture experience, but they have a different structure and a different philosophy from other forms of guidance, such as the guidelines we introduced earlier, or specific methods. One of the intentions of the patterns community is to create a vocabulary, based on the names of the patterns, that designers can use to communicate with one another and with users. Another is to produce a literature in the field that documents experience in a compelling form.

The idea of patterns was first proposed by Christopher Alexander, a British architect who described patterns in architecture. His hope was to capture the "quality without a name" that is recognizable in something when you know it's good.

But what is a pattern? One simple definition is that it is a solution to a problem in a context. What this means is that a pattern describes a problem, a solution, and where this solution has been found to work. This means that users of the pattern can see not only the problem and solution, but also understand when and where it has worked before and access a rationale for why it worked. This helps designers in adopting it (or not) for themselves.

Patterns on their own are interesting, but are not as powerful as a pattern language. A pattern language is a network of patterns that reference one another and work together to create a complete structure.

The application of patterns in HCI is still in its infancy. But some work has been done in the area, and some pattern languages have been proposed. One of the most mature languages is that described by Jan Borchers (2001) for interactive music exhibits. Borchers presents three related languages: one for music, one for HCI and one for software engineering, all of which have arisen from his experience of designing music exhibits. The HCI language addresses issues such as accommodating groups as well as single users, handling complexity, content structure, and interaction devices.

BOX 8.7 Designing for the Web

Web pages need to exhibit the kinds of good interaction design we talk about in this chapter, but they also have some specific requirements. For example, Nielsen (2000) has suggested a set of evaluation criteria specifically for the web (see Chapter 13 for more detail).

The key design issues for websites that are different from other interaction designs are captured very well by three questions proposed by Keith Instone (quoted in Veen, 2001): Where am I? What's here? Where can I go? Each web page should be designed with these three questions in mind. The answers must be clear to users. Jeffrey Veen (2001) expands on these questions. He suggests that a simple way to view a web page is to deconstruct it into three areas (see Figure 8.15). Across the top would be the answer to "Where am I?" Because users can arrive at a site from any direction (and rarely through the front door, or home page), telling them where they are is critical. Having an area at the top of the page that "brands" the page instantly provides that information. Down the left-hand side is an area in which navigation or menus sit. This should allow users immediately to see what else is available on the site, and answers the question "Where can I go?'

The most important information, and the reason a user has come to the site in the first place, is provided in the third area, the content area, which answers the question "What's here?" Content for web pages must be designed differently from standard documents, since the way users read web pages is different. On web pages, content should be short and precise, with crisp sentences. Using headlines to capture the main points of a paragraph is one way to increase the chances of your message getting over to a user who will, perhaps, merely scan the page rather than look at it in detail.

Branding a page is important for the reasons given above. We have also talked about the need to keep screens uncluttered so that people can find their way around and see clearly what is available. However, there may be occasions when the need to maintain a brand overrides other design issues. For example, the website for the Swedish newspaper *Aftonbladet*, while quite busy and crowded (see Figure 8.16), was designed to continue the style of the paper-based version, which also has a busy and crowded appearance.

Download times are critical for the success of websites. Users who have to wait too long for a page will move on somewhere else. So although it's attractive to have graphics on your pages, use them sparingly. One suggestion from Nielsen (2000) is to have few graphics on the welcoming or more abstract pages, and offer users the chance to see pictures of products, or maps, or whatever, only when they explicitly ask for them. It is quite common to use thumbnails, miniaturized versions of the full picture, as links.

Traditionally, hyperlinks have been indicated by highlighting the text in blue and underlining it.

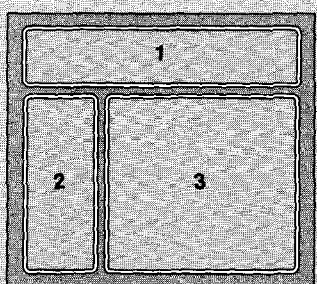


Figure 8.15 Any web page has three main areas.

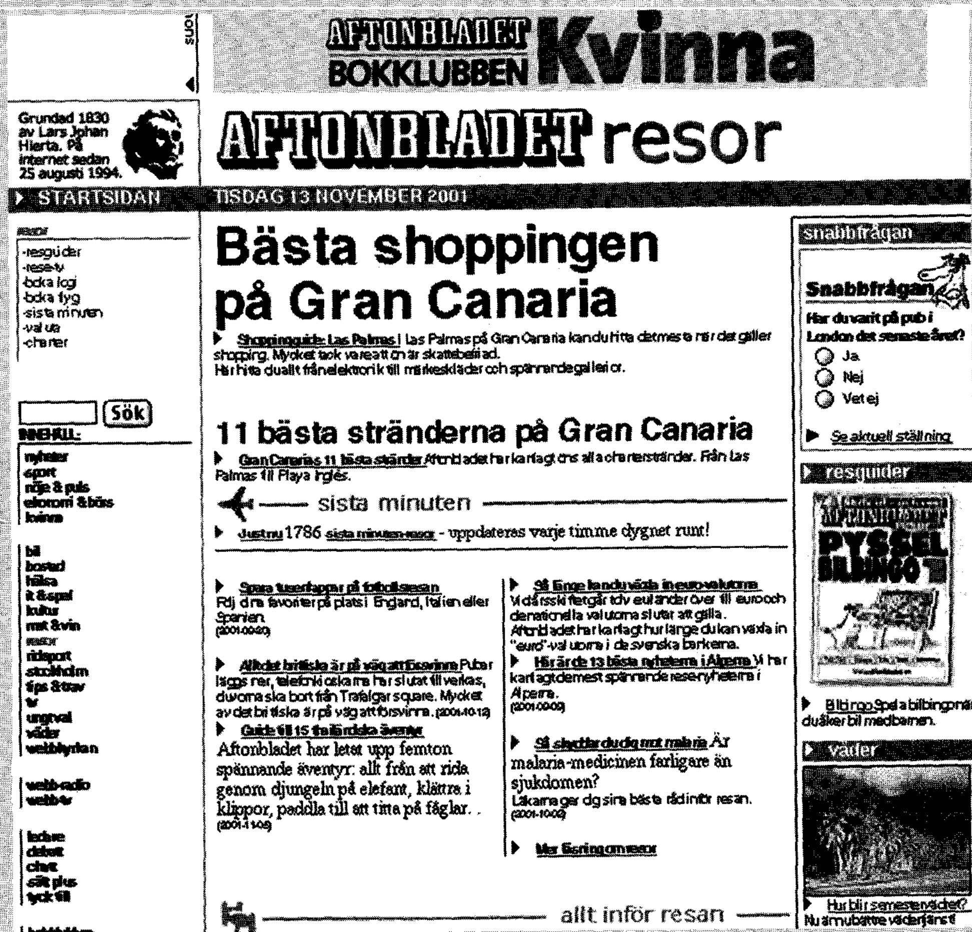


Figure 8.16 The front web page of the Aftonbladet newspaper.

This is how links were indicated in hypertext, and this convention transferred to web pages. However, now that web pages can be designed more flexibly, some sites are moving away from the traditional blue underlining and are using "roll-over" tags to signify links, i.e., when the mouse rolls over a word then a label appears. This lack of consistency, together with the fact that underlined text is not always a link, can cause considerable confusion for the user.

will become confused and distracted. But too much open space and consequently many screens can lead to frequent screen changes, and a disjointed series of interactions.

information *display*. Making sure that the relevant information is available for the task is one aspect of information display, but another concerns the format. Different types of information lend themselves to different kinds of display. For example, data that is discrete in nature, such as sales figures for the last month, could be displayed graphically using a digital technique, while data that is continuous in nature, such as the percentage increase in sales over the last month, is better displayed using an analog device.

If data is to be transferred to the device from a paper-based medium or *vice* versa, it makes sense to have the two consistent. This reduces user confusion and search time in reconciling data displayed with data on the paper.

In the shared calendar application, there is potentially a lot of information to display. If you have five members of the department, each with their own calendars, and the departmental calendar too, then you need to display six sets of engagement information. When we showed the prototype system to our user, he suggested that dates should be chosen through a matrix of some kind rather than a drop-down list. Displaying information appropriately can make communication a lot easier.

8.5 Tool support

The tools available to support the activities described here are wide-ranging and various. We mentioned development environments when talking about prototypes in Section 8.2, but other kinds of support are available.

Much research has been done into appropriate support for different kinds of design and software production, resulting in a huge variety of tools. Because technology moves so quickly, any discussion of specific tools would be quickly out of date. Up-to-date information about support tools can be found on our website (www.ID-book.com). Here we report on some general observations about software tools.

Brad Myers (1995) suggests nine facilities that user interface software tools might provide:

- help design the interface given a specification of the end users' tasks
- help implement the interface given a specification of the design
- create easy-to-use interfaces
- allow the designer to rapidly investigate different designs
- allow nonprogrammers to design and implement user interfaces
- automatically evaluate the interface and propose improvements
- allow the end user to customize the interface
- provide portability
- be easy to use

In a later paper Myers et al. (2000), look at the past, present, and future of user interface tools. Box 8.8 describes some types of tool that have been successful and some that have been unsuccessful.

BOX 8.8 Successes and Failures for User Interface Tools *(Myers et al., 2000)*

Looking at the history of user interface design tools, we can see some tools that have been successful and have withstood the test of time, and others that have fallen by the wayside. Understanding something of what works and what doesn't gives us lessons for the future of such tools.

Tools that have been successful are:

Window Managers and Toolkits. The idea of overlapping windows was first proposed by Alan Kay (Kay, 1969). These have been successful because they help to manage scarce resources: screen space and human perceptual and cognitive resources such as limited visual field and attention.

Event languages that are designed to program actions based on external events: for example, when the left mouse button is depressed, move the cursor here. These have worked because they map well to the direct manipulation graphical user interface style.

Interactive graphical tools or interface builders, such as Visual Basic. These allow the easy construction of user interfaces by placing interface elements on a screen using a mouse. They have been successful because they use a graphical means to design a graphical layout, i.e., you can build a graphical screen layout by grabbing and placing graphical elements without touching any program code.

Component systems are based on the idea of dynamically combining individual components that have been separately written and compiled. Sun's Java Beans uses this approach. One reason for its success is that it addresses the important software engineering goal of modularity.

Scripting languages have become popular because they support fast prototyping. Example scripting languages are Python and Perl.

Hypertext allows elements of a document to be linked in a multitude of ways, rather than the traditional linear layout. Most people are aware of hypertext links because of their use on the web.

Object-oriented programming. This programming approach is successful in interface development because the objects of an interface such as buttons and other widgets can so readily be cast as objects in the language.

Promising approaches that have not caught on are:

Technology has changed so fast that in some cases the tools to support the development of certain technologies have failed to keep up with the rapidly changing requirements. Good ideas that have fallen by the wayside include:

User interface management tools (UIMS). The idea behind UIMS was akin to the idea behind database management systems. Their purpose was to abstract away the details of interface implementation to allow developers to specify and manipulate interfaces at a higher level of abstraction. This separation turned out to be undesirable, as it is not always appropriate to be able to understand and manipulate interface elements only at a high level of abstraction.

Formal language based tools. Many systems in the 1980s were based on formal language concepts such as state transition diagrams and parsers for context-free grammars. These failed to catch on because: the dialog-based interfaces for which these tools were particularly suited were overtaken by direct manipulation interfaces; they were very good at producing sequential interfaces, but not at expressing unordered sequences of action; and they were difficult to learn even for programmers.

Constraints. Tools that were designed to maintain constraints, i.e., relationships among elements of an interface such as that the scroll bar should always be on the right of the window, or that the color of one item should be the same as the color of other items. These systems have not caught on because they can be unpredictable. Once constraints are set up, the tool must find a solution to maintain them, and since there is more than one solution, the tool may find a solution the user didn't expect.

Model-based and automatic techniques. The aim of these systems was to let developers specify interfaces at a high level of abstraction and then for an interface to be automatically generated according to a predefined set of interpretation rules. These too have suffered from problems of unpredictability, since the generation of the interfaces relies on heuristics and rules that are themselves unpredictable in concert.

Assignment

This Assignment continues work on the web-based ticket reservation system at the end of Chapter 7.

(a) Based on the information gleaned from the assignment in Chapter 7, suggest three different conceptual models for this system. YOU should consider each of the aspects of a conceptual model discussed in this chapter: interaction paradigm, interaction mode, metaphors, activities it will support, functions, relationships between functions, and information requirements. Of these, decide which one seems most appropriate and articulate the reasons why.

(b) Produce the following prototypes for your chosen conceptual model.

(i) Using the scenarios generated for the ticket reservation system, produce a storyboard for the task of buying a ticket for one of your conceptual models. Show it to two or three potential users and get some informal feedback.

(ii) Now develop a prototype based on cards and post-it notes to represent the structure of the ticket reservation task, incorporating the feedback from the first evaluation. Show this new prototype to a different set of potential users and get some more informal feedback.

(iii) Using a software-based prototyping tool (e.g., Visual Basic or Director) or web authoring tool (e.g., Dreamweaver), develop a software-based prototype that incorporates all the feedback you've had so far. If you do not have experience in using any of these, create a few HTML web pages to represent the basic structure of your website.

(c) Consider the web page's detailed design. Sketch out the application's main screen (home page or data entry). Consider the screen layout, use of colors, navigation audio, animation, etc. While doing this, use the three main questions introduced in Box 8.7 as guidance: Where am I? What's here? Where can I go? Write one or two sentences explaining your choices, and consider whether the choice is a usability consideration or a user experience consideration.

Summary

This chapter has explored the activities of design prototyping and construction. Prototyping and scenarios are used throughout the design process to test out ideas for feasibility and user acceptance. We have looked at the different forms of prototyping, and the activities have encouraged you to think about and apply prototyping techniques in the design process.

Key points

- Prototyping may be low fidelity (such as paper-based) or high fidelity (such as software-based).
- High-fidelity prototypes may be vertical or horizontal.
- Low-fidelity prototypes are quick and easy to produce and modify and are used in the early stages of design.
- There are two aspects to the design activity: conceptual design and physical design.
- Conceptual design develops a model of what the product will do and how it will behave, while physical design specifies the details of the design such as screen layout and menu structure.

- **We have explored three perspectives to help you develop conceptual models: an interaction paradigm point of view, an interaction mode point of view, and a metaphor point of view.**
- **Scenarios and prototypes can be used effectively in conceptual design to explore ideas.**
- **We have discussed four areas of physical design: menu design, icon design, screen design, and information display.**
- **There is a wide variety of support tools available to interaction designers.**

Further reading

WINOGRAD, TERRY (1996) ***Bringing Design to Software.*** Addison-Wesley and ACM Press. This book is a collection of articles all based on the theme of applying ideas from other design disciplines in software design. It has a good mixture of interviews, articles, and profiles of exemplary systems, projects or techniques. Anyone interested in software design will find it inspiring.

CARROLL, JOHN M. (ed.) (1995) ***Scenario-based Design.*** John Wiley & Sons, Inc. This volume is an edited collection of papers arising from a three-day workshop on use-oriented design. The book contains a variety of papers including case studies of scenario use within design, and techniques for using them with object-oriented development, task models and usability engineering. This is a good place to get a broad understanding of this form of development.

MULLET, KEVIN, AND SANO, DARELL (1995) ***Designing Visual Interfaces.*** SunSoft Press. This book is full of practical guidance for designing interactions that focus on communication. The ideas here come from communication-oriented visual designers. Mullet and Sano show how to apply these techniques to interaction design, and they also show some common errors made by interaction designers that contravene the principles.

VEEN, JEFFREY (2001) ***The Art and Science of Web Design.*** New Riders. A very bright book, providing a lot of practical information taken from the visual arts about how to design websites. It also includes sections on common mistakes to help you avoid these pitfalls.

MYERS, BRAD, HUDSON, S. E., AND PAUSCH, R. (2000) Past, present and future of user interface software tools. ACM ***Transactions on Computer-Human Interaction,*** 7(1), ***3–28.*** This paper presents an interesting description of user interface tools, expanding on the information given in Box 8.8.

Chapter 9

User-centered approaches to interaction design

9.1 Introduction

As you would expect, user-centered development involves finding out a lot about the users and their tasks, and using this information to inform design. In Chapter 7 we introduced some data-gathering techniques which can be used to collect this information, including naturalistic observation. Studying people in their "natural" surroundings as they go about their work can provide insights that other data-gathering techniques cannot, and so interaction designers are keen to use this approach where appropriate. One particular method that has been used successfully for naturalistic observation in the social sciences is ethnography. It has also been used with some success in product development but there have been some difficulties knowing how to interpret and present the data gathered this way so that it can be translated into practical design.

Another aspect of user-centered development is user involvement in the development process. There are different degrees of involvement, one of which is through evaluation studies, as discussed in Chapters 10 through 14. Another is for users to contribute actively to the design itself—to become co-designers. As Gillian Crampton Smith said in the interview at the end of Chapter 6, users are not designers, but the payoffs for allowing users to contribute to the design themselves are quite high in terms of user acceptance of the product. So techniques have been developed that engage users actively and productively in design.

In this chapter, we discuss some issues surrounding user involvement, and expand on the principles underlying a user-centered approach. Then we describe two approaches to using ethnographic data to inform design and two approaches to involving users actively in design.

The main aims of this chapter are to:

- Explain some advantages of involving users in development.
- Explain the main principles of a user-centered approach.
- Describe some ethnographic-based methods aimed at understanding users' work.
- Describe some participative design techniques that help users take an active part in design decisions.

9.2 Why is it important to involve users at all?

We talked in Chapter 6 about the importance of identifying stakeholders and of consulting the appropriate set of people. In the past, developers would often talk to managers or to "proxy-users," i.e., people who role-played as users, when eliciting requirements. But the best way to ensure that development continues to take users' activities into account is to involve real users throughout. In this way, developers can gain a better understanding of their needs and their goals, leading to a more appropriate, more useable product. However, two other aspects which have nothing to do with functionality are equally as important if the product is to be usable and used: expectation management and ownership.

Expectation management is the process of making sure that the users' views and expectations of the new product are realistic. The purpose of expectation management is to ensure that there are no surprises for users when the product arrives. If users feel they have been "cheated" by promises that have not been fulfilled, then this will cause resistance and maybe rejection. Expectation management is relevant whether you are dealing with an organization introducing a new software system or a company developing a new interactive toy. In both cases, the marketing of the new arrival must be careful not to misrepresent the product. How many times have you seen an advert for something you thought would be really good to have, but when you see one, discover that the marketing "hype" was a little exaggerated? I expect you felt quite disappointed and let down. Well, this is the kind of feeling that expectation management tries to avoid.

It is better to exceed users' expedtations than to fall below them. This does not mean just adding more features, how*, but that the product supports the users' work more effectively than they expect. Inuolving users throughout development helps with expectation management because they can see from an early stage what the product's capabilities are and what they are not. They will also understand better how it will affect their jobs and what 'they can expect to do with the product; they are less likely to be disappointed. Users can also see the capabilities develop and understand, at least to some extent, why the features are the way they are.

Adequate and timely training is another technique for managing expectations. If you give people the chance to work with the product before it is released, either by training them on the real system or by offering hands-on demonstrations of a prerelease version, then they will understand better what to expect when the final product is released.

A second reason for user involvement is ownership. Users who are involved and feel that they have contributed to a product's development, are more likely to feel a sense of "ownership" towards it and to be receptive to it when it finally emerges. Remember Suzanne Robertson's comment in her interview at the end of Chapter **7** about how important it is for people to feel heard? Well, this is true throughout development, not just at the requirements stage.

9.2.1 Degrees of involvement

Different degrees of user involvement may be implemented in order to manage expectations and to create a feeling of ownership. At one end of the spectrum, users may be co-opted to the design team so that they are major contributors. For any one user, this may be on a full-time basis or a part-time basis, and it may be for the duration of the project or for a limited time only. There are advantages and disadvantages to each situation. If a user is co-opted full-time for the whole project, their input will be consistent and they will become very familiar with the system and its rationale. However, if the project takes many years they may lose touch with the rest of the user group, making their input less valuable. If a user is co-opted part-time for the whole project, she will offer consistent input to development while remaining in touch with other users. Depending on the situation, this will need careful management as the user will be trying to learn new jargon and handle unfamiliar material as a member of the design team, yet concurrently trying to fulfill the demands of their original job. This can become very stressful for the individuals. If a number of users from each user group are co-opted part-time for a limited period, input is not necessarily consistent across the whole project, but careful coordination between users can alleviate this problem. In this case, one user may be part of the design team for six months, then another takes over for the next six months, and so on.

At the other end of the spectrum, users may be kept informed through regular newsletters or other channels of communication. Provided they are given a chance to feed into the development process through workshops or similar events, this can be an effective approach to expectation management and ownership. In a situation with hundreds or even thousands of users it would not be feasible to involve them all as members of the team, and so this might be the only viable option.

If you have a large number of users, then a compromise situation is probably the best. Representatives from each user group may be co-opted onto the team on a full-time basis, while other users are involved through design workshops, evaluation sessions, and other data-gathering activities.

The individual circumstances of the particular project affect what is realistic and appropriate. If your end user groups are identifiable, e.g., you are developing a product for a particular company, then it is easier to involve them. If, however, you are developing a product for the open market, it is unlikely that you will be able to co-opt a user to your design team. Box 9.1 explains how Microsoft involves users in its developments.

One of the reasons often cited for not involving users in development is the amount of time it takes to organize, manage, and control such involvement. This issue may appear particularly acute in developing systems to run on the Internet where ever-shorter timescales are being forced on teams—in this fast-moving area, projects lasting three months or less are common. You might think, therefore, that it would be particularly difficult to involve users in such projects. However, Braiterman et al. (2000) report two case studies showing how to involve users successfully in large-scale but very short multidisciplinary projects, belying the claim that involving users can waste valuable development time.

The first case study was a three-week project to develop the interaction for a new web shopping application. The team included a usability designer, an information architect, a project manager, content strategists, and two graphic designers. In such a short timeframe, long research and prototyping sessions were impossible, so the team produced a hand-drawn paper prototype of the application that was

BOX 9.1 How Microsoft Involves Users *(Cusumano and Selby, 1995)*

The synch-and-stabilize process of development used by Microsoft was described in Chapter 6. Here we look at some of the main ways in which users are involved in the development process.

Users are involved throughout development in a variety of ways, from product and feature identification to feature development and testing, and via the customer support call centers.

Microsoft bases feature selection and prioritization on a technique called "activity-based planning." This technique involves studying what users do to achieve a certain activity like writing a letter, and using the results of the study to choose product features. Each new release of a software product is limited to supporting about four new major activities. Each of these proposed new activities can be broken down into subactivities and these mapped against features already existing in the software. Any new features required are noted. If a feature can support more than one activity, then it is placed higher in the priority list. The techniques used to gather customer data for activity-based planning do not appear to be prescribed in any way, and can vary from visiting customers through to asking them to use an instrumented version of the software, i.e., a version that records the actions they take. Microsoft also employs contextual inquiry (see below) to learn about their customers' work, although they find that it can be time-consuming and the results ambiguous.

Because the world of applications software changes so rapidly, developers need to continually observe and test with users. Throughout the development phase, usability tests are carried out in Microsoft's usability lab. Each time a developer believes that a feature is finished, then it is scheduled for testing in the usability lab. A group of about 10 users "off the street" are invited into the lab to perform certain tasks while their behavior is observed and their performance recorded. The data is then analyzed and the findings fed back into development. This results in thorough testing of all features. As an example, Office 4.0 (incorporating Word, Excel, PowerPoint, and other common office software) went through over 8000 hours of usability testing.

Once a product is complete, it is used internally by Microsoft staff (who are selected users and atypical, but are using it in a realistic working environment); then it may be released in a beta version to selected customers.

Microsoft has millions of customers around the world, about 30% of whom call their customer support lines with problems and frustrations resulting from poor features or software errors. This data about customer behavior and their problems with the products is a further source of information that is fed back into product development and improvement.

revised daily in response to customer testing. The customers were asked to perform tasks with the prototype, which was manipulated by one of the team in order to simulate interaction, e.g., changing screens. After half the sessions were conducted, the team produced a more formal version of the prototype in Adobe Illustrator. They found that customers were enthusiastic about using the paper prototype and were keen to offer improvements.

The second case study involved the development of a website for a video game publisher over three months. In order to understand what attracts people to such gaming sites, the multidisciplinary team felt they needed to understand the essence of gaming. To do this, they met 32 teenage gamers over a ten-day period, during which they observed and interviewed them in groups and individually. This allowed the team to understand something of the social nature of gaming and gave insights into the gamers themselves. During design, the team also conducted research and testing sessions in their office lab. This led them to develop new strategies and web designs based on the gamers' habits, likes, and dislikes.

Box 9.2 describes a situation in which users were asked to manage a software development project. There were hundreds of potential users, and so in addition,

DILEMMA Too Much of a Good Thing?

Involving users in development is a good thing. Or is it? And how much should they become involved? Box 9.2 describes a project in which users were appointed as project managers and were actively involved in development throughout. But are users qualified to lead a technical development project? And does this matter, provided there is sufficient technical expertise in the team?

Involving users at any level incurs costs, whether in terms of time for communication, or for workshops, or time spent explaining technical issues. Detailed user studies may also require the use of recording equipment and the subsequent cost of transcription and analysis. What evidence is there that user involvement is productive, or that it is worth putting the required level of resources into development? Research by Keil and Carmel (1995) indicates that the more successful projects do have direct links to users and customers. Kujala and Mäntylä (2000) performed some empirical work to investigate the costs and benefits of user studies early in product development. They concluded that user studies do in fact produce benefits that outweigh the costs of conducting them.

On the other hand, Heinbokel et al. (1996) suggest that a high user involvement has some negative effects. They found that projects with high user participation showed lower overall success, fewer innovations, a lower degree of flexibility, and low team effectiveness, although these effects were noticeable only later in the project (at least 6–12 months into the project). In short, projects with a high level of user participation tended to run less smoothly. They identified four issues related to communication among users and developers that they suggest caused problems. First, as the project progressed, users developed more sophisticated ideas, and they wanted them to be incorporated late in the project. Second, users were fearful of job losses or worsening job conditions and this led to a tendency for participation to be not constructive. Third, users were unpredictable and not always sympathetic to software development matters. For example, they asked for significant changes to be made just as testing was due to start. Fourth, user orientation in the designers may lead to higher aspirations and hence higher levels of stress.

Webb (1996) too has concerns about user involvement, but Scaife et al. (1997) suggest that it is not the fact of user involvement that is in question, but how and at what stage in development they should get involved.

BOX 9.2 Users as Project Team Leaders *(M880, 2000)*

The Open University (OU) in the UK is a large distance education university with many thousands of students enrolled each year in a variety of courses (undergraduate, graduate, vocational and professional) in a variety of subjects (Technology, Languages, Education, etc.). The courses are presented through paper-based and electronic media including video and audio tapes. It has a network of centers through which it supports and distributes courses to students throughout the UK and Europe. The OU employs about 3000 academic and other full-time staff and about 6000 part-time and counseling staff. In 1998/9 the university had over 200,000 students and customers for its education packs, and distributed materials throughout the UK and Europe to over 165,000 students. Such an operation requires considerable computerized support: in 1993 approximately 54 major systems of varying sizes were held on mainframe UN-LX host/workstations, VAX hosts, or PCs.

Traditionally, the systems had been built by an in-house software development team, who, due to resource constraints, sometimes needed to make business decisions although their expertise was in technical issues, not in the business side of the university. When it was time to re-develop these information systems, the OU decided that a new approach to development was required: users were to have a much more significant role.

Development was divided into a number of areas, each with its own project team and its own schedule. Consistency across the areas was maintained through the development of a GUI interface standard style guide that ensured that all systems had the same look and feel (style guides are discussed in Chapter 8). Users were involved in development on a number of different levels, typically 30–80% of their time. For example, in one area (Area E), one user was appointed full-time to manage the project team, two others joined the project team part-time for a limited period (about 18 months each), one user was consulted on a regular basis, and a wider set of users were involved through workshops and prototyping sessions. The project team also included technically trained analysts and developers.

When asked for the most successful and the least successful aspects of the project, both users and technical developers agreed that the most successful had been getting users involved in development. They said that this had made the system closer to what the users wanted. One user commented that, because users were part of the team for only a limited time, they did not see the development through from the beginning, but saw only some of the phases, and that this led to lack of continuity. Another user commented on the fact that the business had changed faster than the software could be developed, and hence the system had to be changed. The users' reactions were not all favorable, however. Another group of users who were consulted mainly through workshops and prototyping sessions did not feel that their needs had been adequately addressed.

One of the user project managers had this to say:

> The most successful thing has been getting people to go back to basics. We didn't look at existing systems and say, "We want the same thing but with go-faster stripes." We've examined what the University wants from the area. The most disappointing part has been that increased user involvement has not brought about ownership of the system by user areas. There was an expectation that we could move away from the traditional view of, "This is a computer system devised by computer people for you to use." In practice it's been far more difficult to get users to make decisions; they tend to say, "That's part of development. You decide."

This lack of ownership was commented upon by users and developers alike. One of the analysts commented:

> The user-led aspect has resulted in [the system's][1] greatest successes and greatest failures. User project managers do not have a systems background. Depending on their character they can be open to ideas or very blinkered. If they come from a user area with a system already it can be hard for them to see beyond their current system.

---

[1]When reporting raw data such as quotations anonymously, it is common practice to replace specific words or phrases that might compromise anonymity with similar words enclosed in square brackets to indicate that they are not the speaker's original words.

users became design team members on a full- and part-time basis; regular design workshops, debriefings, and training sessions were also held.

How actively users should be involved is a matter for debate. Some studies have shown that too much user involvement can lead to problems. This issue is discussed in the Dilemma box below.

9.3 What is a user-centered approach?

Throughout this book, we have emphasized the need for a user-centered approach to development. By this we mean that the real users and their goals, not just technology, should be the driving force behind development of a product. As a consequence, a well-designed system should make the most of human skill and judgment, should be directly relevant to the work in hand, and should support rather than constrain the user. This is less a technique and more a philosophy.

In 1985, Gould and Lewis (1985) laid down three principles they believed would lead to a "useful and easy to use computer system." These are very similar to the three key characteristics of interaction design introduced in Chapter 6.

1. ***Early focus on users and tasks.*** This means first understanding ***who*** the users will be by directly studying their cognitive, behavioral, anthropomorphic, and attitudinal characteristics. This required observing users doing their normal tasks, studying the nature of those tasks, and then involving users in the design process.
2. ***Empirical measurement.*** Early in development, the reactions and performance of intended users to printed scenarios, manuals, etc. is observed and measured. Later on, users interact with simulations and prototypes and their performance and reactions are observed, recorded, and analyzed.
3. ***Iterative design.*** When problems are found in user testing, they are fixed and then more tests and observations are carried out to see the effects of the fixes. This means that design and development is iterative, with cycles of "design, test, measure, and redesign" being repeated as often as necessary.

Iteration is something we have emphasized throughout these chapters on design, and it is now widely accepted that iteration is required. When Gould and Lewis wrote their paper, however, the iterative nature of design was not accepted by most developers. In fact, they comment in their paper how "obvious" these principles are, and remark that when they started recommending these to designers, the designers' reactions implied that these principles were indeed obvious. However, when they asked designers at a human factors symposium for the major steps in software design, most of them did not cite most of the principles—in fact, only 2% mentioned all of them. So maybe they had "obvious" merit, but were not so easy to put into practice. The Olympic Messaging System (OMS) (Gould et al., 1987) was the first reported large computer-based system to be developed using these three principles. Here a combination of techniques was used to elicit users' reactions to designs, from the earliest prototypes through to the final product. In this case, users were mainly involved in evaluating designs. The OMS is discussed further in Chapter 10.

The iterative nature of design and the need to develop usability goals have been discussed in Chapter 6. Here, we focus on the first principle, early focus on users and tasks, and suggest five further principles that expand and clarify what this means:

1. ***User's tasks and goals are the driving force behind the development.*** In a user-centered approach to design, while technology will inform design options and choices, it should not be the driving force. Instead of saying, "Where can we deploy this new technology?," say, "What technologies are available to provide better support for users' goals?"
2. ***Users' behavior and context of use are studied and the system is designed to support them.*** This is about more than just capturing the tasks and the users' goals. How people perform their tasks is also significant. Understanding behavior highlights priorities, preferences, and implicit intentions. One argument against studying current behavior is that we are looking to improve work, not to capture bad habits in automation. The implication is that exposing designers to users is likely to stifle innovation and creativity, but experience tells us that the opposite is true (Beyer and Holtzblatt, 1998). In addition, if something is designed to support an activity with little understanding of the real work involved, it is likely to be incompatible with current practice, and users don't like to deviate from their learned habits if operating a new device with similar properties (Norman, **1988).**
3. ***Users' characteristics are captured and designed for.*** When things go wrong with technology, we often say that it is our fault. But as humans, we are prone to making errors and we have certain limitations, both cognitive and physical. Products designed to support humans should take these limitations into account and should limit the mistakes we make. Cognitive aspects such as attention, memory, and perception issues were introduced in Chapter **3.** Physical aspects include height, mobility, and strength. Some characteristics are general, such as that about one man in 12 has some form of color blindness, but some characteristics may be associated more with the job or particular task at hand. So as well as general characteristics, we need to capture those specific to the intended user group.
4. ***Users are consulted throughout development from earliest phases to the latest and their input is seriously taken into account.*** As discussed above, there are different levels of user involvement and there are different ways in which to consult users. However involvement is organized, it is important that users are respected by designers.
5. ***All design decisions are taken within the context of the users, their work, and their environment.*** This does not necessarily mean that users are actively involved in design decisions. As you read in Gillian Crampton Smith's interview at the end of Chapter 6, not everyone believes that it is a good idea for users to be designers. As long as designers remain aware of the users while

making their decisions, then this principle will be upheld. Keeping this context in mind can be difficult, but an easily accessible collection of gathered data is one way to achieve this. Some design teams set up a specific design room for the project where data and informal records of brainstorming sessions are pinned on the walls or left on the table. (This is discussed again in Section 9.4.2 on Contextual Design.)

ACTIVITY 9.1 Assume that you are involved in developing a new e-commerce site for selling garden plants. Suggest ways of applying the above principles in this task.

Comment To address the first three principles, we would need to find out about potential users of the site. As this is a new site, there is no immediate set of users to consult. However, the tasks and goals, behavior, and characteristics of potential users of this site can be identified by investigating how people shop in existing online and physical shopping situations—for example, shopping through interactive television, through other online sites, in a garden center, in the local corner shop, and so on. For each of these, you will find advantages and disadvantages to the shopping environment and you will observe different behaviors. By investigating behavior and patterns in a physical garden center, you can find out a lot about who might be interested in buying plants, how these people choose plants, what criteria are important, and what their buying habits are. From existing online shopping behavior, you could determine likely contexts of use for the new site.

For the fourth principle, because we don't have an easily tapped set of users available, we could follow a similar route to the Internet company described in Section 9.2, and try to recruit people we believe to be representative of the group. These people may be involved in workshops or in evaluation sessions, possibly in a physical shopping environment. Valuable input can be gained in targeted workshops, focus groups, and evaluation sessions. The last principle could be supported through the creation of a design room to house all the data collected.

"We created this model to appeal to the youth market. The monitor is tattooed and the CD-ROM tray is pierced with a gold earring."

9.4 Understanding users' work: applying ethnography in design

Kuhn (1996) provides a good example illustrating the importance of understanding users' work. She describes a case where a computer system was introduced to cut down the amount of time spent on conversations between telephone-company repair personnel. Such conversations were regarded as inefficient and "off-task." What management had failed to realize was that in the conversations workers were often consulting one another about problems, and were pooling their knowledge to solve them. By removing the need for conversation, they removed a key mechanism for solving problems. If only the designers had understood the work properly, they would not have considered removing it.

Ethnography is a method that comes originally from anthropology and literally means "writing the culture" (Hammersley and Atkinson, 1983). It has been used in the social sciences to display the social organization of activities, and hence to understand work. It aims to find the order within an activity rather than impose any framework of interpretation on it. It is a broad-based approach in which users are observed as they go about their normal activities. The observers immerse themselves in the users' environment and participate in their day-to-day work, joining in conversations, attending meetings, reading documents, and so on. The aim of an ethnographic study is to make the implicit explicit. Those in the situation, the users in this case, are so familiar with their surroundings and their daily tasks that they often don't see the importance of familiar actions or happenings, and hence don't remark upon them in interviews or other data-gathering sessions.

There are different ways in which this method can be associated with design. Beynon-Davies (1997) has suggested that ethnography can be associated with development as "ethnography *of*," "ethnography for," and "ethnography within." Ethnography of development refers to studies of developers themselves and their workplace, with the aim of understanding the practices of development (e.g. Button and Sharrock, 1994; Sharp et al., 1999). Ethnography for development yields ethnographic studies that can be used as a resource for development, e.g., studies of organizational work. Ethnography within software development is the most common form of study (e.g., Hughes et al., 1993a); here the techniques associated with ethnography are integrated into methods and approaches for development (e.g., Viller and Sommerville, 1999).

Because of the very nature of the ethnographic experience, it is very difficult to describe explicitly what data is collected through such an exercise. It is an experience rather than a data-collection exercise. However, the experience must be shared with other team members, and therefore needs to be documented and rationalized. Box 9.3 provides an example ethnographic account in the form of a description of an ethnographic study of a new media company. In this case, the intention was not explicitly concerned with designing an interactive product, but was a business-oriented ethnography. The style and content of the piece, however, are typical of ethnographies.

Studying the context of work and watching work being done reveals information that might be missed by other methods that concentrate on asking about work away from its natural setting. For example, it can shed light on how people do the "real" work as opposed to the formal procedures that you'd find in documentation;

BOX 9.3 An Example Ethnography
(printed with the permission of Fiona Hovenden, the ethnographer)

Background I was asked to design a retreat for a new media company. They were about to shift from working in a very open, unstructured way to formalizing their working processes. The main reason for this was that they had signed a deal with a large organization that was going to act as a financial patron in return for first options on the new media company's ideas and designs.

This proposed shift was causing some tension and anxiety within the company, with people feeling that their current working practices worked very well and that imposing structure would stifle the creativity on which their work depended.

Method Over a four-day period, I had a desk in the office and observed the rhythm of work and the working practices. I spent two days just observing, and then I conducted one-to-one face-to-face interviews with every member of the company, and one-to-one email interviews with the three people from the patron organization who were coming to join the company.

The account (excerpted here) is my notes, built up over the four-day period. The structure, and the content are built up iteratively. For example, on the surface the company appeared very collectivist—everyone in the company treated everyone else as a peer, people invited other people to work with them on projects, nobody seemed to tell anyone else what to do. But during the interviews it became clear that everyone waited for the opinion of the leader before doing anything. In fact, on the surface it looked as though there were three joint partners, but the entire company implicitly and explicitly deferred to one of them. So, although in the account the surface appearance is what I first noted, the interviews indicated that there was no attempt at consensus.

Brief Characterization of User Community This is an apparently loose aggregation of artists, artist-technicians, information designers, producers, and a small, nontraditional operations team. There is a commitment to an open, collectivist way of working that seems to translate as anyone can say anything or ask for anything, and they will be given a hearing. However, the way things actually get done does not seem to be by consensus. There are obvious and accepted loci of power associated with particular individuals. It is these individuals who bestow a hearing.

The commitment to collectivism is currently undergoing a shift, as a new patron relationship requires a more formal business structure. A business focus will also be a more explicit part of the new community life. There is a shifting power structure, in which [current leader] will be joined by [X] and [Y] from the patron organization at a more formal managerial level. They will become the gateway through which all project ideas will have to pass, and will also control the financing of projects. They will therefore have a great deal of power, while the power available to the collective will be the power to persuade/seduce.

Community Practices and Productions This community creates new media products. The nature of this work means that there is a strong visual bias and a highly developed visual sophistication.

One of the original motives for the formation of the company was to explore non-traditional narratives. The lead information designer has also described the work the company does as "story telling." Both in the ongoing championing of the exploration of narrative and in the actual client work done, the community is practiced at the visual presentation and production of stories. This (dominant) aspect of work could be described as translating speech to visuals.

Client projects seem primarily visual, sound is not the focus. In terms of workspace noise, sound bleeds in from the surrounding environment. Public music is played sporadically, and usually by [designer], who has brought in a CD player that sits on his desk. However, many community members wear headphones for significant periods of their time at work.

Many of the working practices seem informal and fluidly structured. This is according to the accounts given by community members in interviews, and the specific request from [leader] to make formalizing working practices a major part of the retreat. More information on working practices is not covered by the available data. However, there is one community practice I would like to briefly mention here. I attended one Tuesday morning production meeting. This seems to be the one set time in the week when everyone in the community comes together to create a community-wide status report. However, not everyone showed up. Meetings are possibly the most significant rituals of modern business practice, and seem to work best when the form and function of the ritual is known, understood, and felt to be relevant by everyone involved. The most startling aspect of the production meeting to me was that individuals seemed to decide when the meeting was over and left the room accordingly, without marking the fact in any other way, say by announcing their departure. The effect (to me) was of the meeting dribbling away, which seemed to lessen its importance.

the nature and purposes of collaboration, awareness of other's work, and implicit goals that may not even be recognized by the workers themselves. For example, Heath et al. (1993) have been exploring the implications of ethnographic studies of real-world settings for the design of cooperative systems. We described their underground control room study in Chapter 4, but they have also studied medical centers, architects' practices, and TV and radio studios.

In one of their studies Heath et al. (1993) looked at how dealers in a stock exchange work together. A main motivation was to see whether proposed technological support for market trading was indeed suitable for that particular setting. One of the tasks examined in detail was the process of writing tickets to record deals. It had been commented upon earlier by others that this process of deal capture, using "old-fashioned" paper and pencil technology, was currently time-consuming and prone to error. Based on this finding, it had been further suggested that the existing way of making deals could be improved by introducing new technologies, including touch screens to input the details of transactions, and headphones to eliminate distracting external noise.

However, when Heath et al. began observing the deal capture in practice, they quickly discovered that these proposals were misguided. In particular, they warned that these new technologies would destroy the very means by which the traders currently communicate and keep informed of what others are up to. The touch screens would reduce the availability of information to others on how deals were progressing, while headphones would impede the dealers' ability to inadvertently monitor one another's conversations. They pointed out how this kind of peripheral monitoring of other dealers' actions was central to the way deals are done. Moreover, if any dealers failed to keep up with what the other dealers were doing by continuously monitoring them, it was likely to affect their position in the market, which ultimately could prove very costly to the bank they were working for.

Hence, the ethnographic study proved to be very useful in warning against attempts to integrate new technologies into a workplace without thinking through the implications for the work practice. As an alternative, Heath et al. suggested pen-based mobile systems with gestural recognition that could allow deals to be made efficiently while also allowing the other dealers to continue to monitor one another unobtrusively.

Hughes et al (1993) state that "doing" ethnography is about being reasonable, courteous and unthreatening, and interested in what's happening. This is particularly important when trying to perform studies in people's homes, such as those described in Box 9.4. There is, of course, more to it than this. Training and practice are required to produce good ethnographies.

BOX 9.4 Ethnographies of the Home

Home use of technology such as the personal computer, wireless telephones, cell phones, remote controls, and so on has grown over the last decade. Although consumer surveys and similar questionnaires may be able to gather some information about this market, ethnographic studies have been used to gain that extra insight that ensures that products do not just perform needed functions but are also pleasurable and easy to use.

Dray and Mrazek (1996) report on an international study of families' use of technology in which they visited 20 families in America, Germany and France. They spent at least four hours in each of the homes, talking with all members of the family, including children of all ages, about their use of computer technology. They give no details of the data collected, but assert that the study was extremely useful, that "there is no substitute for contextual studies," and that the results have influenced many design decisions and specifications for new products. One aspect of the study they emphasize is the need to develop a rapport with the family. They focused their attention on building a strong positive rapport in the first few minutes of the visit. In all cases, they used food as an icebreaker, by either bringing dinner with them for themselves and the family, or by ordering food to be delivered. This provided a mundane topic of conversation that allowed a natural conversation to be held.

After dinner, they moved to the location of the computer and began by asking the children about their use of the technology. Each family member was engaged in conversation about the technology, and printed samples of work were gathered by the researchers. A protocol designed by the marketing and engineering departments of the company was used to guide the conduct of this part of the study, but after all of the protocol had been covered, families were encouraged to discuss topics they were interested in. Immediately after a visit, the team held a formal debriefing session during which all photos, videotapes, products, and notes were reviewed and a summary debriefing questionnaire was completed. A thank-you letter was later sent to the families.

From this description you can see that a huge amount of preparation was required in order to ensure that the study resulted in getting the right data, i.e., in collecting data that was going to answer the relevant questions.

Mateas et al. (1996) report on a pilot ethnographic study that was also aimed at informing the design and development of domestic computing systems. They visited ten families and also emphasize the importance of making families feel comfortable with them. In their study, this was partly achieved by bringing a pizza dinner for everyone. After dinner, the adults and the children were separated. The researchers wanted to get an understanding of a typical day in the home. To do this, each family member was asked to walk through a typical day, using a felt board with a layout of their house, and felt rooms, products, activities, and people that could be moved around on the felt house.

From their work they derived a model of space, time, and social communication that differed from the model implied by the standard PC. For example, the standard PC is designed to be used in one location by one user for long periods of uninterrupted time. The studies revealed that on the other hand, family activity is distributed throughout multiple spaces, is rarely conducted alone, and is not partitioned into long periods of uninterrupted use. In addition, the PC does not support communication among co-located members of the family, which is a key element of family life. They conclude that small, integrated computational appliances supporting multiple co-located users are more appropriate to domestic activity than the single PC.

Collecting ethnographic data is not hard although it may seem a little bewildering to those accustomed to using a frame of reference to focus the data collection rather than letting the frame of reference arise from the available data. You collect what is available, what is "ordinary," what it is that people do, say, how they work. The data collected therefore has many forms: documents, notes of your own, pictures, room layouts. Notebook notes may include snippets of conversation and descriptions of rooms, meetings, what someone did, or how people reacted to a situation. It is opportunistic in that you collect what you can collect and make the most of opportunities presented to you. You don't go in with a firm plan, and so the data you collect is not specifiable in advance. You have to do it rather than read about it. What you record can become more focused after being in the field for a while.

ACTIVITY 9.2 Look up from reading this book and observe your surroundings. Wherever you are, the chances are that you can see and hear lots of things, and probably other people too. Start to make a list of what you observe, and when things change or people move, write down what has happened and how it happened. For example, if someone spoke, what did his voice sound like? Angry, calm, whispering, happy? Spend just a few minutes observing what you can see.

Now think about the same observations but begin to interpret them: imagine that you have to place the main items or people that you can see into categories. For example, on a train you might consider who might be getting off at which station, in a bedroom you might think about how to tidy up the items lying around.

How easy is it to go from the detailed description to the more abstracted one?

Comment

As I am writing this, I am in a room on my own. I therefore don't have people to observe, but my desk is covered with things: a pen, a boarding pass from a recent trip abroad, a rosette from a parcel wrapping, and many books, papers, disks etc. If I look around then I can see the wallpaper and the curtains, clothes hanging and in piles on the bed. In the background I can hear cars moving along the road, and the television downstairs. To spend any length of time really describing any one of the things I observe would take up a lot of words, and that's a lot of data.

If I now consider how to file the things I can see, then I would start to think of categories such as which are books, which are research papers, what can be thrown away, and so on. It becomes easier to feel like I'm making progress. The other thing to notice is that some things I can observe are blocked out of my sphere of interest, such as the cars outside.

In some ways, the goals of design and the goals of ethnography are at opposite ends of a spectrum. Design is concerned with abstraction and rationalization. Ethnography, on the other hand, is about detail. An ethnographer's account will be concerned with the minutiae of observation, while a designer is looking for useful abstractions that can be used to inform design. One of the difficulties faced by those wishing to use this very powerful technique is how to harness the data gathered in a form that can be used in design.

Below, we introduce one framework that has been developed specifically to help structure the presentation of ethnographies in a way that enables designers to use them (other frameworks to help orient observers and how to organize this kind

DILEMMA What To Lose When You Abstract?

In Chapter 7, we discussed the need for data interpretation and analysis. This involves structuring and abstracting from the data, so that important aspects of a situation can be reasoned about at a higher level of generalisation without getting bogged down in details. It is inevitable that when moving from a more detailed description to a more abstract one, information will be lost. But what is important and what is irrelevant? This is a key question to answer if ethnographic data is to be used to inform design.

of study are described in Chapter 12). This framework has three main dimensions (Hughes et al, 1997):

1. The ***distributed co-ordination*** dimension focuses on the distributed nature of the tasks and activities, and the means and mechanisms by which they are co-ordinated. This has implications for the kind of automated support required.
2. The ***plans and procedures*** dimension focuses on the organizational support for the work, such as workflow models and organizational charts, and how these are used to support the work. Understanding this aspect impacts on how the system is designed to utilize this kind of support.
3. The ***awareness of work*** dimension focuses on how people keep themselves aware of others' work. No-one works in isolation, and it has been shown that being aware of others' actions and work activities can be a crucial element of doing a good job. In the stock market example described above, this was one aspect that ethnographers identified. Implications here relate to the sharing of information.

Rather than taking data from ethnographers and interpreting this in design, an alternative approach is to train developers to collect ethnographic data themselves. This has the advantage of giving the designers first-hand experience of the situation. Telling someone how to perform a task, or explaining what an experience is like, is very different from showing them or even gaining the experience themselves. Finding people with the skills of ethnographers and interaction designers may be difficult, but it is possible to provide notational and procedural mechanisms to allow designers to gain some of the insights first-hand. The two methods described below provide such support.

9.4.1 Coherence

The Coherence method (Viller and Sommerville, 1999) combines experiences of using ethnography to inform design with developments in requirements engineering. Specifically, it is intended to integrate social analysis with object-oriented analysis from software engineering (which includes producing use cases as described in Chapter 7). Coherence does not prescribe how to move from the social analysis to use cases, but claims that presenting the data from an ethnographic study based around a set of "viewpoints" and "concerns" facilitates the identification of the product's most important use cases.

Viewpoints and concerns

Coherence builds upon the framework introduced above and provides a set of focus questions for each of the three dimensions, here called "viewpoints". The focus questions (see Figure 9.1) are intended to guide the observer to particular aspects of the workplace. They can be used as a starting point to which other questions may be added as experience in the domain and the method increases.

In addition to viewpoints, Coherence has a set of concerns and associated questions. Concerns are a kind of goal, and they represent criteria that guide the requirements activity. These concerns are addressed within each appropriate viewpoint. One of the first tasks is to determine whether the concern is indeed relevant to the viewpoint. If it is relevant, then a set of elaboration questions is used to explore the concern further. The concerns, which have arisen from experience of using ethnography in systems design, are:

1. ***Paperwork and computer work.*** These are embodiments of plans and procedures, and at the same time are a mechanism for developing and sharing an awareness of work.
2. ***Skill and the use of local knowledge.*** This refers to the "workarounds" that are developed in organizations and are at the heart of how the real work gets done.

Distributed coordination

- How is the division of labor manifest through the work of individuals and its coordination with others?
- How clear are the boundaries between one person's responsibilities and another's?
- What appreciation do people have of the work/tasks/roles of others?
  How is the work of individuals oriented towards the others?

Plans and procedures

How do plans and procedures function in the workplace?
- Do they always work?
- How do they fail?
  What happens when they fail?
- How, and in what situations, are they circumvented?

Awareness of work

- How does the spatial organization of the workplace facilitate interaction between workers and with the objects they use?
- How do workers organize the space around them? Which artifacts that are kept to hand are likely to be important to the achievement of everyday work?
- What are the notes and lists that the workers regularly refer to?
- What are the location(s) of objects, who uses them, how often?

Figure 9.1 Focus questions for the three viewpoints.

Paperwork and computer work

- How do forms and other artifacts on paper or screen act as embodiments of the process?
- To what extent do the paper and computer work make it clear to others what stage people are at in their work?
- How flexible is the technology at supporting the work process—is a particular process enforced, or are alternatives permitted?

Skill and the use of local knowledge

- What are the everyday skills employed by individuals and teams in order to get the work done?
- How is local knowledge used and made available, e.g., through the use of personalized checklists, asking experts, etc.?
- To what extent have standard procedures been adapted to take local factors into account?

Spatial and temporal organization

- How does the spatial organization of the workplace reflect how the work is performed?
- Which aspects of the work to be supported are time-dependent?
- Does any data have a "use-by-date"?
- How do workers make sure that they make use of the most up-to-date information?

Organizational memory

- How do people learn and remember how to perform their work?
- How well do formal records match the reality of how work is done?

Figure 9.2 Elaboration questions for the four concerns.

3. ***Spatial and temporal organization.*** This concern looks at the physical layout of the workplace and areas where time is important.
4. ***Organizational memory.*** Formal documents are not the only way in which things are remembered within an organization. Individuals may keep their own records, or there may be local gurus.

The elaboration questions associated with these concerns are listed in Figure 9.2 and a sample social concern from the air traffic control domain, together with resultant requirements, is shown in Figure 9.3.

9.4.2 Contextual Design

Contextual Design is another technique that was developed to handle the collection and interpretation of data from fieldwork with the intention of building a software-based product. It provides a structured approach to gathering and representing information from fieldwork such as ethnography, with the purpose

Paperwork and computer work

Flight strips embody the process of an aircraft's progress through the sector of airspace controlled by a suite. As an aircraft approaches the sector, its strip is moved progressively to the bottom of the rack until it becomes the current strip for the controller to deal with. The work of the controller can therefore be viewed in terms of dealing with the flow of strips as aircraft enter, traverse, and leave the controller's sector.

The collection of strips in various racks in a suite provide an 'at a glance' means of determining the current and future workload of a particular controller. The practice of 'cocking out' strips, i.e., raising them slightly in the racks, informs the controller that there is something non-standard about the flight concerned. This may be done by the assistant controller when inserting the strip, or by the controller as a reminder. Glancing at the strips provides a controller with an indication of their current and future workload, in the same way as it allows other controllers to see the relative loading on other sectors. This feature of the organization of the strips is used in particular at change over of shifts, where the incoming controller will spend up to 10 minutes looking over the shoulder of the out-going controller in order to 'get the picture' of the current state of the sector.

Flight strips provide incredibly flexible support for the work of controllers. Different practices exist regarding whether strips are placed into the racks in a top to bottom sequence or vice versa. All instructions given by controllers to pilots, and the pilots' acknowledgements, are recorded onto the relevant flight strip. These annotations are made using a standard set of symbols, and different coloured pens according to the annotator's role within the controlling team. In this way, flight strips constitute a record of a flight's progress through a sector.

Requirement 1. The system shall support controllers 'getting the picture' by providing the ability to determine current and future load for a sector 'at a glance'

Requirement 2. The system shall provide a facility to mark exceptional or non-standard flights requiring special attention

Requirement 3. Annotations to flight records shall be recorded and presented in such a way that they identify the person who made them.

Figure 9.3 Elaboration of paperwork and computer work.

of feeding it into design. It has been used on a number of projects, e.g., see Box 9.5.

Contextual Design has seven parts: Contextual Inquiry, Work Modeling, Consolidation, Work Redesign, User Environment Design, Mockup and Test with Customers, and Putting It into Practice. In this chapter we are focusing on understanding users' work, and so shall discuss only the first three steps. Step 4 involves changing work practices, which is outside our scope here. Step 5 produces a prototype that is used with customers, and the final step concerns the practicality of the working system. The activities involved in these last two steps have been discussed in general terms in Section 8.2.

Contextual inquiry

Contextual inquiry is an approach to ethnographic study used for design that follows an apprenticeship model: the designer works as an apprentice to the user. The

BOX 9.5 Using Contextual Design for Office Products

Page (1996) reports on the use of Contextual Design in customer research for a new version of the word processor WordPerfect. The company already had some experience of field research, since the initial version of WordPerfect had been based on informal user observation and user testing, although it wasn't seen as such at the time.

The scope of this study was quite wide, with the team wanting to learn about "the making of documents": how they were conceived, created, reviewed, approved, and distributed. To cover this scope, the team was multidisciplinary, involving expertise in word-processor development, human factors, documentation, marketing, and usability. Contextual Design was chosen because it leads the team systematically through the data-gathering and interpretation activities to product design.

The team undertook three weeks of training, organized as one week of training, four weeks of work, one week of training, four weeks of work, etc. Users were chosen carefully so as to reflect different types, including those who use the existing version of the product and those who don't use computers at all, and to be sure that they were representative of the company's main client base. The set of users was refined as data collection progressed and it became obvious where gaps were, and what kinds of user were needed to fill them.

Even though the intentions of the researchers had been communicated to the collaborators, they often arrived at sites to find that a focus group had been arranged rather than an opportunity for observation. Also, some people thought that the researchers were there to help solve their software problems and expected them to spend time on this rather than on data collection. Observing people at higher levels of management proved difficult at times, despite arranging visits well in advance. The result of this was that they collected more data about support roles, such as administrators and secretaries, than about others.

All team members took part in observation, and interviews were conducted in the worker's workplace, as laid down in the Contextual Design method. Generally, the interviews were taped, although if the data was interpreted within 24 to 48 hours there was no need to listen to the tape again.

Data was interpreted with the entire team. The observer would review her notes while other team members asked questions to draw out information. One team member was charged with writing down each important factor identified by the team, and others were responsible for drawing the workflow, sequence, physical, and context models. It was felt, however, that the contextual model did not represent the cultural influences in any useful way, and so it was not used. To structure the data, they used the affinity model, consolidated models, redesigned work models, user environment model, and user interface designs. A portion of the affinity diagram is shown in Figure 9.4.

When producing the redesigned work model, the goal was to streamline processes and eliminate breakdowns. Any emerging technologies that might help in this were identified and studied. For the user interface designs, paper prototypes were developed early on and tested with users, and as concepts became more certain, running prototypes were created in ToolBook and Delphi.

The researchers knew they might have problems in selling the idea to the implementors. To overcome this, they started having open days for the rest of the company as soon as they had their first affinity diagram and consolidated models. In some cases, members of the development teams were on the Contextual Design teams; where developers were not on the teams, they were invited to contribute to the design ideas before products were in their final form. This involvement helped to increase the developers' sense of ownership.

Page (1996) gives two examples of actual features that were a result of their fieldwork: "Make It Fit" and "QuickTasks." Make It Fit is a feature in WordPerfect 6.1 for Windows that takes the text and makes it fit into the available space, either by expanding it to fill blank space or by shrinking it. QuickTasks is a feature of PerfectOffice 3.0 that automates a series of steps across multiple applications, prompting users for information as it is needed.

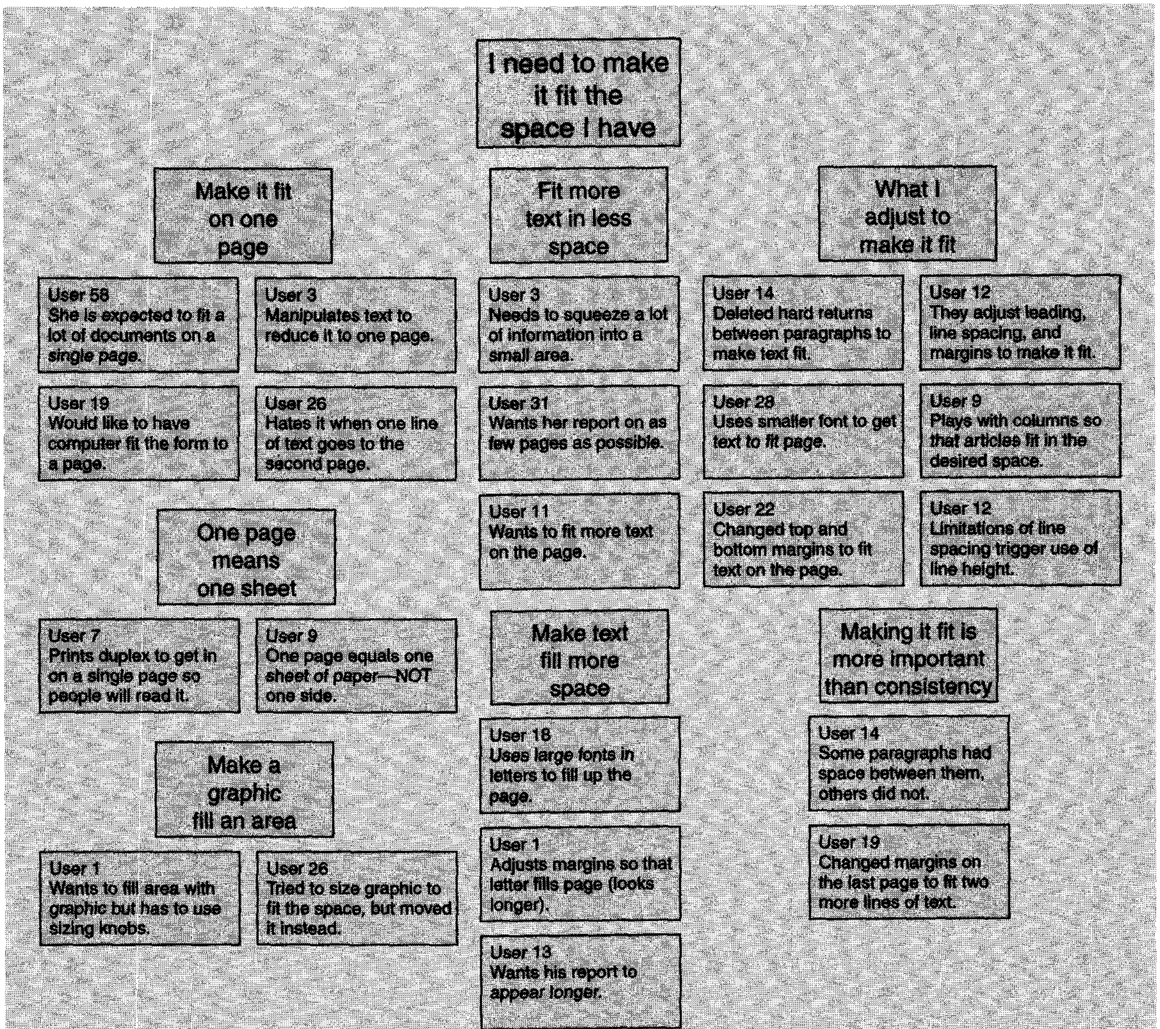


Figure 9.4 A portion of the affinity diagram for the WordPerfect fieldwork studies.

most typical format for contextual inquiry is a contextual interview, which is a combination of observatfbn, discussion, and reconstruction of past events. Contextual inquiry rests on four main principles: context, partnership, interpretation and focus.

The context principle emphasizes the importance of going to the workplace and seeing what happens. The partnership principle states that the developer and the user should collaborate in understanding the work; in a traditional interviewing or workshop situation, the interviewer or workshop leader is in control, but in contextual inquiry the spirit of partnership means that the understanding is developed through cooperation.

The interpretation principle says that the observations must be interpreted in order to be used in design, and this interpretation should also be developed in cooperation between the user and the developer. For example, I have a set of paper cards stuck on my screen at work. They are covered in notes; some list telephone numbers and some list commands for the software I use. Someone coming into my office might interpret these facts in a number of ways: that I don't have access to a telephone directory; that I don't have a user manual for my software; that I use the software infrequently; that the commands are particularly difficult to remember. The best way to interpret these facts is to discuss them with me. In fact, I do have a telephone directory, but I keep the numbers on a note to save me the trouble of looking them up in the directory. I also have a telephone with a memory, but it isn't clear to me how to put the numbers in memory, so I use the notes instead. The commands are there because I often forget them and waste time searching through menu structures.

The fourth principle, the focus principle, was touched upon above in our discussion of ethnography and was also addressed in Coherence: how do you know what to look for? In contextual inquiry, it is important that the discussion remains pertinent for the design being developed. To this end, a project focus is established to guide the interviewer, which will then be augmented by the individual's own focus that arises from their perspective and background. The contextual inquiry interview differs from ethnographic studies in a number of ways:

1. It is much shorter than a typical ethnographic study. A contextual inquiry interview lasts about two or three hours, while an ethnographic study tends to be longer, probably weeks or months.
2. The interview is much more intense and focused than an ethnographic study, which takes in a wide view of the environment.
3. In the interview, the designer is not taking on a role of participant observer, but is inquiring about the work. The designer is observing, and is questioning behavior, but is not participating.
4. In the interview, the intention is to design a new system, but when conducting an ethnography, there is no particular agenda to be followed.

ACTIVITY 9.3 How does the contextual inquiry interview compare with the interviews introduced in Chapter 7?

Comment We introduced structured, unstructured, and semi-structured interviews in Chapter 7. Contextual inquiry could be viewed as an unstructured interview, but is more wide-ranging than this. The interviewer does not have a set list of questions to ask, and can be guided by the interviewee. Contextual inquiry, however, is to be conducted at the interviewee's place of work, while normal work continues. It incorporates other data-gathering techniques such as observation although other interviews too may be used in conjunction with other techniques.

Normally, each team member conducts at least one contextual inquiry session. Data is collected in the form of notes and perhaps audio and video recording, but a lot of information is in the observer's head. It is important to review the experience and to start documenting the findings as soon as possible after the session. Contextual Design includes an interpretation session in which a number of models are generated (see below). Figures 9.5 to 9.8 show flow, sequence, cultural, and physical models focused around the system manager of an organization (Holtzblatt and Beyer, 1996).

Work Modeling

For customer-centered design, the first task of a design team is to shift focus from the system that the team is chartered to build and redirect it to the work of potential customers. Work, and understanding work becomes the primary consideration. But "work" is a slippery concept. What is work? (Beyer and Holtzblatt, 1998, p. 81)

Contextual design identifies five aspects to modeling "work," each of which guides the team to take a different perspective on what they have observed:

- The ***workflow model*** (Figure ***9.5)*** represents the people involved in the work and the communication and coordination that takes place among them in order to achieve the work.

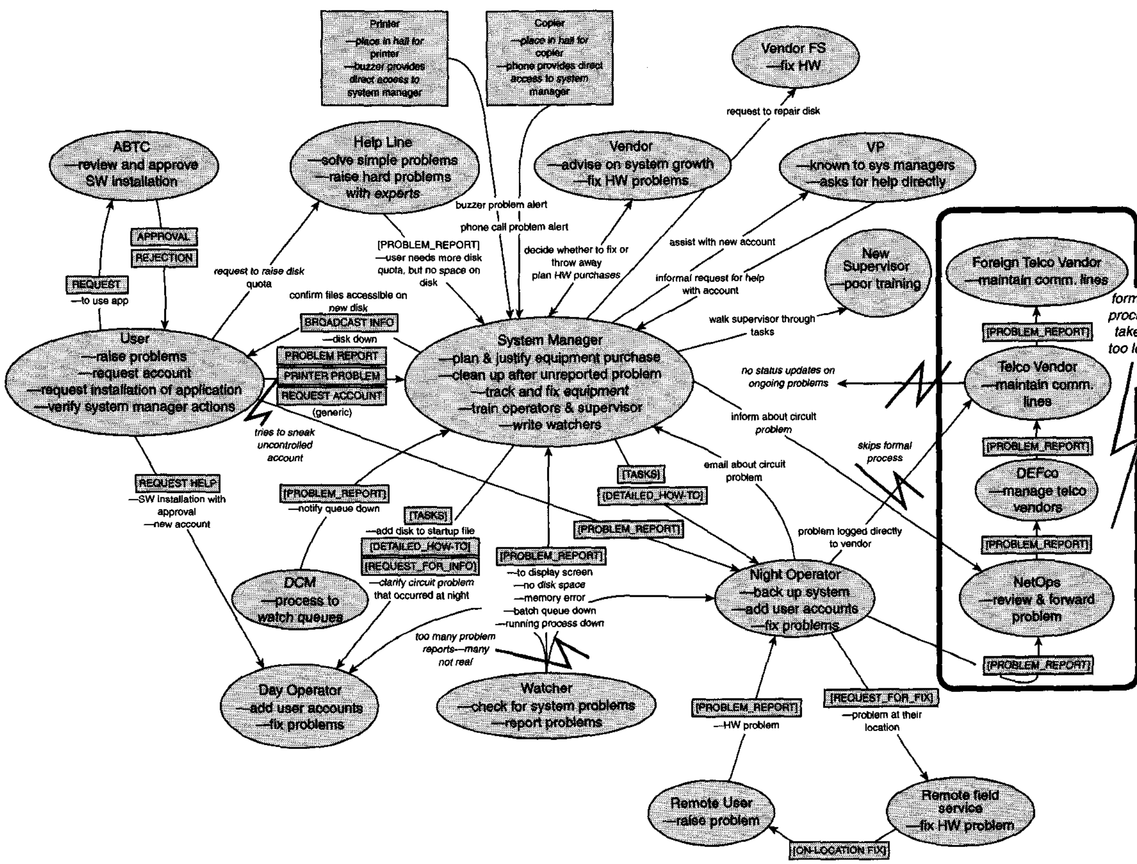


Figure 9.5 An example work flow model.

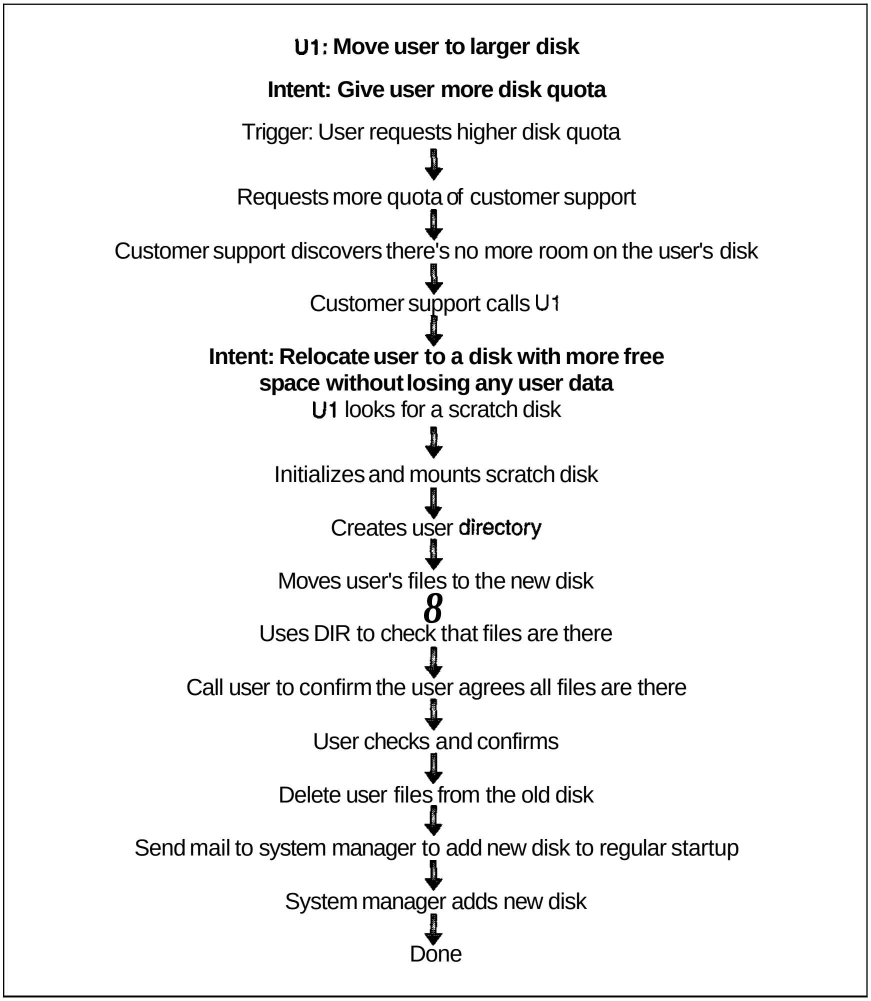


Figure 9.6 An example sequence model.

- The ***sequence model*** (Figure 9.6) shows the detailed work steps necessary to achieve a goal. Sequences are collected during the contextual interview, as the user works. However, understanding the steps alone is not sufficient, since although you may be able to streamline the steps themselves, if you do not understand the goals you may create a nonsensical work sequence. The sequence model also states the trigger for the set of steps.
- The ***artifact model*** represents the physical things created to do the work, such as the sticky notes at my desk, described above. The model consists of an annotated picture (or drawing) of each significant physical artifact used in achieving the work.
- The ***cultural model*** (Figure 9.7) represents constraints on the system caused by organizational culture. Organizations have cultures, teams build up their

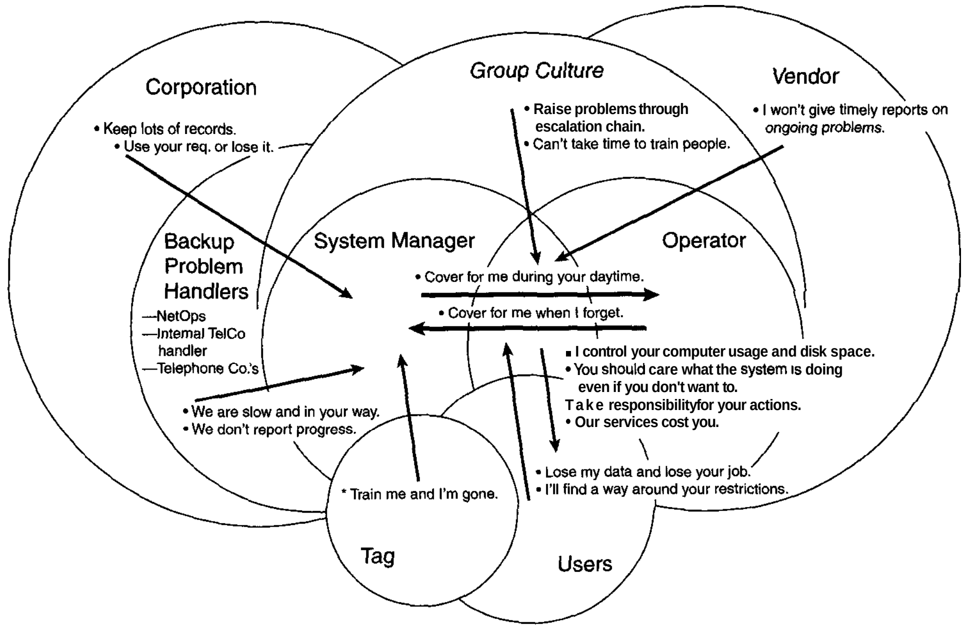


Figure 9.7 **An** example cultural model.

own culture, and work is performed in a cultural context. Culture influences the values and beliefs held by those taking part in the culture, and it determines rituals, expectations, and behavior. As a simple example, consider the dress codes for different situations in which you may find yourself. If you turn up at a baseball game in a three-piece suit, people will think you're a bit odd. On the other hand, if you turn up at a formal dinner in jeans and T-shirt, you will be refused entry. The cultural model aims to identify the main influencers on work, i.e., people or groups who constrain or affect work in some way.

- The physical model (Figure 9.8) shows the physical structure of the work. It may be a physical plan of the users' work environment, e.g., the office, or it may be a schematic of a communications network showing how components are linked together. The model captures the physical characteristics that constrain work and may make some work patterns infeasible.

The interpretation session

The work models are captured during an interpretation session. The team has to build an agreed view of the customers, their work, and the system to be built. Each developer therefore has to communicate to all the others on the team everything learned from her own interviewing experiences. So, after a contextual inquiry interview has been conducted, the team comes together to produce one consolidated view of the users' work.

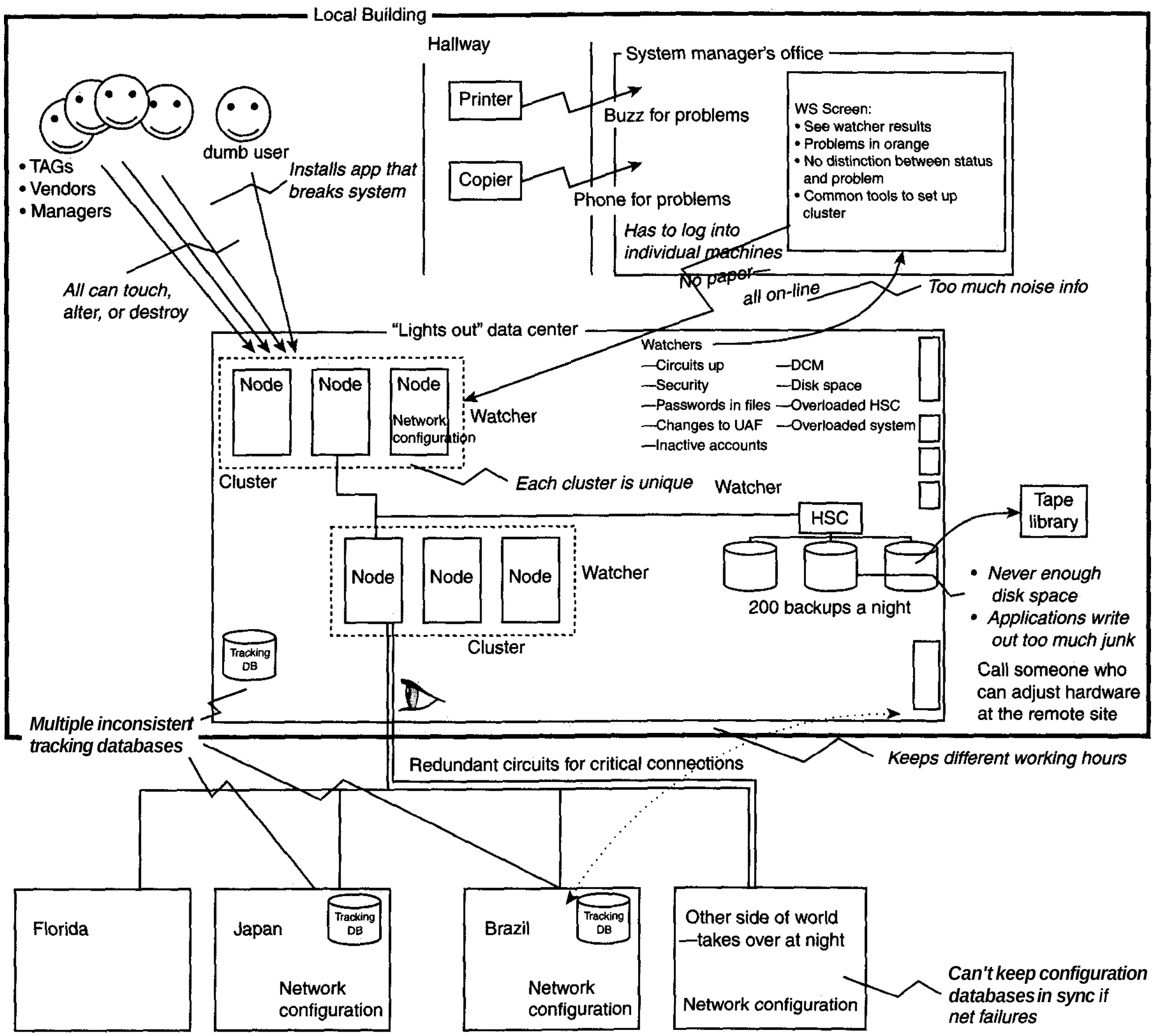


Figure 9.8 An example physical model.

Certain roles need to be adopted by the participants of this session. The interviewer is the person who has conducted the interviews and whose models are being examined. He must describe to the team what happened and in what order. During this recounting, the other members of the team can question the interviewer for clarification and extra information. Work modelers draw the work models as they emerge from the description given by the interviewer. The recorder keeps notes of the interpretation session that provide a sequential record of the meeting. The rest of the team (participants) listen to the description, ask questions, suggest design ideas (which are noted and not discussed at this time), observe, and contribute to the building of the models. The moderator stage-manages the meeting, keeps discussions focused on the main issue, keeps the pace of the meeting brisk, encourages everyone to take part, and notes where in the story the interviewer was in case of interruptions. The ***rat-hole watcher*** steers the conversation away from any distractions.

The output from this session is a set of models associated with the particular contextual inquiry interview. Each contextual inquiry interview generates its own set of models that is inevitably focused on the interviewee. These sets of models must be consolidated to gain a more general view of the work as described below.

ACTIVITY 9.4 The thick lightning marks in the flow models represent points at which breakdowns in communication or coordination occur. Alongside each lightning bolt is a description of the cause for this breakdown. Study the flow model in Figure 9.5 and identify all the breakdowns and their causes.

Comment There are five breakdowns:

(a) too many problem reports—many not real

(b) the flow "problem logged directly to vendor" skips the formal process.

(c) no status updates on ongoing problems

(d) formal process takes too long

(e) tries to sneak uncontrolled account

Consolidating the models

The affinity diagram (see Figure 9.9) aims to organize the individual notes captured in the interpretation sessions into a hierarchy showing common structures and themes. Notes are grouped together because they are similar in some fashion. The groups are not predefined, but must emerge from the data. The process was originally introduced into the software quality community from Japan, where it is regarded as one of the seven quality processes. The affinity diagram is constructed after a cross-section of users has been interviewed and the corresponding interpretation sessions completed.

The affinity diagram is built by a process of induction. One note is put up first, and then the team searches for other notes that are related in some way.

The models produced during the interpretation session need to be consolidated so as to get a more general model of the work, one that is valid across individuals. The primary aim in consolidating flow models is to identify key roles. Any one individual may take on more than one role, and so it is necessary to identify and compare roles across and among individuals. For example, two different people may take on the role of quality assessor in different departments, and one of these may also be a production manager. To do this, the individuals' responsibilities are listed and a group of them that all lead towards one goal is identified. This goal and its set of responsibilities represents one role. Like the affinity diagram, this activity is concerned with grouping elements together along theme lines. Sometimes individuals use different names for the same role. The artifacts and communications among people need to be consolidated, too, in terms of flows between roles.

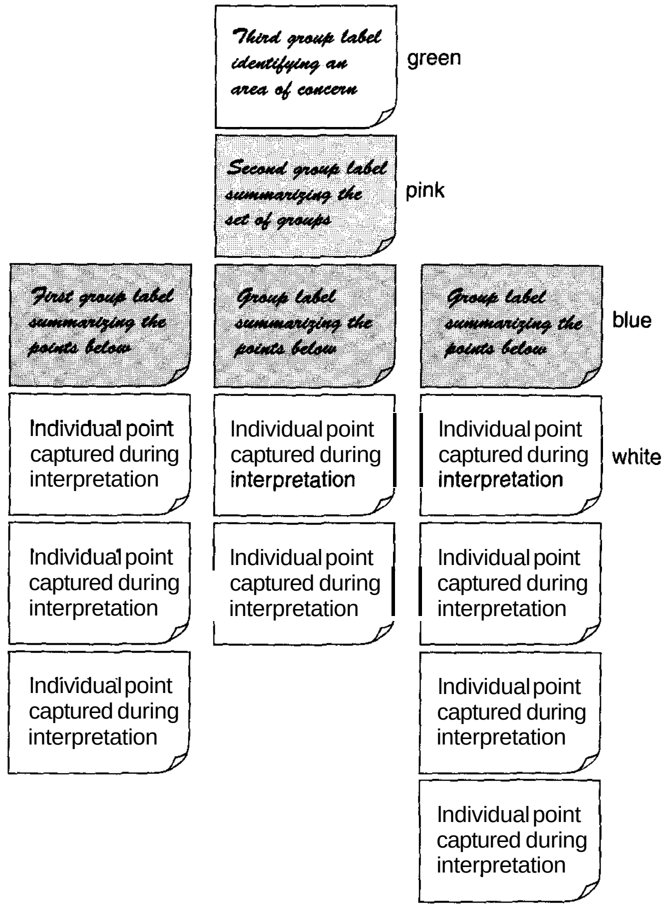


Figure 9.9 The structure of an affinity diagram.

Consolidated sequence models show the structure of a task and common strategies. The consolidated sequence model allows the team to identify what really needs to happen to accomplish the work, and hence what needs to be supported.

Artifact models show how people organize and structure their work, so a consolidated model shows common approaches to this across different people. The sequence models show the steps in the task, while the artifact model shows what is manipulated in order to achieve the task.

Physical space also has commonalities. For example, most companies have an entrance lobby with a receptionist or security guard, then beyond that personal offices and meeting rooms. Within one organization, even if it is distributed across different buildings, there is commonality of physical structure and hence constraints under which the work must be accomplished.

The cultural models help in identifying what matters to people who are doing the work. The cultural model identifies the influencers, so a consolidated model shows the set of common influencers within the organization.

All together, the consolidated models help designers to understand the users' intent, strategy to achieve that intent, structures to support the strategy, concepts to help manage and think about work, and the users' mind set.

The Design Room

An important element of Contextual Design is the design room, where all the work models are kept, pinned to the wall. The room is an environment that contains everything the team knows about the customer and their work. Design discussions held in the room can refer to data collected at the beginning of the project, and this can be used to support design ideas and decisions. This physical space in which the team is surrounded by the data is a key element of Contextual Design.

Contextual Design has been used successfully in a variety of situations from cell phone design (see Chapter 15) to office products (see Box 9.5). Its strength lies in the fact that it provides a clear route from observing users through to interpreting and structuring the data, prototyping and feeding the results into product development. This systematic approach means that, with suitable training, interaction designers can perform the observations and subsequent interpretation themselves, thus avoiding some of the misunderstandings that can happen if observations are conducted by others. Contextual Design is discussed further in the interview with Karen Holtzblatt at the end of this chapter.

9.5 Involving users in design: Participatory Design

Another approach to involving users is ***Participatory Design.*** In contrast to Contextual Design, users are actively involved in development. The intention is that they become an equal partner in the design team, and they design the product in cooperation with the designers.

The idea of participatory design emerged in Scandinavia in the late 1960s and early 1970s. There were two influences on this early work: the desire to be able to communicate information about complex systems, and the labor union movement pushing for workers to have democratic control over changes in their work. In the 1970s, new laws gave workers the right to have a say in how their working environment was changed, and such laws are still in force today. A fuller history of the movement is given in Ehn (1989) and Nygaard (1990).

Several projects at this time attempted to involve users in design and tried to focus on work rather than on simply producing a product. One of the most discussed is the UTOPIA project, a cooperative effort between the Nordic Graphics Workers Union and research institutions in Denmark and Sweden to design computer-based tools for text and image processing.

Involving users in design decisions is not simple, however. Cultural differences can become acute when users and designers are asked to work together to produce a specification for a system. Bødker et al. (1991) recount the following scene from the UTOPIA project:

> ***Late one afternoon, when the designers were almost through with a long presentation of a proposal for the user interface of an integrated text and image processing system, one of the typographers commented on the lack of information about typographical code-***

Sort machine mock-up. The headline reads: "We did not understand the blueprints, so we made our own mock-ups."

Figure 9.10 A newspaper cutting showing a parcel-sorting machine mockup.

> *structure. He didn't think that it was a big error (he was a polite person), but he just wanted to point out that the computer scientists who had prepared the proposal had forgotten to specify how the codes were to be presented on the screen. Would it read “<bf/” or perhaps just “\b” when the text that followed was to be printed in boldface?*

In fact, the system being described by the designers was a WYSIWYG (what you see is what you get) system, and so text that needed to be in bold typeface would appear as bold (although most typographic systems at that time did require such codes). The typographer was unable to link his knowledge and experience with what he was being told. In response to this kind of problem, the project started using mockups (introduced in Chapter 8). Simulating the working situation helped workers to draw on their experience and tacit knowledge, and designers to get a better understanding of the actual work typographers needed to do. An example mockup for a computer-controlled parcel-sorting system, from another project, is shown in Figure 9.10 (Ehn and Kyng, 1991). The headline of this newspaper clipping reads, "We did not understand the blueprints, so we made our own mockups".

Mockups are one way to make effective use of the users' experience and knowledge. Other paper-based prototyping techniques that have been developed for participatory design are PICTIVE (Muller, 1991) and CARD (Tudor, 1993).

9.5.1 PICTIVE

PICTIVE (Plastic Interface for Collaborative Technology Initiatives through Video Exploration) uses low-fidelity office items, such as sticky notes and pens, and a collection of design objects to investigate specific screen and window layouts for a system. The motives for developing the techniques were to:

- empower users to act as full participants in the design process
- improve knowledge acquisition for design

A PICTIVE session may involve one-on-one collaboration or it may involve a small group. To perform a PICTIVE session you need video recording equipment, simple office supplies such as pens, pencils, paper, sticky notes, cards, etc., and some design components prepared by the design team such as dialog boxes, menu bars, and icons. These plastic design components may be generic or they may be specific to the system being developed, based on the development so far. The shared design surface is where the design will be created, jointly between the designers and the users, by manipulating and changing the design components and using the office supplies to create new elements. The video equipment records what happens on the shared design surface. Sample design objects and the layout for a PICTIVE session are shown in Figure **9.11** (Muller, **1991).**

Before a session, each participant is asked to prepare a "homework assignment." Typically, users are asked to generate scenarios of use for the system, illustrating what they would like the system to do for them (along the lines of the scenarios we discussed in Chapter 7). Developers are asked to develop a set of system components that they think may be relevant to the system. These may be generic elements that will be used in many design exercises, they may be specifically for the system under discussion, or a combination of these.

The design session itself is divided roughly into four parts (Muller et al., **1995).** First of all, the stakeholders all introduce themselves, specifically describing their personal **and/or** organizational stake in the project. Then there may be some brief tutorials about the different domains represented at the meeting. The third part of the meeting concentrates on brainstorming the designs, using the design objects and the homework assignments. The design objects are manipulated during the session to produce a synthesis of each participant's view. The scenarios developed by the users may help provide concrete detail about the work flow of the design. The final session is a walkthrough of the design and the decisions discussed. The role of the video recording is mainly that of record-keeper, so that there is a complete and informal record of the design decisions made and how they were made.

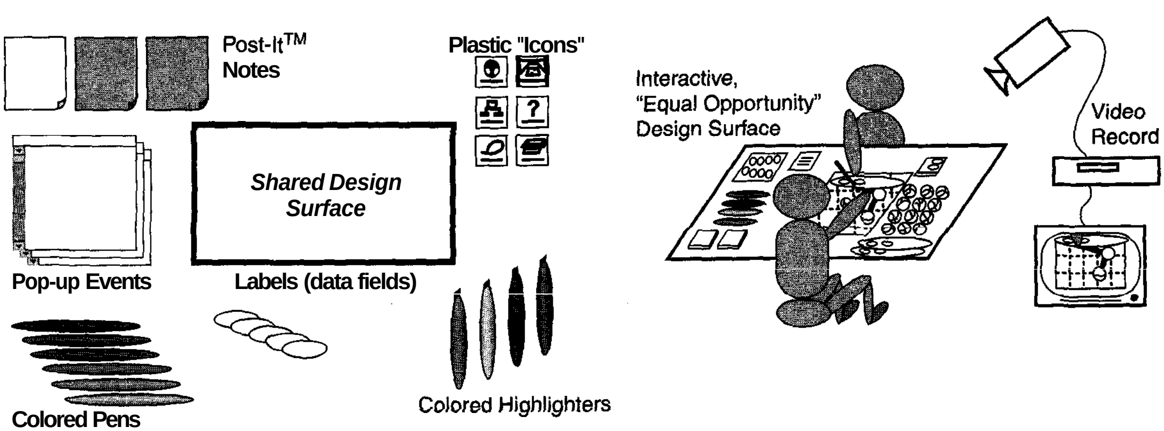


Figure 9.11 PICTIVE design objects **and PICTIVE** setting.

ACTIVITY 9.5 Describe a set of design components you would develop for a PICTIVE session for the shared calendar application discussed in Chapter 8.

Comment From our earlier design activities, we know that having dialog boxes and icons for arranging a meeting would be appropriate. Also, different mechanisms for specifying the people to attend the meeting and for choosing dates, e.g., drop-down lists, free text entry, or planner-style date display. These components could be based on our preliminary designs. We will also need a menu bar and associated menu lists, calendar page display, and function button components. It would also be important to have some blank components that could be completed during the brainstorming session.

9.5.2 CARD

CARD (Collaborative Analysis of Requirements and Design) is similar to PICTIVE, but uses playing cards with pictures of computers and screen dumps on them to explore workflow options (see Figure 9.12 for an example set of cards

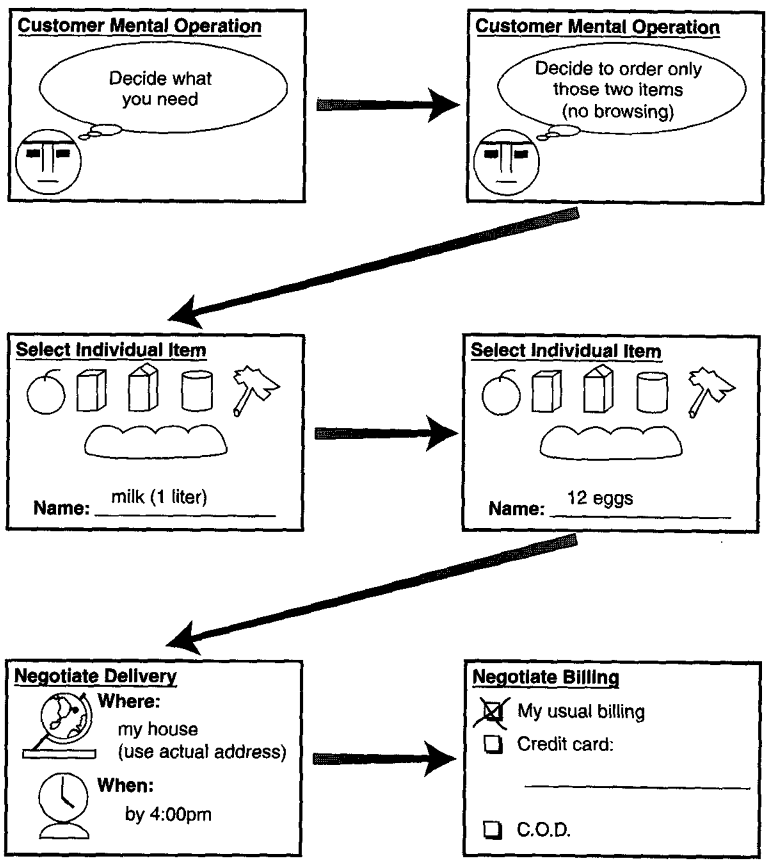


Figure 9.12 Example of CARD.

(Muller et al., 1995)). Whereas PICTIVE concentrates on detailed aspects of the system, CARD takes a more macroscopic view of the task flow. CARD is a form of storyboarding (see Chapter 8).

A CARD session could have the same format as that described for PICTIVE. During the design brainstorming part of the session, the playing cards are manipulated by the participants in order to show the work flow between computer screens or task decision points. The example in Figure 9.12 shows how the task of buying groceries through a computer screen such as via the Internet can be represented by

Table 9.1 A comparison of techniques introduced in this chapter

	Ethnography	Coherence	Contextual Design	Participatory Design[2]
Active user involvement	Low level	Low level	Medium to low level	Equal partners, users can be very influential
Role of designer1 researcher	Uncover findings about work	Collect and present ethnographic data according to the viewpoints and concerns	Steer discussion Interpret findings	Equal partners with users
Length of study	Typically continuous and extensive.	N/A	A series of 2-hour interviews	A series of 2-hour design sessions
Benefits	Yields a good understanding of the work	Overcomes the problem of representing ethnographic data for design	Systematic Is designed to feed into the design process	Users' sense of ownership is increased User contact is beneficial for designers
Drawbacks	Requires expertise Difficulties translating findings into design Requires a long lead-in time	Coverage limited to presenting ethnographic data Limited support currently for progressing to design	Involves many diagrams and notations May be complicated for users to understand the output	Users' thinking can be constrained by what they know If users are involved too much they get bored and it becomes counter-productive
When to use	Most settings where there is sufficient time and expertise	If an ethnographic study for interaction design is to be conducted (by ethnographer or designer)	When a user-centered focus is required Particularly useful for innovative product design	Whenever users are available and willing to become actively involved in design

[2]The main difference between CARD and PICTIVE lies in the level of detail at which design takes place. For the purpose of this comparison, they can be considered under the common title of Participatory Design.

playing cards. Note that the cards can be used to represent users' goals or intentions as well as specific computer screens or task elements. Participants can easily create new cards during the session as deemed appropriate.

CARD can be used to complement PICTIVE as it provides a different granularity of focus. Muller et al. (1995) characterized this as a bifocal view, CARD giving a macroscopic view, and PICTIVE the microscopic.

At the beginning of this chapter, we explained that there are different levels of user involvement, from newsletters and workshops through to full-time membership of the design team. Each project will need to decide on the level of user involvement required. To support this involvement, a project may also choose to use one or a combination of the techniques introduced in Sections 9.4 and 9.5. For example, Contextual Design could be used even if one of the users is a member of the design team; an ethnographic study might be running alongside a series of user workshops. These techniques expand the level of user involvement. However, each approach has advantages and disadvantages, and Table 9.1 provides a brief comparison between the main techniques introduced in this chapter.

Assignment

This assignment asks you to apply some elements of Coherence and Contextual Design to your own work or home circumstances.

(a) Using the questions for elaborating the viewpoints and concerns in Coherence, study the environment of your workplace, university library or somewhere similar that you know. Begin by deciding which concerns are relevant to each viewpoint, e.g., ask, "Are there paper artifacts used in the workplace?" or "Is local knowledge used?" Then answer the questions of elaboration for the three viewpoints and the four concerns.

Study your answers to the questions and see if you can identify priorities or constraints within the organization that you were not aware of before.

(b) Again using your workplace or similar location, attempt to draw the five Contextual Design work models introduced in Section **9.4.3.**

First of all, identify a key player in the workplace. This may be one of the librarians, a clerk or secretary, or a manager. If possible, run a contextual inquiry interview by sitting with her while working and asking her to tell you about one major aspect of work. If this is not possible, then identify one of the main tasks that is visible to you, such as the librarian issuing books, and sit and watch how the task is performed.

Draw the models from the information you have collected. If you find that you need more data, go back and collect more. Once you feel that the models are complete, take them back to the person you interviewed (if possible) and ask for comments.

Summary

This chapter has elaborated on some issues surrounding the involvement of users in the design process. We have also introduced the method of ethnography as a useful source of information for a user-centered design process. One of the main disadvantages to using ethnography is finding a way to represent the output of the study so that it can be fed into the design process. We have described two approaches to design (Coherence and Contextual Design) that were derived from ethnography and other approaches, to address this problem.

Users may be involved passively or they may be more actively involved in making design decisions. Participatory design is an approach in which users are co-designers. We have described two techniques (PICTIVE and CARD) that have helped users' input to be more effective.

Key Points

- Involving users in the design process helps with expectation management and feelings of ownership, but how and when to involve users is a matter of dispute.
- Putting a user-centered approach into practice requires much information about the users to be gathered and interpreted.
- Ethnography is a good method for studying users in their natural surroundings.
- Representing the information gleaned from an ethnographic study so that it can be used in design has been problematic.
- The goals of ethnography are to study the details, while the goals of system design are to produce abstractions; hence they are not immediately compatible.
- Coherence is a method that provides focus questions to help guide the ethnographer towards issues that have proved to be important in systems development.
- Contextual Design is a method that provides models and techniques for gathering contextual data and representing it in a form suitable for practical design.
- PICTIVE and CARD are both participatory design techniques that empower users to take an active part in design decisions.

Further reading

GREENBAUM, JOAN, AND KYNG, MORTEN (eds.) (1991) ***Design at Work: Co-operative Design of Computer Systems.*** Hillsdale, NJ: Lawrence Erlbaum. This book is a good collection of papers about the co-design of software systems: both why it is worthwhile and experience of how to do it.

BEYER, HUGH AND HOLTZBLATT, KAREN (1998) *Contextual Design: Defining Customer-Centered Systems.* San Francisco: Morgan Kaufmann. This book will tell you more about contextual design and the rationale behind the steps and the models.

CUSUMANO, M.A., AND SELBY, R. W. (1995) ***Microsoft Secrets.*** London: Harper-Collins Business. This is a fascinating book based on a two-and-a-half-year study of Microsoft and how they build software. The book details findings about strategies to manage an innovative organization competing in a rapidly changing world, to develop and ship products that appeal to mass markets, and to continually build on and improve market position.

WIXON, DENNIS, AND RAMEY, JUDITH (eds.) (1996) ***Field Methods Casebook for Software Design.*** New York: John Wiley & Sons, Inc. This book is a collection of papers about practical use of field research methods in software design, some of which are directly mentioned in the present chapter. The three main approaches that these papers cover are ethnography, participative design, and contextual design. There are 14 chapters describing case studies and three chapters giving an overview of the main methods. For anyone interested in the practical use of these methods in software development, it's a fascinating read!

INTERVIEW Karen H att

Karen Holtzblatt is th origi- n of Contextual quiry, a process r gathering data on d use, which was the precursor to Contextual Design, a complete method for the design of systems. Together with Hugh Beyer, the codeveloper of Contextual Design, K Holtzblatt is co-founder of InContext Enterprises, which specializes in process and product design consulting.

HS: What is Contextual Design?

KH: If you're going to build something that people want, there are basically three large steps that you have to go through. The first question that you ask as a company is, "What in the world matters to the customer such that if we make something, they're likely to buy it?" So the question is "What matters?" Now once you identify what the issues are, every corporation will have the corporate response, or "vision." Then you have to work out the details and structure it into a product. In any design process, whether it's formalized or not, every company must do those things. They have to find out what matters, they have to vision their corporate response, and then they have to structure it into a system.

Contextual Design gives you team and individual activities that bring you through those processes in an orderly fashion so as to bring the cross-functions of an organization together. So you could say that Contextual Design is a set of techniques to be used in a customer-centered design process with design teams. It is also a set of practices that help people engage in creative and productive design thinking with customer data and it helps them co-operate and design together.

HS: What are the steps of Contextual Design?

KH: In the "what matters" piece, we go out into the field, we talk with people about their work as they do it: that's Contextual Inquiry and that's a one-on-one, two to two-and-a-half-hour field interview. Then we interpret that data with a cross-functional team, and we model the work with five work models: communication and coordination, the cultural environment, tl physical environment, ., and artifact. We also capture individual points on p notes. After the t p t i every person we interviewed has a set of models and a t of post it Our t t is to s all that data because you don't want to be designing from one n from s lf, or from any one interview; we need to at the structure of tl practice it The lid ti n p means that v p with an affinity diagram and five consolidated models showing the issues *across* the market.

At that point, we have modeled the work practice as it is and we have now six communication devices that the team can dialog with. Each one of them poses a point of view on which to have the conversation "what matters?"

Now the team moves into that second piece, which is "what should our corporate response be?" We have a visioning process that is a very large group story-telling about reinventing work practice given technological possibility and the core competency of the organization. And after that, we develop storyboards driven by the consolidated data and the vision. At this point we have not done a systems design; we want to design the work practice first, seeing the technology as it will appear within the work.

To structure the system we start by rolling the storyboards into a user environment design—the structure of the system itself, independent of the user interface and the object model. The user environment design operates like a software floor plan that structures the movement inside the product. This is used to drive the user interface design, which is mocked up in paper and tested and iterated with the user. When it has stabilized, the User Environment design, the storyboards, and the user interface drive development of the object model.

This is the whole process of Contextual Design, a full front-end design process. Because it is done with a cross-functional team, everyone in the organization knows what they're doing at each point: they know how to select the data, they know how to work in groups to get all these different steps done. So not only do you end up with a set of design thinking techniques that help you to design, you have an organizational process that helps the organization actually do it.

HS: How did the idea of contextual design emerge?

KH: Contextual Design started with the invention of Contextual Inquiry in a post-doctoral internship with John Whiteside at Digital. At the time, usability testing and usability issues had been around maybe eight years or so and he was asking the question, "Usability identifies about 10 to 20% of the fixes at the tail end of the process to make the frosting on the cake look a little better to the user. What would it take to really infuse usability?" Contextual Inquiry was my answer to that question. After that, I took a job with Lou Cohen's Quality group at DEC, where I picked up the affinity diagram idea. Also at that time, Pelle Ehn and Kim Madsen were talking about Morten Kyng's ideas on paper mockups and I added paper prototyping with post-its to check out the design. Hugh and I hooked up 13 years ago. He's a software and object-oriented developer. We started working with teams and we noticed that they didn't know how to go from the data to the design and they didn't know how to structure the system to think about it. So then we invented more of the work models and the user environment design.

So the Contextual Design method came from looking at the practice; we evolved every single step of this process based on what people needed. The whole process was worked out with real people doing real design in real companies. So, where did it come from? It came from dialog with the problem.

HS: What are the main problems that organizations face when putting Contextual Design into practice?

KH: The question is, "What does organizational change look like?" because that's what we're talking about. The problem is that people want to change and they **don't** want to change. What we communicate to people is that organizational change is piecemeal. In order to own a process you have to say what's wrong with it, you have to change it a little **bit,** you have to say how whoever invented the process is wrong and how the people in the organization want to fix it, you have to make it fit with your organizational culture and issues. Most people will adopt the field-data gathering first and that's all they'll do and they'll tell me that they don't have time for anything else and they don't need anything else, and that's fine. And then they'll wake up one day and they'll say, "We have all this qualitative stuff and nobody's using it ... maybe we should have a debriefing session." So then they have debriefing sessions. Then they wake up later on and they say, "We don't have any way of structuring this information ... models are a good idea." And basically they reconstruct the whole process as they hit the next problem.

Now it's not quite that clean, but my point is that organizational adoption is about people making it their own and taking on the parts, changing them, doing what they can. You have to get somebody to do something and then once they do something it snowballs.

What's nice about the Contextual Design way of doing everything on paper is that it creates a design room, the design room creates a talk event, and the talk event pulls everyone in because they want to know what you're doing. Then if they like the data, **they feel left *out,*** and because they feel left out they want to do a project and they want to have a room for themselves as well.

The biggest complaint about Contextual Design is that it takes too long. Some of that is about time, some of it is about thought. You have people who are used to coding and now have to think about field data. They're not used to that.

HS: What's the future direction of Contextual Design?

KH: Every process can always be tweaked. I think the primary parts of Contextual Design are there. There are interesting directions in which it can go, but there's only so much we can get our audience to buy.

I think that for us there are two key things that we're doing. One is we're starting to talk about design and what design is, so we can talk about the role of design in design thinking. And we are still helping train everyone who wants to learn. But the other thing we're finding is that sometimes the best way to support the client is to do the design work for them. So we have the design wing of the business where we put together the contextual design teams.

We're working with distributed teams, we're working with creativity and invention, we're working with how it impacts with business processes and marketing, we're working with the balance of all those things. But it's only going to be in the context of a team that's actually very advanced in the standard process that new process inventions will occur. Out of that will come lessons that can then be put back into the standard contextual design. For most organizations looking to adopt a customer-centered design process, the standard contextual design is enough for now, they have to get started. And because Contextual Design is a scaffolding, they can plug other processes into it. They take their usability testing and they can plug it here, if they have their special creativity thing they can plug it here; if they have a focus group they can plug it here. But most people haven't got a backbone for design, and Contextual Design is a good backbone to start with.

Chapter I 0

Introducing evaluation

10.1 Introduction

Recently I met two web designers who, proud of their newest site, looked at me in astonishment when I asked if they had tested it with users. "No," they said "but we know it's OK." So, I probed further and discovered that they had asked the "web whiz-kids" in their company to look at it. These guys, I was told, knew all the tricks of web design.

The web's presence has heightened awareness about usability, but unfortunately this reaction is all too common. Designers assume that if they and their colleagues can use the software and find it attractive, others will too. Furthermore, they prefer to avoid doing evaluation because it adds development time and costs money. So why is evaluation important? Because without evaluation, designers cannot be sure that their software is usable and is what users want. But what do we mean by evaluation? There are many definitions and many different evaluation techniques, some of which involve users directly, while others call indirectly on an understanding of users' needs and psychology. In this book we define evaluation as the process of systematically collecting data that informs us about what it is like for a particular user or group of users to use a product for a particular task in a certain type of environment.

As you read in Chapter 9, the basic premise of user-centered design is that users' needs are taken into account throughout design and development. This is achieved by evaluating the design at various stages as it develops and by amending it to suit users' needs (Gould and Lewis, 1985). The design, therefore, progresses in iterative cycles of design-evaluate redesign. Being an effective interaction designer requires knowing how to evaluate different kinds of systems at different stages of development. Furthermore, developing systems in this way usually turns out to be less expensive than fixing problems that are discovered after the systems have been shipped to customers (Karat, 1993). Studies also suggest that the business case for using systems with good usability is compelling (Dumas and Redish, 1999; Mayhew, 1999): thousands of dollars can be saved.

Many techniques are available for supporting design and evaluation. Chapter 9 discussed techniques for involving users in design and part of this involvement comes through evaluation. In this and the next four chapters you will learn how different techniques are used at different stages of design to examine different aspects of the design. You will also meet some of the same techniques that are used for gathering user requirements, but this time used to collect data to evaluate the design. Another aim is to show you how to do evaluation.

This chapter begins by discussing ***what*** evaluation is, ***why*** evaluation is important, and ***when*** to use different evaluation techniques and approaches. Then a case study is presented about the evaluation techniques used by Microsoft researchers and the Fred Hutchinson Cancer Research Center in developing HutchWorld (Cheng et al., 2000), a virtual world to support cancer patients, their families, and friends. This case study is chosen because it illustrates how a range of techniques is used during the development of a new product. It introduces some of the practical problems that evaluators encounter and shows how iterative product development is informed by a series of evaluation studies. The HutchWorld study also lays the foundation for the evaluation framework that is discussed in Chapter 11.

The main aims of this chapter are to:

- Explain the key concepts and terms used to discuss evaluation.
- Discuss and critique the HutchWorld case study.
- Examine how different techniques are used at different stages in the development of HutchWorld.
- Show how developers cope with real-world constraints in the development of HutchWorld.

10.2 What, why, and when to evaluate

Users want systems that are easy to learn and to use as well as effective, efficient, safe, and satisfying. Being entertaining, attractive, and challenging, etc. is also essential for some products. ***So,*** knowing what to evaluate, why it is important, and when to evaluate are key skills for interaction designers.

10.2.1 What to evaluate

There is a huge variety of interactive products with a vast array of features that need to be evaluated. Some features, such as the sequence of links to be followed to find an item on a website, are often best evaluated in a laboratory, since such a setting allows the evaluators to control what they want to investigate. Other aspects, such as whether a collaborative toy is robust and whether children enjoy interacting with it, are better evaluated in natural settings, so that evaluators can see what children do when left to their own devices.

You may remember from Chapters 2, *6* and 9 that John Gould and his colleagues (Gould et al., 1990; Gould and Lewis, 1985) recommended three similar principles for developing the 1984 Olympic Message System:

- focus on users and their tasks
- observe, measure, and analyze their performance with the system
- design iteratively

Box 10.1 takes up the evaluation part of the 1984 Olympic Messaging System story and lists the many evaluation techniques used to examine different parts of the OMS during its development. Each technique supported Gould et al.'s three principles.

Since the OMS study, a number of new evaluation techniques have been developed. There has also been a growing trend towards observing how people interact with the system in their work, home, and other settings, the goal being to obtain a better understanding of how the product is (or will be) used in its intended setting. **For example, at work people are frequently being interrupted by phone calls, oth**ers knocking at their door, email arriving, and so on—to the extent that many tasks are interrupt-driven. Only rarely does someone carry a task out from beginning to end without stopping to do something else. Hence the way people carry out an activity (e.g., preparing a report) in the real world is very different from how it may be observed in a laboratory. Furthermore, this observation has implications for the way products should be designed.

10.2.2 Why you need to evaluate

Just as designers shouldn't assume that everyone is like them, they also shouldn't presume that following design guidelines guarantees good usability, Evaluation is needed to check that users can use the product and like it. Furthermore, nowadays users look for much more than just a usable system, as the Nielsen Norman Group, a usability consultancy company, point out (www.nngroup.com):

> ***"User experience" encompasses all aspects of the end-user's interaction . . . the first requirement for an exemplary user experience is to meet the exact needs of the customer, without fuss or bother. Next comes simplicity and elegance that produce products that are a joy to own, a joy to use."***

Bruce Tognazzini, another successful usability consultant, comments (www.asktog.com) that:

> ***"Iterative design, with its repeating cycle of design and testing, is the only validated methodology in existence that will consistently produce successful results. If you don't have user-testing as an integral part of your design process you are going to throw buckets of money down the drain."***

BOX 10.1 The Story of the 1984 Olympic Messaging System

The 1984 Olympic Message System (OMS), a voice mail system, was developed by IBM so that Olympic Games contestants and their families and friends could send and receive messages (Gould et al., 1990). They could hear the message and the actual voice of the sender exactly as it was spoken. This system could be used from almost any push-button phone system around the world: this may not sound amazing when compared with today's technology, but in 1983 it was highly innovative.

Non-Olympians called their own country's National Olympic Committee using either push-button or dial telephones and spoke in their own language. They were helped to connect to OMS so that they could leave their messages. The voice message was immediately transferred by a central telephone operator to the message boxes of the Olympian for whom it was intended. The OMS worked in 12 languages. The kiosks looked like the one in Figure 10.1 and the dialog is shown in Figure 10.2

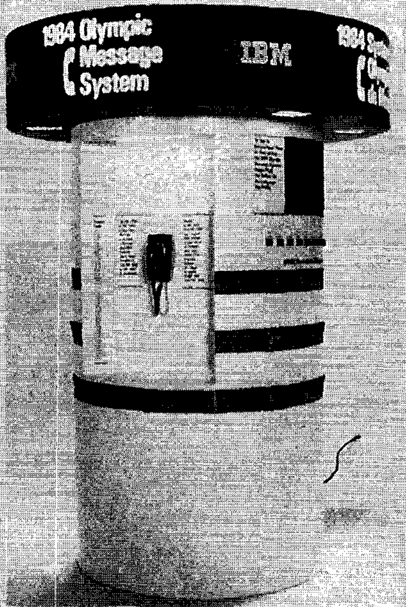


Figure 10.1 The Olympic message system kiosk.

During development, the evaluation activities included:

- Use of printed scenarios of the screens to get feedback from the Olympic committee and the Olympians themselves.
- Iteratively testing the user guides for the OMS with the Olympians, their families, and friends.
- Developing early simulations of a telephone keypad with a person speaking the commands back. These simulations really tested how much a user needed to know about the system, what feedback was needed, and any incorrect assumptions about user behavior made by the designers.
- Developing early demonstrations to test the reactions of people outside the US who did not know much about computers.
- An Olympian joining the design team to discuss ideas and provide feedback.
- Interviews with Olympians to make sure that the system being developed was what the users wanted.
- Overseas tests of the interface with friends and family.
- Free coffee and donut tests: 65 people were enticed to test the system in return for these treats.
- More traditional usability tests (discussed in Chapter 14) of the prototype involving about 100 participants.
- A 'try-to-destroy-it' test in which 24 computer science students were challenged to bring down the system. One of these tests involved all the students calling into the OMS at the same time. The students enjoyed the challenge and didn't need any other motivation!
- A pre-Olympic field test of the interface at an international event with competitors from 65 countries. The outcome of this test was surprising because, despite all the other testing, 57 different usability problems were recorded by the end of the five-day test period. The lesson for the design team was that the results of

field tests could be surprising. In this case they discovered that strong cultural differences affected how users from different countries used the OMS. Testers from Oman, Colombia, Pakistan, Japan, and Korea were unable to use the system. Gould and his colleagues comment that "watching this helplessness and hopelessness had a far greater impact than reading about it. It was embarrassing . . ." (Gould et al., 1990, p. 274).

- Two other tests examined the reliability of the system with heavy traffic generated by 2800 and 1000 people respectively.

This extensive evaluation was needed because the Olympics was such a high-profile event and IBM's reputation was at stake. Less intensive evaluation is more normal. However, the take-away message from this study is that the more evaluation with users, the better the final product.

Caller:	(Dials 213-888-8888.)
Operator:	Irish National Olympic Committee. Can I help you?
Caller:	I want to leave a message for my son, Michael.
Operator:	Is he from Ireland?
Caller:	Yes.
Operator:	How do you spell his name?
Caller:	*K-E-L-L-Y*.
Operator:	Thank you. Please hold for about 30 seconds while I connect you to the Olympic Message System.
Operator:	Are you ready?
Caller:	Yes.
OMS:	When you have completed your message, hang up and it will be automatically sent to Michael Kelly. Begin talking when you are ready.
Caller:	'Michael, your Mother and I will be hoping you win. Good luck.' (Caller hangs up.)

Figure 10.2 Parent leaving a voice message for an Olympian.

Tognazzini points out that there are five good reasons for investing in user testing:

1. Problems are fixed before the product is shipped, not after.
2. The team can concentrate on real problems, not imaginary ones.
3. Engineers code instead of debating.
4. Time to market is sharply reduced.
5. Finally, upon first release, your sales department has a rock-solid design it can sell without having to pepper their pitches with how it will all actually work in release 1.1 or 2.0.

Now that there is a diversity of interactive products, it is not surprising that the range of features to be evaluated is very broad. For example, developers of a new web browser may want to know if users find items faster with their product. Government authorities may ask if a computerized system for controlling traffic lights results in fewer accidents. Makers of a toy may ask if six-year-olds can manipulate the controls and whether they are engaged by its furry case and pixie face. A company that develops the casing for cell phones may ask if the shape, size, and color of the case is appealing to teenagers. A new dotcom company may want to assess market reaction to its new home page design.

This diversity of interactive products, coupled with new user expectations, poses interesting challenges for evaluators, who, armed with many well tried and tested techniques, must now adapt them and develop new ones. As well as usability, user experience goals can be extremely important for a product's success, as discussed in Chapter 1.

ACTIVITY 10.1 Think of examples of the following systems and write down the usability and user experience features that are important for the success of each:

(a) a word processor

(b) a cell phone

(c) a website that sells clothes

(d) an online patient support community

Comment

(a) It must be as easy as possible for the intended users to learn and to use and it must be satisfying. Note, that wrapped into this are characteristics such as consistency, reliability, predictability, etc., that are necessary for ease of use.

(b) A cell phone must also have all the above characteristics; in addition, the physical design (e.g., color, shape, size, position of keys, etc.) must be usable and attractive (e.g., pleasing feel, shape, and color).

(c) A website that sells clothes needs to have the basic usability features too. In particular, navigation through the system needs to be straightforward and well supported. You may have noticed, for example, that some sites always show a site map to indicate where you are. This is an important part of being easy to use. So at a deeper level you can see that the meaning of "easy to use and to learn" is different for different systems. In addition, the website must be attractive, with good graphics of the clothes—who would want to buy clothes they can't see or that look unattractive? Trust is also a big issue in online shopping, so a well-designed procedure for taking customer credit card details is essential: it must not only be clear but must take into account the need to provide feedback that engenders trust.

(d) An online patient support group must support the exchange of factual and emotional information. So as well as the standard usability features, it needs to enable patients to express emotions either publicly or privately, using emoticons. Some 3D environments enable users to show themselves on the screen as avatars that can jump, wave, look happy or sad, move close to another person, or move away. Designers have to identify the types of social interactions that users want to express (i.e., sociability) and then find ways to support them (Preece, 2000).

From this selection of examples, you can see that success of some interactive products depends on much more than just usability. Aesthetic, emotional, engaging, and motivating qualities are important too.

Usability testing involves measuring the performance of typical users on typical tasks. In addition, satisfaction can be evaluated through questionnaires and interviews. As mentioned in Chapter 1, there has been a growing trend towards developing ways of evaluating the more subjective user-experience goals, like emotionally satisfying, motivating, fun to use, etc.

10.2.3 When to evaluate

The product being developed may be a brand-new product or an upgrade of an existing product. If the product is new, then considerable time is usually invested in market research. Designers often support this process by developing mockups of the potential product that are used to elicit reactions from potential users. As well as helping to assess market need, this activity contributes to understanding users' needs and early requirements. As we said in Chapter 8, sketches, screen mockups, and other low-fidelity prototyping techniques are used to represent design ideas. Many of these same techniques are used to elicit users' opinions in evaluation (e.g., questionnaires and interviews), but the purpose and focus of evaluation is different. The goal of evaluation is to assess how well a design fulfills users' needs and whether users like it.

In the case of an upgrade, there is limited scope for change and attention is focused on improving the overall product. This type of design is well suited to usability engineering in which evaluations compare user performance and attitudes with those for previous versions. Some products, such as office systems, go through many versions, and successful products may reach double-digit version numbers. In contrast, new products do not have previous versions and there may be nothing comparable on the market, so more radical changes are possible if evaluation results indicate a problem.

Evaluations done during design to check that the product continues to meet users' needs are know as *formative evaluations.* Evaluations that are done to assess the success of a finished product, such as those to satisfy a sponsoring agency or to check that a standard is being upheld, are know as *summative evaluation.* Agencies such as National Institute of Standards and Technology (NIST) in the USA, the International Standards Organization (ISO) and the British Standards Institute (BSI) set standards by which products produced by others are evaluated.

ACTIVITY 10.2 Re-read the discussion of the 1984 Olympic Messaging System (OMS) in Box 10.1 and briefly describe some of the things that were evaluated, why it was necessary to do the evaluations, and when the evaluations were done.

Comment Because the Olympic Games is such a high-profile event and IBM's reputation was at stake, the OMS was intensively evaluated throughout its development. We're told that early evaluations included obtaining feedback from Olympic officials with scenarios that used printed screens and tests of the user guides with Olympians, their friends, and family. Early evaluations of simulations were done to test the usability of the human-computer dialog. These were done first in the US and then with people outside of the US. Later on, more formal tests investigated how well 100 participants could interact with the system. The system's robustness was also tested when used by many users simultaneously. Finally, tests were done with users from minority cultural groups to check that they could understand how to use the OMS.

So how do designers decide ***which*** evaluation techniques to use, ***when*** to use them, and ***how*** to use the findings? To address these concerns, we provide a case study showing how a range of evaluation techniques were used during the development of a new system. Based on this, we then discuss issues surrounding the "which, when, and how" questions relating to evaluation.

10.3 HutchWorld case study

HutchWorld is a distributed virtual community developed through collaboration between Microsoft's Virtual Worlds Research Group and librarians and clinicians at the Fred Hutchinson Cancer Research Center in Seattle, Washington. The system enables cancer patients, their caregivers, family, and friends to chat with one another, tell their stories, discuss their experiences and coping strategies, and gain emotional and practical support from one another (Cheng et. al., 2000). The design team decided to focus on this particular population because caregivers and cancer patients are socially isolated: cancer patients must often avoid physical contact with others because their treatments suppress their immune systems. Similarly, their caregivers have to be careful not to transmit infections to patients.

The big question for the team was how to make HutchWorld a useful, engaging, easy-to-use and emotionally satisfying environment for its users. It also had to provide privacy when needed and foster trust among participants. **A** common approach to evaluation in a large project like Hutchworld is to begin by carrying out a number of informal studies. Typically, this involves asking a small number of users to comment on early prototypes. These findings are then fed back into the iterative development of the prototypes. This process is then followed by more formal usability testing and field study techniques. Both aspects are illustrated in this case study. In addition, you will read about how the development team managed their work while dealing with the constraints of working with sick people in a hospital environment.

10.3.1 How the design team got started: early design ideas

Before developing this product, the team needed to learn about the patient experience at the Fred Hutchinson Center. For instance, what is the typical treatment process, what resources are available to the patient community, and what are the needs of the different user groups within this community? They had to be particularly careful about doing this because many patients were very sick. Cancer patients also typically go through bouts of low emotional and physical energy. Caregivers also may have difficult emotional times, including depression, exhaustion, and stress. Furthermore, users vary along other dimensions, such as education and experience with computers, age and gender and they come from different cultural backgrounds with different expectations.

It was clear from the onset that developing a virtual community for this population would be challenging, and there were many questions that needed to be answered. For example, what kind of world should it be and what should it provide? What exactly do users want to do there? How will people interact? What should it look like? To get answers, the team interviewed potential users from all the stakeholder groups—patients, caregivers, family, friends, clinicians, and social support staff—and observed their daily activity in the clinic and hospital. They also read the latest research literature, talked to experts and former patients, toured the Fred Hutchinson (Hutch) research facilities, read the Hutch web pages, and visited the Hutch school for pediatric patients and juvenile patient family members. No stone was left unturned.

The development team decided that HutchWorld should be available for patients any time of day or night, regardless of their geographical location. The team knew from reading the research literature that participants in virtual communities are often more open and uninhibited about themselves and will talk about problems and feelings in a way that would be difficultin face-to-face situations. On the downside, the team also knew that the potential for misunderstanding is higher in virtual communities when there is inadequate non-verbal feedback (e.g., facial expressions and other body language, tone of voice, etc.). On balance, however, research indicates that social support helps cancer patients both in the psychological adjustments needed to cope and in their physical wellbeing. For example, research showed that women with breast cancer who received group therapy lived on average twice as long as those who did not (Spiegel, et **al.,** 1989). The team's motivation to create HutchWorld was therefore high. The combination of information from research literature and from observations and interviews with users convinced them that this was a worthwhile project. But what did they do then?

The team's informal visits to the Fred Hutchinson Center led to the development of an early prototype. They followed a user-centered development methodology. Having got a good feel for the users' needs, the team brainstormed different ideas for an organizing theme to shape the conceptual design—a conceptual model possibly based on a metaphor. After much discussion, they decided to make the design resemble the outpatient clinic lobby of the Fred Hutchinson Cancer Research Center. By using this real-world metaphor, they hoped that the users would easily infer what functionality was available in HutchWorld from their knowledge of the real clinic. The next step was to decide upon the kind of communication environment to use. Should it be synchronous or asynchronous? Which would support social and affective communications best? **A** synchronous chat environment was selected because the team thought that this would be more realistic and personal than an asynchronous environment. They also decided to include 3D photographic avatars so that users could enjoy having an identifiable online presence and could easily recognize each other.

Figure 10.3 shows the preliminary stages of this design with examples of the avatars. You can also see the outpatient clinic lobby, the auditorium, the virtual garden, and the school. Outside the world, at the top right-hand side of the screen, is a list of commands in a palette and a list of participants. On the right-hand side at the bottom is a picture of participants' avatars, and underneath the window is the textual chat window. Participants can move their avatars and make them gesture to tour the virtual environment. They can also click on objects such as pictures and interact with them.

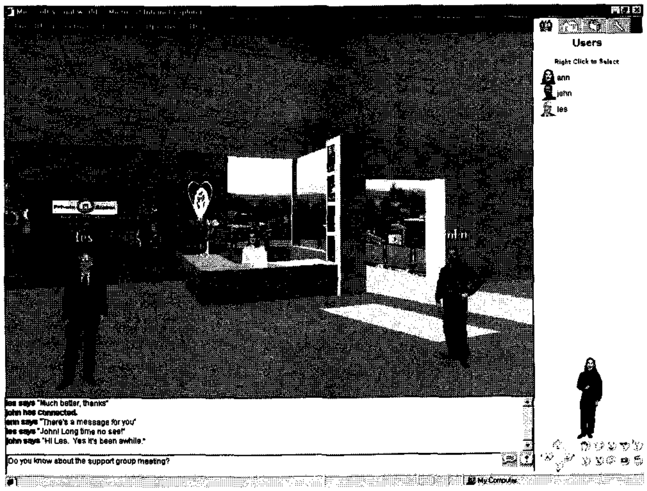


Figure 10.3 Preliminary design showing a view of the entrance into Hutch-World.

The prototype was reviewed with users throughout early development and was later tested more rigorously in the real environment of the Hutch Center using a variety of techniques. A Microsoft product called V-Chat was used to develop a second interactive prototype with the subset of the features in the preliminary design shown in Figure 10.3; however, only the lobby was fully developed, not the auditorium or school, as you can see in the new prototype in Figure 10.4.

Before testing could begin, the team had to solve some logistical issues. There were two key questions. Who would provide training for the testers and help for the patients? And how many systems were needed for testing and where should they be placed? As in many high-tech companies, the *Microsoft* team was used to short, market-driven production schedules, but this time they were in for a shock. Organizing the testing took much longer than they anticipated, but they soon

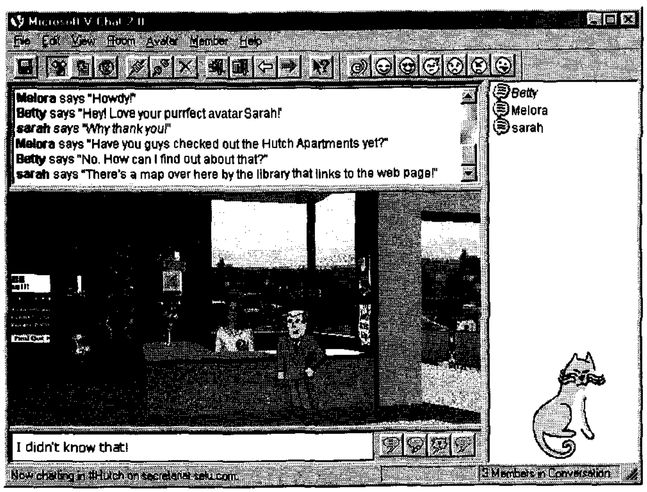


Figure 10.4 The Hutch V-Chat prototype.

learned to set realistic expectations that were in synch with hospital activity and the unexpected delays that occur when working with people who are unwell.

10.3.2 How was the testing done?

The team ran two main sets of user tests. The first set of tests was informally run onsite at the Fred Hutchinson Center in the hospital setting. After observing the system in use on computers located in the hospital setting, the team redesigned the software and then ran formal usability tests in the usability labs at Microsoft.

Test 1: Early observations onsite

In the informal test at the hospital, six computers were set up and maintained by Hutch staff members. A simple, scaled-back prototype of HutchWorld was built using the existing product, Microsoft V-Chat and was installed on the computers, which patients and their families from various hospital locations used. Over the course of several months, the team trained Hutch volunteers and hosted events in the V-Chat prototype. The team observed the usage of the space during unscheduled times, and they also observed the general usage of the prototype.

Test 1: What was learned?

This V-Chat test brought up major usability issues. First, the user community was relatively small, and there were never enough participants in the chat room for successful communication—a concept known as ***critical mass.*** In addition, many of the patients were not interested in or simultaneously available for chatting. Instead, they preferred asynchronous communication, which does not require an immediate response. Patients and their families used the computers for email, journals, discussion lists, and the bulletin boards largely because they could be used at any time and did not require others to be present at the same time. The team learned that a strong asynchronous base was essential for communication.

The team also observed that the users used the computers to play games and to search the web for cancer sites approved by Hutch clinicians. This information was not included in the virtual environment, and so users were forced to use many different applications. A more "unified" place to find all of the Hutch content was desired that let users rapidly swap among a variety of communication, information, and entertainment tasks.

Test 1: The redesign

Based on this trial, the team redesigned the software to support more asynchronous communication and to include a variety of communication, information, and entertainment areas. They did this by making HutchWorld function as a portal that provides access to information-retrieval tools, communication tools, games, and other types of entertainment. Other features were incorporated too, including email, a bulletin board, a text-chat, a web page creation tool, and a way of checking to see if anyone is around to chat with in the 3D world. The new portal version is show in Figure 10.5.

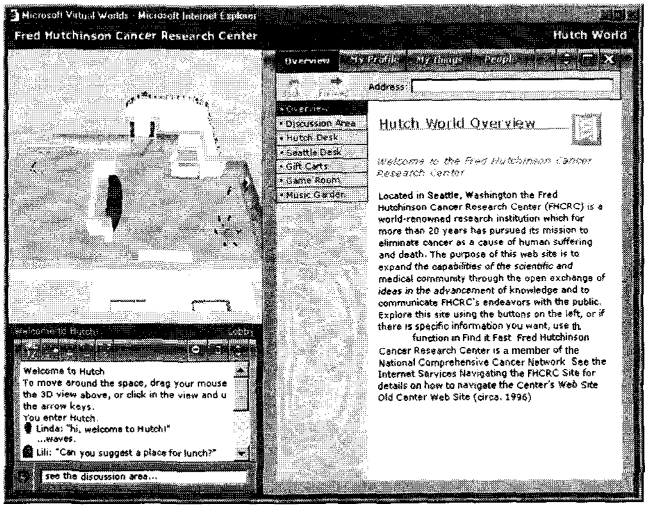


Figure 10.5 HutchWorld portal version.

Test 2: Usability tests

After redesigning the software, the team then ran usability tests in the Microsoft usability labs. Seven participants (four male and three female) were tested. Four of these participants had used chat rooms before and three were regular users. All had browsed the web and some used other communications software. The participants were told that they would use a program called HutchWorld that was designed to provide support for patients and their families. They were then given five minutes to explore HutchWorld. They worked independently and while they explored they provided a running commentary on what they were looking at, what they were thinking, and what they found confusing. This commentary was recorded on video and so were the screens that they visited, so that the Microsoft evaluator, who watched through a one-way mirror, had a record of what happened for later analysis. Participants and the evaluator interacted via a microphone and speakers. When the five-minute exploration period ended, the participants were asked to complete a series of *structured tasks* that were designed to test particular features of the HutchWorld interface.

These tasks focused on how participants:

- dealt with their virtual identity; that is, how they represented themselves and were perceived by others
- communicated with others
- got the information they wanted
- found entertainment

Figure 10.6 shows some of the structured tasks. Notice that the instructions are short, clearly written, and specific.

Welcome to the HutchWorld Usability Study

For this study we are interested in gaining a better understanding of the problems people have when using the program HutchWorld. HutchWorld is an all-purpose program created to offer information and social support to patients and their families at the Fred Hutchinson Cancer Research Center.

The following pages have tasks for you to complete that will help us achieve that better understanding.

While you are completing these tasks, it is important for us know what is going on inside your mind. Therefore, as you complete each task please tell us what you are looking at, what you are thinking about, what is confusing to you, and so forth.

Task #1: Explore HutchWorld

Your first task is to spend five minutes exploring HutchWorld.

A. First, open HutchWorld.

B. Now, explore!

Remember, tell us what you are looking at and what you are thinking about as you are exploring HutchWorld

Task #2: All about Your Identity in HutchWorld

A. Point to the 3 dimensional (3D) view of HutchWorld.

B. Point at yourself in the 3D view of HutchWorld.

C. Get a map view in the 3D view of HutchWorld.

D. Walk around in the 3D view: go forward, turn left and turn right.

E. Change the color of your shirt.

F. Change some information about yourself, such as where you are from.

Task #3: All about Communicating with Others

A. Send someone an email.

B. Read a message on the HutchWorld Bulletin Board.

C. Post a message on the HutchWorld Bulletin Board.

D. Check to see who is currently in HutchWorld.

E. Find out where the other person in HutchWorld is from.

F. Make the other person in HutchWorld a friend.

G. Chat with the other person in HutchWorld

H. Wave to the other person in HutchWorld.

I. Whisper to the other person in HutchWorld.

Task #4: All about Getting Information

A. Imagine you have never been to Seattle before. Your task is to find something to do.

B. Find out how to get to the Fred Hutchinson Cancer Research Center.

C. Go to your favorite website. [Or go to Yahoo: www.yahoo.com]

D. Once you have found a website, resize the screen so you can see the whole web page.

Figure 10.6 A sample of the structured tasks used in the HutchWorld evaluation.

Task #5: All about Entertainment

A. Find a game to play.
B. Get a gift from a Gift Cart and send yourself a gift.
C. Go and open your gift.

Figure 10.6 *(continued).*

During the study, a member of the development team role-played being a participant so that the real participants would be sure to have someone with whom to interact. The evaluator also asked the participants to fill out a short questionnaire after completing the tasks, with the aim of collecting their opinions about their experiences with HutchWorld. The questionnaire asked:

- What did you like about HutchWorld?
- What did you not like about HutchWorld?
- What did you find confusing or difficult to use in HutchWorld?
- How would you suggest improving HutchWorld?

Test 2: What was learned from the usability tests?

When running the usability tests, the team collected masses of data that they had to make sense of by systematical analysis. The following discussion offers a snapshot of their findings. Some participants' problems started right at the beginning of the five-minute exploration. The login page referred to "virtual worlds" rather than the expected HutchWorld and, even though this might seem trivial, it was enough to confuse some users. This isn't unusual; developers tend to overlook small things like this, which is why evaluation is so important. Even careful, highly skilled developers like this team tend to forget that users do not speak their language. Fortunately, finding the "go" button was fairly straightforward. Furthermore, most participants read the welcome message and used the navigation list, and over half used the chat buttons, managed to move around the 3D world, and read the overview. But only one-third chatted and used the navigation buttons. The five-minute free-exploration data was also analyzed to determine what people thought of HutchWorld and how they commented upon the 3D view, the chat area, and the browse area.

Users' performance on the structured tasks was analyzed in detail and participant ratings were tabulated. Participants rated the tasks on a scale of 1–3 where 1 = easy, 2 = OK, 3 = difficult, and bold = needed help. Any activity that received an average rating above 1.5 across participants was deemed to need detailed review by the team. Figure 10.7 shows a fragment of the summary of the analysis.

In addition, the team analyzed all the problems that they observed during the tests. They then looked at all their data and drew up a table of issues, noting whether they were a priority to fix and listing recommendations for changes.

Participant number:	1	2	3	4	5	6	7	Average
Background Information								
Sex	F	F	M	M	F	M	M	3F, 4M
Age	37	41	43	54	46	44	21	40.9
years of college	4	2	4	4	4	1	2	3.0
hours of chat use in past year	0	3	0	0	365	200	170	105.4
hours of web use in past year	9	11	36	208	391	571	771	285.3
Structured Tasks								
Identify 3D view	1	1	1	1	1	1	1	1.0
Identity self in 3D view	1	2	1	1	1	1	1	1.1
Get a map view of 3D view	1	2	2	1	2	**3**	1	1.7
Walk in 3D view	1	3	2	1	**3**	2	1	1.9
Change color of shirt	1	1	**3**	**3**	2	3	2	2.1
Change where self is from	1	1	**3**	1	1	3	1	1.6
Find place to send email	1	**3**	3	1	3	2	2	2.1
Read a bulletin board message	2	1	**3**	1	1	1	–	1.5
Post a bulletin board message	1	**3**	3	3	2	2	–	2.3
Check to see who is currently on	1	3	1	**3**	2	3	2	2.1
Find out where the other person is from	1	1	2	1	1	3	2	1.6
Make the other person a friend	1	1	3	1	1	2	1	1.4
Chat with the other person	**3**	1	**3**	1	1	**3**	1	1.9
Wave to the other person	1	1	1	1	1	1	1	1.0
Whisper to the other person	1	**3**	2	2	1	2	1	1.7
Find something to do in Seattle	2	1	2	1	1	1	2	1.4
Find out how to get to FHCRC	1	3	3	2	1	1	2	1.9
Go to a website	1	**3**	2	**3**	**3**	1	1	2.0
Resize web screen	1	3	2	2	2	**3**	1	2.0
Find a game to play	1	1	2	1	1	1	2	1.3
Send self a gift	1	**3**	**3**	**3**	**3**	**3**	3	2.7
Open gift	**3**	1	2	3	**3**	**3**	**3**	2.6
Participant Average:	1.3	1.9	2.2	1.7	**1.7**	2.0	1.6	

The following descriptions provide examples of some of the problems participants experience.

Get map **view.** People generally did not immediately know how to find the map view. However, they knew to look in the chat buttons, and by going through the buttons they found the map view.

Walk in *3D* view. People found the use of the mouse to move the avatar awkward, especially when they were trying to turn around. However, once they were used to using the mouse they had no difficulty. For a couple of people, it was not clear to them that they should click on the avatar and drag it in the desired direction. A couple of people tried to move by clicking the place they wanted to move to.

Figure 10.7 Participant information and ratings of difficulty in completing the structured tasks. 1 = easy, 2 = okay, 3 = difficult and bold = needed help.

Issue#	Issue Priority	Issue	Recommendation
1	high	Back button sometimes not working.	Fix back button.
2	high	People are not paying attention to navigation buttons.	Make navigation buttons more prominent.
3	low	Fonts too small, hard to read for some people.	Make it possible to change fonts. Make the font colors more distinct from the background color.
4	low	When navigating, people were not aware overview button would take them back to the main page.	Change the overview button to a home button, change the wording of the overview page accordingly.
5	medium	"Virtual worlds" wording in login screen confusing.	Change wording to "HutchWorld".
6	high	People frequently clicking on objects in 3D view expecting something to happen.	Make the 3D view have links to web pages. For example, when people click on the help desk the browser area should show the help desk information.
7	low	People do not readily find map view button.	Make the icon on the map view button more map-like.
8	medium	Moving avatar with mouse took some getting used to.	Encourage the use of the keyboard. Mention clicking and dragging the avatar in the welcome.
9	low	People wanted to turn around in 3D view, but it was awkward to do so.	Make one of the chat buttons a button that lets you turn around.
10	medium	Confusion about the real worldlvirtual world distinction.	Change wording of overview description, to make clear Hutch-World is a "virtual" place made to "resemble" the FHCRC, and is a place where anybody can go.
11	high	People do not initially recognize that other real people could be in HutchWorld, that they can talk to them and see them.	Change wording of overview description, to make clear Hutch-World is a place to "chat" with others who are "currently in" the virtual HutchWorld.
12	high	People not seeing/finding the chat window. Trying to chat to people from the people list where other chat-like features are (whisper, etc.)	Make chat window more prominent. Somehow link chat-like features of navigation list to chat window. Change wording of chat window. Instead of type to speak here. type to chat here.

Figure 10.8 A fragment of the table showing problem rankings.

13	low	Who is here list and who has been here list confused.	Spread them apart more in the people list.
14	medium	Difficulty in finding who is here.	Change People button to "Who is On" button.
15	low	Went to own profile to make someone a friend.	Let people add friends at My profile
16	low	Not clear how to append/reply to a discussion in the bulletin board.	Make an append button pop up when double clicking on a topic. Change wording from "post a message" to "write a message" or "add a message".
17	low	Bulletin board language is inconsistent.	Change so it is either a bulletin board, or a discussion area.

Figure 10.8 ***(continued).***

Figure 10.8 shows part of this table. Notice that issues were ranked in priority: low, medium, **and** high. **There were** just five high-ranking problems that absolutely had to be fixed:

- The back button did not always work.
- People were not paying attention to navigation buttons, so they needed to be more prominent.
- People frequently clicked on objects in the 3D view and expected something to happen. A suggestion for fixing this was to provide links to a web page.
- People did not realize that there could be other real people in the 3D world with whom they could chat, so the wording in the overview description had to be changed.
- People were not noticing the chat window and instead were trying to chat to people in the participant list. The team needed to clarify the instructions about where to chat.

In general, most users found the redesigned software easy to use with little instruction. By running a variety of tests, the informal onsite test, and the formal usability test, key problems were identified at an early stage and various usability issues could be fixed before the actual deployment of the software.

10.3.3 Was it tested again?

Following the usability testing, there were more rounds of observation and testing with six new participants, two males and four females. These tests followed the same general format as those just described but this time they tested multiple users at once, to ensure that the virtual world supported multiuser interactions. The tests were also more detailed and focused. This time the results were more positive, but

DILEMMA When Is It Time to Stop Testing?

Was HutchWorld good enough after these evaluations? When has enough testing been done? This frequently asked question is difficult to answer. Few developers have the luxury of testing as thoroughly as John Gould and his colleagues when developing the 1984 Olympic Messaging System (Gould and Lewis, 1990), or even as much as Microsoft's HutchWorld team. Since every test you do will reveal some area where improvement can be made, you cannot assume that there will be a time when the system is perfect: no system is ever perfect. Normally schedule and budget constraints determine when to stop. Joseph Dumas and Ginny Redish, established usability consultants, point out that for iterative design and testing to be successful, each test should take as little time as possible while still yielding useful information and without burdening the team (Dumas and Redish, 1999).

of course there were still usability problems to be fixed. Then the question arose: what to do next? In particular, had they done enough testing (see Dilemma)?

After making a few more fixes, the team stopped usability testing with specific tasks. But the story didn't end here. The next step was to show HutchWorld to cancer patients and caregivers in a focus-group setting at the Fred Hutchinson Cancer Research Center to get their feedback on the final version. Once the team made adjustments to HutchWorld in response to the focus-group feedback, the final step was to see how well HutchWorld worked in a real clinical environment. It was therefore taken to a residential building used for long-term patient and family stays that was fully wired for Internet access. Here, the team observed what happened when it was used in this natural setting. In particular, they wanted to find out how HutchWorld would integrate with other aspects of patients' lives, particularly with their medical care routines and their access to social support. This informal observation allowed them to examine patterns of use and to see who used which parts of the system, when, and why.

10.3.4 Looking *to* the future

Future studies were planned to evaluate the effects of the computers and the software in the Fred Hutchinson Center. The focus of these studies will be the social support and wellbeing of patients and their caregivers in two different conditions. There will be a control condition in which users (i.e., patients) live in the residential building without computers and an experimental condition in which users live in similar conditions but with computers, Internet access, and HutchWorld. The team will evaluate the user data (performance and observation) and surveys collected in the study to investigate key questions, including:

- How does the computer and software impact the social wellbeing of patients and their caregivers?
- What type of computer-based communication best supports this patient community?
- What are the general usage patterns? i.e., which features were used and at what time of day were they used, etc.?

- How might any medical facility use computers and software like Hutch-World to provide social support for its patients and caregivers?

There is always more to learn about the efficacy of a design and how much users enjoy using a product, especially when designing innovative products like HutchWorld for new environments. This study will provide a longer-term view of how HutchWorld is used in its natural environment that is not provided by the other evaluations. It's an ambitious plan because it involves a comparison between two different environmental settings, one that has computers and HutchWorld and one that doesn't (see Chapter 13 for more on experimental design).

ACTIVITY 10.3

(a) The case study does not say much about early evaluation to test the conceptual design shown in Figure 10.5. What do you think happened?

(b) The evaluators recorded the gender of participants and noted their previous experience with similar systems. Why is this important?

(c) Why do you think it was important to give participants a five-minute exploration period?

(d) ***Triangulation*** is a term that describes how different perspectives are used to understand a problem or situation. Often different techniques are used in triangulation. Which techniques were triangulated in the evaluations of the HutchWorld prototype?

(e) The evaluators collected participants' opinions. What kinds of concerns do you think participants might have about using HutchWorld? Hints: personal information, medical information, communicating feelings, etc.

Comment

(a) There was probably much informal discussion with representative users: patients, medical staff, relatives, friends, and caregivers. The team also visited the clinic and hospital and observed what happened there. They may also have discussed this with the physicians and administrators.

(b) It is possible that our culture causes men and women to react differently in certain circumstances. Experience is an even more important influence than gender, so knowing how much previous experience users have had with various types of computer systems enables evaluators to make informed judgments about their performance. Experts and novices, for example, tend to behave very differently.

(c) The evaluators wanted to see how participants reacted to the system and whether or not they could log on and get started. The exploration period also gave the participants time to get used to the system before doing the set tasks.

(d) Data was collected from the five-minute exploration, from performance on the structured tasks, and from the user satisfaction questionnaire.

(e) Comments and medical details are personal and people want privacy. Patients might be concerned about whether the medical information they get via the computer and from one another is accurate. Participants might be concerned about how clearly and accurately they are communicating because non-verbal communication is reduced online.

10.4 Discussion

In both HutchWorld and the 1984 Olympic Messaging System, a variety of evaluation techniques were used at different stages of design to answer different questions. "Quick and dirty" observation, in which the evaluators informally examine how a prototype is used in the natural environment, was very useful in early design. Following this with rounds of usability testing and redesign revealed important usability problems. However, usability testing alone is not sufficient. Field studies were needed to see how users used the system in their natural environments, and sometimes the results were surprising. For example, in the OMS system users from different cultures behaved differently. A key issue in the HutchWorld study was how use of the system would fit with patients' medical routines and changes in their physical and emotional states. Users' opinions also offered valuable insights. After all, if users don't like a system, it doesn't matter how successful the usability testing is: they probably won't use it. Questionnaires and interviews were used to collect user's opinions.

An interesting point concerns not only how the different techniques can be used to address different issues at different stages of design, but also how these techniques complement each other. Together they provide a broad picture of the system's usability and reveal different perspectives. In addition, some techniques are better than others for getting around practical problems. This is a large part of being a successful evaluator. In the HutchWorld study, for example, there were not many users, so the evaluators needed to involve them sparingly. For example, a technique requiring 20 users to be available at the same time was not feasible in the HutchWorld study, whereas there was no problem with such an approach in the OMS study. Furthermore, the OMS study illustrated how many different techniques, some of which were highly opportunistic, can be brought into play depending on circumstances. Some practical issues that evaluators routinely have to address include:

- what to do when there are not many users
- how to observe users in their natural location (*i.e.*, field studies) without disturbing them

  having appropriate equipment available
- dealing with short schedules and low budgets
- not disturbing users or causing them duress or doing anything unethical
- collecting "useful" data and being able to analyze it
- selecting techniques that match the evaluators' expertise

There are many evaluation techniques from which to choose and these practical issues play a large role in determining which are selected. Furthermore, selection depends strongly on the stage in the design and the particular questions to be answered. In addition, each of the disciplines that contributes to interaction design has preferred bodies of theory and techniques that can influence this choice. These issues are discussed further in the next chapter.

Assignment

1. Reconsider the HutchWorld design and evaluation case study and note ***what*** was evaluated, ***why*** and ***when,*** and ***what*** was learned at each stage?
2. How was the design advanced after each round of evaluation?
3. What were the main constraints that influenced the evaluation?
4. How did the stages and choice of techniques build on and complement each other (i.e., triangulate)?
5. Which parts of the evaluation were directed at usability goals and which at user experience goals? Which additional goals not mentioned in the study could the evaluations have focused upon?

Summary

The aim of this chapter was to introduce basic evaluation concepts that will be revisited and built on in the next four chapters. We selected the HutchWorld case study because it illustrates how a team of designers evaluated a novel system and coped with a variety of practical constraints. It also shows how different techniques are needed for different purposes and how techniques are used together to gain different perspectives on a product's usability. This study highlights how the development team paid careful attention to usability and user experience goals as they designed and evaluated their system.

Key points

- Evaluation and design are very closely integrated in user-centered design.
- Some of the same techniques are used in evaluation as in the activity of establishing requirements and identifying users' needs, but they are used differently (e.g., interviews and questionnaires, etc.).
- Triangulation involves using combinations of techniques in concert to get different perspectives or to examine data in different ways.

  Dealing with constraints, such as gaining access to users or accommodating users' routines, is an important skill for evaluators to develop.

Further reading

CHENG, L., STONE, L., FARNHAM, S., CLARK, A. M., AND ZANER-GODSEY, M. (2000) ***Hutchworld: Lessons Learned. A Collaborative Project: Fred Hutchinson Cancer Research Center & Microsofi Research.*** In the Proceedings of the Virtual Worlds Conference 2000, Paris, France. This paper describes the HutchWorld study and, as the title suggests, it discusses the design lessons that were learned. It also describes the evaluation studies in more detail.

GOULD, J. D., BOIES, S. J., LEVY, S., RICHARDS, J. T., AND SCHOONARD, J. (1990). The 1984 Olympic Message System: A test of behavioral principles of system design. In J. Preece and L. Keller (eds.), ***Human-Computer Interaction (Readings).*** Prentice Hall International Ltd., Hemel Hempstead, UK: 260–283. This edited paper tells the story of the design and evaluation of the OMS.

GOULD, J. D., BOIES, S. J., LEVY, S., RICHARDS, J. T., AND SCHOONARD, J. (1987). The 1984 Olympic Message System: a test of behavioral principles of systems design. ***Communications of the ACM,*** 30(9), 758–769. This is the original, full version of the OMS paper.

Chapter 11

An evaluation framework

11.1 Introduction

Designing useful and attractive products requires skill and creativity. As products evolve from initial ideas through conceptual design and prototypes, iterative cycles of design and evaluation help to ensure that they meet users' needs. But how do evaluators decide *what* and *when* to evaluate? The HutchWorld case study in the previous chapter described how one team did this, but the circumstances surrounding every product's development are different. Certain techniques work better for some than for others.

Identifying usability and user experience goals is essential for making every product successful, and this requires understanding users' needs. The role of evaluation is to make sure that this understanding occurs during all the stages of the product's development. The skillful and sometimes tricky part of doing this is knowing what to focus on at different stages. Initial requirements get the design process started, but, as you have seen, understanding requirements tends to happen by a process of negotiation between designers and users. As designers understand users' needs better, their designs reflect .this understanding. Similarly, as users see and experience design ideas, they are able to give better feedback that enables the designers to improve their designs further. The process is cyclical, with evaluation playing a key role in facilitating understanding between designers and users.

Evaluation is driven by questions about how well the design or particular aspects of it satisfy users' needs. Some of these questions provide high-level goals to guide the evaluation. Others are much more specific. For example, can users find a particular menu item? Is a graphic useful and attractive? Is the product engaging? Practical constraints also play a big role in shaping evaluation plans: tight schedules, low budgets, or little access to users constrain what evaluators can do. You read in chapter 10 how the HutchWorld team had to plan its evaluation around hospital routines and patients' health.

Experienced designers get to know what works and what doesn't, but those with little experience can find doing their first evaluation daunting. However, with careful advance planning, problems can be spotted and ways of dealing with them can be found. Planning evaluation studies involves thinking about key issues and asking questions about the process. In this chapter we propose the DECIDE framework to help you do this.

The main aims of this chapter are to:

- Continue to explain the key concepts and terms used to discuss evaluation.
- Describe the evaluation paradigms and techniques used in interaction design.
- Discuss the conceptual, practical, and ethical issues to be considered when planning evaluation.
- Introduce the DECIDE framework to help you plan your own evaluation studies.

11.2 Evaluation paradigms and techniques

Before we describe the techniques used in evaluation studies, we shall start by proposing some key terms. Terminology in this field tends to be loose and often confusing so it is a good idea to be clear from the start what you mean. We start with the much-used term ***user*** studies, defined by Abigail Sellen in her interview at the end of Chapter 4 as follows: "user studies essentially involve looking at how people behave either in their natural [environments], or in the laboratory, both with old technologies and with new ones." Any kind of evaluation, whether it is a user study or not, is guided either explicitly or implicitly by a set of beliefs that may also be underpinned by theory. These beliefs and the practices (i.e., the methods or techniques) associated with them are known as an evaluation paradigm, which you should not confuse with the "interaction paradigms" discussed in Chapter 2. Often evaluation paradigms are related to a particular discipline in that they strongly influence how people from the discipline think about evaluation. Each paradigm has particular methods and techniques associated with it. So that you are not confused, we want to state explicitly that we will not be distinguishing between methods and techniques. We tend to talk about techniques, but you may find that other books call them methods. An example of the relationship between a paradigm and the techniques used by evaluators following that paradigm can be seen for usability testing, which is an applied science and engineering paradigm. The techniques associated with usability testing are: user testing in a controlled environment; observation of user activity in the controlled environment and the field; and questionnaires and interviews.

11.2.1 Evaluation paradigms

In this book we identify four core evaluation paradigms: (1) "quick and dirty" evaluations; (2) usability testing; (3) field studies; and (4) predictive evaluation. Other texts may use slightly different terms to refer to similar paradigms.

"Quick and dirty" evaluation

A "quick and dirty" evaluation is a common practice in which designers informally get feedback from users or consultants to confirm that their ideas are in line with users' needs and are liked. "Quick and dirty" evaluations can be done at any stage and the emphasis is on fast input rather than carefully documented findings. For example, early in design developers may meet informally with users to get feedback on ideas for a new product (Hughes et al., 1994). At later stages similar meetings may occur to try out an idea for an icon, check whether a graphic is liked, or confirm that information has been appropriately categorized on a webpage. This approach is often called "quick and dirty" because it is meant to be done in a short space of time. Getting this kind of feedback is an essential ingredient of successful design.

As discussed in Chapter 9, any involvement with users will be highly informative and you can learn a lot early in design by observing what people do and talking to them informally. The data collected is usually descriptive and informal and it is fed back into the design process as verbal or written notes, sketches and anecdotes, etc. Another source comes from consultants, who use their knowledge of user behavior, the market place and technical know-how, to review software quickly and provide suggestions for improvement. It is an approach that has become particularly popular in web design where the emphasis is usually on short tirnescales.

Usability testing

Usability testing was the dominant approach in the 1980s (Whiteside et al., 1998), and remains important, although, as you will see, field studies and heuristic evaluations have grown in prominence. Usability testing involves measuring typical users' performance on carefully prepared tasks that are typical of those for which the system was designed. Users' performance is generally measured in terms of number of errors and time to complete the task. As the users perform these tasks, they are watched and recorded on video and by logging their interactions with software. This observational data is used to calculate performance times, identify errors, and help explain why the users did what they did. User satisfaction questionnaires and interviews are also used to elicit users' opinions.

The defining characteristic of usability testing is that it is ***strongly controlled*** by the evaluator (Mayhew, 1999). There is no mistaking that the evaluator is in charge! Typically tests take place in laboratory-like conditions that are controlled. Casual visitors are not allowed and telephone calls are stopped, and there is no possibility of talking to colleagues, checking email, or doing any of the other tasks that most of us rapidly switch among in our normal lives. Everything that the participant does is recorded—every keypress, comment, pause, expression, etc., so that it can be used as data.

Quantifying users' performance is a dominant theme in usability testing. However, unlike research experiments, variables are not manipulated and the typical number of participants is too small for much statistical analysis. User satisfaction data from questionnaires tends to be categorized and average ratings are presented. Sometimes video or anecdotal evidence is also included to illustrate problems that users encounter. Some evaluators then summarize this data in a usability specification so that developers can use it to test future prototypes or versions of the product against it. Optimal performance levels and minimal levels of acceptance are often specified and current levels noted. Changes in the design can then be agreed and engineered—hence the term "usability engineering." User testing is explained further in Chapter 14, how to observe users is described in Chapter 12, and issues concerned with interviews and questionnaires are explored in Chapter 13.

Field studies

The distinguishing feature of field studies is that they are done in natural settings with the aim of increasing understanding about what users do naturally and how technology impacts them. In product design, field studies can be used to (1) help identify opportunities for new technology; (2) determine requirements for design; (3) facilitate the introduction of technology; and (4) evaluate technology (Bly, 1997).

Chapter 9 introduced qualitative techniques such as interviews, observation, participant observation, and ethnography that are used in field studies. The exact choice of techniques is often influenced by the theory used to analyze the data. The data takes the form of events and conversations that are recorded as notes, or by audio or video recording, and later analyzed using a variety of analysis techniques such as content, discourse, and conversational analysis. These techniques vary considerably. In content analysis, for example, the data is analyzed into content categories, whereas in discourse analysis the use of words and phrases is examined. Artifacts are also collected. In fact, anything that helps to show what people do in their natural contexts can be regarded as data.

In this text we distinguish between two overall approaches to field studies. The first involves observing explicitly and recording what is happening, as an ***outsider*** looking on. Qualitative techniques are used to collect the data, which may then be analyzed qualitatively or quantitatively. For example, the number of times a particular event is observed may be presented in a bar graph with means and standard deviations.

In some field studies the evaluator may be an ***insider*** or even a participant. Ethnography is a particular type of insider evaluation in which the aim is to explore the details of what happens in a particular social setting. "In the context of human-computer interaction, ethnography is a means of studying work (or other activities) in order to inform the design of information systems and understand aspects of their use" (Shapiro, 1995, p. 8).

Predictive evaluation

In predictive evaluations experts apply their knowledge of typical users, often guided by heuristics, to predict usability problems. Another approach involves theoretically-based models. The key feature of predictive evaluation is that users need not be present, which makes the process quick, relatively inexpensive, and thus attractive to companies; but it has limitations.

In recent years heuristic evaluation in which experts review the software product guided by tried and tested heuristics has become popular (Nielsen and Mack, 1994). As mentioned in Chapter 1, usability guidelines (e.g., always provide clearly marked exits) were designed primarily for evaluating screen-based products (e.g. form fill-ins, library catalogs, etc.). With the advent of a range of new interactive products (e.g., the web, mobiles, collaborative technologies), this original set of heuristics has been found insufficient. While some are still applicable (e.g., speak the users' language), others are inappropriate. New sets of heuristics are also needed that are aimed at evaluating different classes of interactive products. In particular, specific heuristics are needed that are tailored to evaluating web-based products, mobile devices, collaborative technologies, computerized toys, etc. These should be based on a combination of usability and user experience goals, new research findings and market research. Care is needed in using sets of heuristics. As you will see in Chapter 13, designers are sometimes led astray by findings from heuristic evaluations that turn out not to be as accurate as they at first seemed.

Table 11.1 summarizes the key aspects of each evaluation paradigm for the following issues:

- the role of users
- who controls the process and the relationship between evaluators and users during the evaluation
- the location of the evaluation
- when the evaluation is most useful
- the type of data collected and how it is analyzed
- how the evaluation findings are fed back into the design process
- the philosophy and theory that underlies the evaluation paradigms

Some other terms that you may encounter in your reading are shown in Box 11.1.

ACTIVITY 11.1 Think back to the HutchWorld case study.

(a) Which evaluation paradigms were used in the study and which were not?

(b) How could the missing evaluation paradigms have been used to inform the design and why might they not have been used?

Comment

(a) The team did some "quick and dirty" evaluation during early development but this is not stressed in their report. Usability testing played a strong role, with some tests being carried out at the Fred Hutchinson Center and later tests in Microsoft's usability laboratories. Field studies are not strongly featured, but the team does mention

Table 11.1 Characteristics of different evaluation paradigms

Evaluation paradigms	"Quick and dirty"	Usability testing	Field studies	Predictive
Role of users	Natural behavior.	To carry out set tasks.	Natural behavior.	Users generally not involved.
Who controls	Evaluators take minimum control.	Evaluators strongly in control.	Evaluators try to develop relationships with users.	Expert evaluators.
Location	Natural environment or laboratory.	Laboratory.	Natural environment.	Laboratory-oriented but often happens on customer's premises.
When used	Any time you want to get feedback about a design quickly. Techniques from other evaluation paradigms can be used–e.g., experts review software.	With a prototype or product.	Most often used early in design to check that users' needs are being met or to assess problems or design opportunities.	Expert reviews (often done by consultants) with a prototype, but can occur at any time. Models are used to assess specific aspects of a potential design.
Type of data	Usually qualitative, informal descriptions.	Quantitative. Sometimes statistically validated. Users' opinions collected by questionnaire or interview.	Qualitative descriptions often accompanied with sketches, scenarios, quotes, other artifacts.	List of problems from expert reviews. Quantitative figures from model, e.g., how long it takes to perform a task using two designs.
Fed back into design by . . .	Sketches, quotes, descriptive report.	Report of performance measures, errors etc. Findings provide a benchmark for future versions.	Descriptions that include quotes, sketches, anecdotes, and sometimes time logs.	Reviewers provide a list of problems, often with suggested solutions. Times calculated from models are given to designers.
Philosophy	User-centered, highly practical approach.	Applied approach based on experimentation, i.e., usability engineering.	May be objective observation or ethnographic.	Practical heuristics and practitioner expertise underpin expert reviews. Theory underpins models.

BOX 11.1 Some Definitions to Help You

Objective and subjective Objective evaluations are based on techniques that use quantitative measurement rather than users' or experts' opinions. Subjective evaluations are based on opinions and anecdotes.

Quantitative and qualitative Quantitative evaluations involve measurements, whereas qualitative evaluations involve descriptions and anecdotes. Quantitative evaluations tend to be seen as objective and impartial, whereas many qualitative evaluations tend to be seen as subjective but this isn't necessarily true.

Laboratory and field or naturalistic studies Laboratory studies occur in controlled environments. They may be done in a specially built laboratory or in a space that is specially adapted for the purpose. Field or naturalistic studies are situated in the real-world context in which the system is or will be used.

observing how patients used HutchWorld in the Center. Field studies were planned in which patients, who have access to HutchWorld and the web, could be systematically compared with another group who does not have these facilities. However, distinguishing between evaluation paradigms isn't always clear-cut. In practice elements typically found in one may be transferred to another (e.g., the controlled approach the HutchWorld team planned to use in the field). The only evaluation paradigm that is not mentioned in the study is predictive evaluation.

(b) Expert reviews could have been done any time during its development but the team may have thought they were not needed, or there wasn't time, or perhaps they were performed but not reported.

11.2.2 Techniques

There are many evaluation techniques and they can be categorized in various ways, but in this text we will examine techniques for:

- observing users
- asking users their opinions
- asking experts their opinions
- testing users' performance
- modeling users' task performance to predict the efficacy of a user interface

The brief descriptions below offer an ***overview*** of each category, which we discuss in detail in the next three chapters. Be aware that some techniques are used in different ways in different evaluation paradigms.

Observing users

Observation techniques help to identify needs leading to new types of products and help to evaluate prototypes. Notes, audio, video, and interaction logs are well-known ways of recording observations and each has benefits and drawbacks. Obvious challenges for evaluators are how to observe without disturbing the people being observed and how to analyze the data, particularly when large quantities of video data are collected or when several different types must be integrated to tell the story (e.g., notes, pictures, sketches from observers). You met several observation techniques in Chapter 7 in the context of the requirements activity; in Chapter **12** we will focus on how they are used in evaluation.

Asking users

Asking users what they think of a product—whether it does what they want; whether they like it; whether the aesthetic design appeals; whether they had problems using it; whether they want to use it again—is an obvious way of getting feedback. Interviews and questionnaires are the main techniques for doing this. The questions asked can be unstructured or tightly structured. They can be asked of a few people or of hundreds. Interview and questionnaire techniques are also being developed for use with email and the web. We discuss these techniques in Chapter **13.**

Asking experts

Software inspections and reviews are long established techniques for evaluating software code and structure. During the 1980s versions of similar techniques were developed for evaluating usability. Guided by heuristics, experts step through tasks role-playing typical users and identify problems. Developers like this approach because it is usually relatively inexpensive and quick to perform compared with laboratory and field evaluations that involve users. In addition, experts frequently suggest solutions to problems. In Chapter 13 you will learn a few inspection techniques for evaluating usability.

User testing

Measuring user performance to compare two or more designs has been the bedrock of usability testing. As we said earlier when discussing usability testing, these tests are usually conducted in controlled settings and involve typical users performing typical, well-defined tasks. Data is collected so that performance can be analyzed. Generally the time taken to complete a task, the number of errors made, and the navigation path through the product are recorded. Descriptive statistical measures such as means and standard deviations are commonly used to report the results. In Chapter **14** you will learn the basics of user testing and how it differs from scientific experiments.

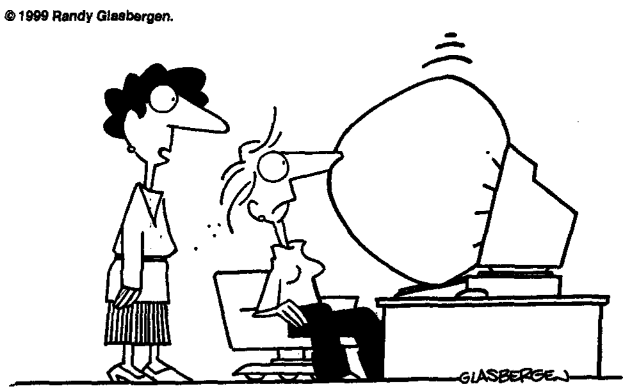
Modeling users' task performance

There have been various attempts to model human-computer interaction so as to predict the efficiency and problems associated with different designs at an early stage without building elaborate prototypes. These techniques are successful for systems with limited functionality such as telephone systems. GOMS and the keystroke model are the best known techniques. They have already been mentioned in Chapter 3 and in Chapter **14** we examine their role in evaluation.

Table **11.2** summarizes the categories of techniques and indicates how they are commonly used in the four evaluation paradigms.

Table 11.2 The relationship between evaluation paradigms and techniques.

	Evaluation paradigms			
Techniques	**"Quick and dirty"**	**Usability testing**	**Field studies**	**Predictive**
Observing users	Important for seeing how users behave in their natural environments.	Video and interaction logging, which can be analyzed to identify errors, investigate routes through the software, or calculate performance time.	Observation is the central part of any field study. In ethnographic studies evaluators immerse themselves in the environment. In other types of studies the evaluator looks on objectively.	N/A
Asking users	Discussions with users and potential users individually, in groups or focus groups.	User satisfaction questionnaires are administered to collect users' opinions. Interviews may also be used to get more details.	The evaluator may interview or discuss what she sees with participants. Ethnographic interviews are used in ethnographic studies.	N/A
Asking experts	To provide critiques (called "crit reports") of the usability of a prototype.	N/A	N/A	Experts use heuristics early in design to predict the efficacy of an interface.
User testing	N/A	Testing typical users on typical tasks in a controlled laboratory-like setting is the cornerstone of usability testing.	N/A	N/A
Modeling users' task performance	N/A	N/A	N/A	Models are used to predict the efficacy of an interface or compare performance times between versions.



"It's the latest innovation in office safety. When your computer crashes, an *air* bag is activated so you won't bang your head in frustration."

11.3 DECIDE: A framework to guide evaluation

Well-planned evaluations are driven by clear goals and appropriate questions (Basili et al., 1994). To guide our evaluations we use the DECIDE framework, which provides the following checklist to help novice evaluators:

1. Determine the overall ***goals*** that the evaluation addresses.
2. Explore the specific ***questions*** to be answered.
3. Choose the ***evaluation paradigm*** and ***techniques*** to answer the questions.
4. Identify the ***practical issues*** that must be addressed, such as selecting participants.
5. Decide how to deal with the ***ethical issues.***
6. Evaluate, interpret, and present the ***data.***

11.3.1 Determine the goals

What are the high-level goals of the evaluation? Who wants it and why? An evaluation to help clarify user needs has different goals from an evaluation to determine the best metaphor for a conceptual design, or to fine-tune an interface, or to examine how technology changes working practices, or to inform how the next version of a product should be changed.

Goals should guide an evaluation, so determining what these goals are is the first step in planning an evaluation. For example, we can restate the general goal statements just mentioned more clearly as:

- Check that the evaluators have understood the users' needs.
- Identify the metaphor on which to base the design.

- Check to ensure that the final interface is consistent.
- Investigate the degree to which technology influences working practices.
- Identify how the interface of an existing product could be engineered to improve its usability.

These goals influence the evaluation approach, that is, which evaluation paradigm guides the study. For example, engineering a user interface involves a quantitative engineering style of working in which measurements are used to judge the quality of the interface. Hence usability testing would be appropriate. Exploring how children talk together in order to see if an innovative new groupware product would help them to be more engaged would probably be better informed by a field study.

11.3.2 Explore the questions

In order to make goals operational, questions that must be answered to satisfy them have to be identified. For example, the goal of finding out why many customers prefer to purchase paper airline tickets over the counter rather than e-tickets can be broken down into a number of relevant questions for investigation. What are customers' attitudes to these new tickets? Perhaps they don't trust the system and are not sure that they will actually get on the flight without a ticket in their hand. Do customers have adequate access to computers to make bookings? Are they concerned about security? Does this electronic system have a bad reputation? Is the user interface to the ticketing system so poor that they can't use it? Maybe very few people managed to complete the transaction.

Questions can be broken down into very specific sub-questions to make the evaluation even more specific. For example, what does it mean to ask, "Is the user interface poor?": Is the system difficult to navigate? Is the terminology confusing because it is inconsistent? Is response time too slow? Is the feedback confusing or maybe insufficient? Sub-questions can, in turn, be further decomposed into even finer-grained questions, and so on.

11.3.3 Choose the evaluation paradigm and techniques

Having identified the goals and main questions, the next step is to choose the evaluation paradigm and techniques. As discussed in the previous section, the evaluation paradigm determines the kinds of techniques that are used. Practical and ethical issues (discussed next) must also be considered and trade-offs made. For example, what seems to be the most appropriate set of techniques may be too expensive, or may take too long, or may require equipment or expertise that is not available, so compromises are needed.

As you saw in the HutchWorld case study, combinations of techniques can be used to obtain different perspectives. Each type of data tells the story from a different point of view. Using this triangulation reveals a broad picture.

11.3.4 Identify the practical issues

There are many practical issues to consider when doing any kind of evaluation and it is important to identify them ***before*** starting. Some issues that should be considered include users, facilities and equipment, schedules and budgets, and evaluators' expertise. Depending on the availability of resources, compromises may involve adapting or substituting techniques.

Users

It goes without saying that a key aspect of an evaluation is involving ***appropriate*** users. For laboratory studies, users must be found and screened to ensure that they represent the user population to which the product is targeted. For example, usability tests often need to involve users with a particular level of experience e.g., novices or experts, or users with a range of expertise. The number of men and women within a particular age range, cultural diversity, educational experience, and personality differences may also need to be taken into account, depending on the kind of product being evaluated. In usability tests participants are typically screened to ensure that they meet some predetermined characteristic. For example, they might be tested to ensure that they have attained a certain skill level or fall within a particular demographic range. Questionnaire surveys require large numbers of participants so ways of identifying and reaching a representative sample of participants are needed. For field studies to be successful, an appropriate and accessible site must be found where the evaluator can work with the users in their natural setting.

Another issue to consider is how the users will be involved. The tasks used in a laboratory study should be representative of those for which the product is designed. However, there are no written rules about the length of time that a user should be expected to spend on an evaluation task. Ten minutes is too short for most tasks and two hours is a long time, but what is reasonable? Task times will vary according to the type of evaluation, but when tasks go on for more than 20 minutes, consider offering breaks. It is accepted that people using computers should stop, move around and change their position regularly after every 20 minutes spent at the keyboard to avoid repetitive strain injury. Evaluators also need to put users at ease so they are not anxious and will perform normally. Even when users are paid to participate, it is important to treat them courteously. At no time should users be treated condescendingly or made to feel uncomfortable when they make mistakes. Greeting users, explaining that it is the system that is being tested and not them, and planning an activity to familiarize them with the system before starting the task all help to put users at ease.

Facilities and equipment

There are many practical issues concerned with using equipment in an evaluation. For example, when using video you need to think about how you will do the recording: how many cameras and where do you put them? Some people are disturbed by having a camera pointed at them and will not perform normally, so how can you avoid making them feel uncomfortable? Spare film and batteries may also be needed.

Schedule and budget constraints

Time and budget constraints are important considerations to keep in mind. It might seem ideal to have 20 users test your interface, but if you need to pay them, then it could get costly. Planning evaluations that can be completed on schedule is also important, particularly in commercial settings. However, as you will see in the interview with Sara Bly in the next chapter, there is never enough time to do evaluations as you would ideally like, so you have to compromise and plan to do a good job with the resources and time available.

Expertise

Does the evaluation team have the expertise needed to do the evaluation? For example, if no one has used models to evaluate systems before, then basing an evaluation on this approach is not sensible. It is no use planning to use experts to review an interface if none are available. Similarly, running usability tests requires expertise. Analyzing video can take many hours, so someone with appropriate expertise and equipment must be available to do it. If statistics are to be used, then a statistician should be consulted before starting the evaluation and then again later for analysis, if appropriate.

ACTIVITY 11.2 Informal observation, user performance testing, and questionnaires were used in the HutchWorld case study. What practical issues are mentioned in the case study? What other issues do you think the developers had to take into account?

Comment No particular practical issues are mentioned for the informal observation, but there probably were restrictions on where and what the team could observe. For example, it is likely that access would be denied to very sick patients and during treatment times. Not surprisingly, user testing posed more problems, such as finding participants, putting equipment in place, managing the tests, and underestimation of the time needed to work in a hospital setting compared with the fast production times at Microsoft.

11.3.5 Decide how to deal with the ethical issues

The Association for Computing Machinery (ACM) and many other professional organizations provide ethical codes (Box 11.2) that they expect their members to uphold, particularly if their activities involve other human beings. For example, people's privacy should be protected, which means that their name should not be associated with data collected about them or disclosed in written reports (unless they give permission). Personal records containing details about health, employment, education, financial status, and where participants live should be confidential. Similarly,

BOX 11.2 ACM Code of Ethics

The ACM code outlines many ethical issues that professionals are likely to face. Section 1 outlines fundamental ethical considerations, while section 2 addresses additional, more specific considerations of professional conduct. Statements in section 3 pertain more specifically to individuals who have a leadership role. Principles involving compliance with the code are given in section 4. Two principles of particular relevance to this discussion are:

- Ensure that users and those who will be affected by a system have their needs clearly articulated during the assessment of requirements; later the system must be validated to meet requirements.
- Articulate and support policies that protect the dignity of users and others affected by a computing system.

it should not be possible to identify individuals from comments written in reports. For example, if a focus group involves nine men and one woman, the pronoun "she" should not be used in the report because it will be obvious to whom it refers.

Most professional societies, universities, government and other research offices require researchers to provide information about activities in which human participants will be involved. This documentation is reviewed by a panel and the researchers are notified whether their plan of work, particularly the details about how human participants will be treated, is acceptable.

People give their time and their trust when they agree to participate in an evaluation study and both should be respected. But what does it mean to be respectful to users? What should participants be told about the evaluation? What are participants' rights? Many institutions and project managers require participants to read and sign an informed consent form similar to the one in Box 11.3. This form explains the aim of the tests or research and promises participants that their personal details and performance will not be made public and will be used only for the purpose stated. It is an

BOX 11.3 Informed Consent Form

I state that I am over 18 years of age and wish to participate in a program of research being conducted by Dr. Hoo Hah and his colleagues at the College of Extraordinary Research, University of Highland, College Estate.

The purpose of the research is to assess the usability of HighFly, a website developed at the National Library to provide information to the general public.

The procedures involve the monitored use of HighFly. I will be asked to perform specific tasks using HighFly. I will also be asked open-ended questions about HighFly and my experience using it.

All information collected in the study is confidential, and my name will not be identified at any time.

I understand that I am free to ask questions or to withdraw from participation at any time without penalty.

____________________ __________

Signature of Participant Date

(Adapted from Cogdill, 1999.)

agreement between the evaluator and the evaluation participants that helps to confirm the professional relationship that exists between them. If your university or organization does not provide such a form it is advisable to develop one, partly to protect yourself in the unhappy event of litigation and partly because the act of constructing it will remind you what you should consider.

The following guidelines will help ensure that evaluations are done ethically and that adequate steps to protect users' rights have been taken.

- Tell participants the goals of the study and exactly what they should expect if they participate. The information given to them should include outlining the process, the approximate amount of time the study will take, the kind of data that will be collected, and how that data will be analyzed. The form of the final report should be described and, if possible, a copy offered to them. Any payment offered should also be clearly stated.
- Be sure to explain that demographic, financial, health, or other sensitive information that users disclose or is discovered from the tests is confidential. A coding system should be used to record each user and, if a user must be identified for a follow-up interview, the code and the person's demographic details should be stored separately from the data. Anonymity should also be promised if audio and video are used.
- Make sure users know that they are free to stop the evaluation at any time if they feel uncomfortable with the procedure.
- Pay users when possible because this creates a formal relationship in which mutual commitment and responsibility are expected.
- Avoid including quotes or descriptions that inadvertently reveal a person's identity, as in the example mentioned above, of avoiding use of the pronoun "she" in the focus group. If quotes need to be reported, e.g., to justify conclusions, then it is convention to replace words that would reveal the source with representative words, in square brackets. We used this convention in Boxes 9.2 and 9.3.
- Ask users' permission in advance to quote them, promise them anonymity, and offer to show them a copy of the report before it is distributed.

The general rule to remember when doing evaluations is ***do unto others only what you would not mind being done to you.***

ACTIVITY 11.3 Think back to the HutchWorld case study. What ethical issues did the developers have to consider?

Comment The developers of HutchWorld considered all the issues listed above. In addition, because the study involved patients, they had to be particularly careful that medical and other personal information was kept confidential. They were also sensitive to the fact that cancer patients may become too tired or sick to participate so they reassured them that they could stop at any time if the task became onerous.

ACTIVITY 11.4 Usability laboratories often have a one-way mirror that allows evaluators to watch users doing their tasks in the laboratory without the users seeing the evaluators. Should users be told that they are being watched?

Comment Yes, users should be told that they will be observed through a one-way mirror. It is unethical not to. This honest approach will not compromise the study because users forget about the mirror as they get more absorbed in their tasks. Telling users what is happening helps to build trust.

The recent explosion in Internet and web usage has resulted in more research on how people use these technologies and their effects on everyday life. Consequently, there are many projects in which developers and researchers are logging users' interactions, analyzing web traffic, or examining conversations in chatrooms, bulletin boards, or on email. Unlike most previous evaluations in human-computer interaction, these studies can be done without users knowing that they are being studied. This raises ethical concerns, chief among which are issues of privacy, confidentiality, informed consent, and appropriation of others' personal stories (Sharf, 1999). People often say things online that they would not say face to face. Furthermore, many people are unaware that personal information **they share** online can be read by someone with technical know-how years later, even after they have deleted it from their personal mailbox (Erickson et al., 1999).

ACTIVITY 11.5 Studies of user behavior on the Internet may involve logging users' interactions and keeping a copy of their conversations with others. Should users be told that this is happening?

DILEMMA What Would You Do?

There is a famous and controversial story about a 1961–62 experiment by Yale social psychologist Stanley Milgram to investigate how people respond to orders given by people in authority. Much has been written about this experiment and details have been changed and embellished over the years, but the basic ethical issues it raises are still worth considering, even if the details of the actual study have been distorted.

The subjects were ordinary residents of New Haven who were asked to administer increasingly high levels of electric shocks to victims when they made errors in the tasks they were given. As the electric shocks got more and more severe, so did the apparent pain of the victims receiving them, to the extent that some appeared to be on the verge of dying. Not surprisingly, those administering the shocks became more and more disturbed by what they were being asked to do, but several continued, believing that they should do as their superiors told them. What they did not realize was that the so-called victims were, in fact, very convincing actors who were not being injured at all. Instead, the shock administrators were themselves the real subjects. It was their responses to authority that were being studied in this deceptive experiment.

This story raises several important ethical issues. First, this experiment reveals how power relationships can be used to control others. Second and equally important, this experiment relied on deception. The experimenters were, in fact, the subjects and the fake subjects colluded with the real scientists to deceive them. Without this deception the experiment would not have worked.

Is it acceptable to deceive subjects to this extent? What do you think?

Comment Yes, it is better to tell users in advance that they are being logged. As in the previous example, the users' knowledge that they are being logged often ceases to be an issue as they become involved in what they are doing.

11.3.6 Evaluate, interpret, and present the data

Choosing the evaluation paradigm and techniques to answer the questions that satisfy the evaluation goal is an important step. So is identifying the practical and ethical issues to be resolved. However, decisions are also needed about what data to collect, how to analyze it, and how to present the findings to the development team. To a great extent the technique used determines the type of data collected, but there are still some choices. For example, should the data be treated statistically? If qualitative data is collected, how should it be analyzed and represented? Some general questions also need to be asked (Preece et al., 1994): Is the technique reliable? Will the approach measure what is intended, i.e., what is its validity? Are biases creeping in that will distort the results? Are the results generalizable, i.e., what is their scope? Is the evaluation ecologically valid or is the fundamental nature of the process being changed by studying it?

Reliability

The reliability or consistency of a technique is how well it produces the ***same*** results on separate occasions under the ***same*** circumstances. Different evaluation processes have different degrees of reliability. For example, a carefully controlled experiment will have high reliability. Another evaluator or researcher who follows exactly the same procedure should get similar results. In contrast, an informal, unstructured interview will have low reliability: it would be difficult if not impossible to repeat exactly the same discussion.

Validity

Validity is concerned with whether the evaluation technique measures what it is supposed to measure. This encompasses both the technique itself and the way it is performed. If for example, the goal of an evaluation is to find out how users use a new product in their homes, then it is not appropriate to plan a laboratory experiment. An ethnographic study in users' homes would be more appropriate. If the goal is to find average performance times for completing a task, then counting only the number of user errors would be invalid.

Biases

Bias occurs when the results are distorted. For example, expert evaluators performing a heuristic evaluation may be much more sensitive to certain kinds of design flaws than others. Evaluators collecting observational data may consistently fail to notice certain types of behavior because they do not deem them important.

Put another way, they may selectively gather data that they think is important. Interviewers may unconsciously influence responses from interviewees by their tone of voice, their facial expressions, or the way questions are phrased, so it is important to be sensitive to the possibility of biases.

Scope

The scope of an evaluation study refers to how much its findings can be generalized. For example, some modeling techniques, like the keystroke model, have a narrow, precise scope. The model predicts expert, error-free behavior so, for example, the results cannot be used to describe novices learning to use the system.

Ecological validity

Ecological validity concerns how the environment in which an evaluation is conducted influences or even distorts the results. For example, laboratory experiments are strongly controlled and are quite different from workplace, home, or leisure environments. ***Laboratory*** experiments therefore have low ecological validity because the results are unlikely to represent what happens in the real world. In contrast, ethnographic studies do not impact the environment, so they have high ecological validity.

Ecological validity is also affected when participants are aware of being studied. This is sometimes called the ***Hawthorne effect*** after a series of experiments at the Western Electric Company's Hawthorne factory in the US in the 1920s and 1930s. The studies investigated changes in length of working day, heating, lighting, etc., but eventually it was discovered that the workers were reacting positively to being given special treatment rather than just to the experimental conditions.

11.4 Pilot studies

It is always worth testing plans for an evaluation by doing a pilot study before launching into the main study. **A** pilot study is a small trial run of the main study. The aim is to make sure that the plan is viable before embarking on the real study. For example, the equipment and instructions for its use can be checked. It is also an opportunity to practice interviewing skills, or to check that the questions in a questionnaire are clear or that an experimental procedure works properly. **A** pilot study will identify potential problems in advance so that they can be corrected. Sending out 500 questionnaires and then being told that two of the questions were very confusing wastes time, annoys participants, and is expensive.

Many evaluators run several pilot studies. As in iterative design, they get feedback, amend the procedure, and test it again until they know they have a good study. If it is difficult to find people to participate or if access to participants is limited, colleagues or peers can be asked to comment. Getting comments from peers is quick and inexpensive and can save a lot of trouble later. In theory, at least, there is no limit to the number of pilot studies that can be run, although there will be practical constraints.

Assignment

Find a journal or conference publication that describes an interesting evaluation study or select one using www.hcibib.org. Then use the DECIDE framework to determine which paradigms and techniques were used. Also consider how well it fared on ethical and practical issues.

(a) Which evaluation paradigms and techniques are used?

(b) Is triangulation used? How?

(c) Comment on the reliability, validity, ecological validity, biases and scope of the techniques described.

(d) Is there evidence of one or more pilot studies?

(e) What are the strengths and weakness of the study report? Write a 50–100 word critique that would help the author(s) improve their report.

Summary

This chapter has introduced four core evaluation paradigms and five categories of techniques and has shown how they relate to each other. The DECIDE framework identifies the main issues that need to be considered when planning an evaluation. It also introduces many of the basic concepts that will be revisited and built upon in the next three chapters: Chapter 12, which discusses observation techniques; Chapter 13, which examines techniques for gathering users' and experts' opinions; and Chapter 14, which discusses user testing and techniques for modeling users' task performance.

Key points

- An evaluation paradigm is an approach in which the methods used are influenced by particular theories and philosophies. Four evaluation paradigms were identified:
  1. "quick and dirty"
  2. usability testing
  3. field studies
  4. predictive evaluation
- Methods are combinations of techniques used to answer a question but in this book we often use the terms "methods" and "techniques" interchangeably. Five categories were identified:
  1. observing users
  2. asking users
  3. asking experts
  4. user testing
  5. modeling users' task performance
- The DECIDE framework has six parts:
  1. Determine the overall goals of the evaluation.
  2. Explore the questions that need to be answered to satisfy the goals.
  3. Choose the evaluation paradigm and techniques to answer the questions.
  4. Identify the practical issues that need to be considered.
  5. Decide on the ethical issues and how to ensure high ethical standards.
  6. Evaluate, interpret, and present the data.
- Drawing up a schedule for your evaluation study and doing one or several pilot studies will help to ensure that the study is well designed and likely to be successful.

Further reading

DENZIN, N. K. AND LINCOLN, Y. S. (1994) ***Handbook of Qualitative Research.*** London: Sage. This book is a collection of chapters by experts in qualitative research. It is an excellent reference source.

DIX, A., FINLAY, J., ABOWD, G. AND BEALE, R. (1998) ***Human-Computer Interaction*** (2d ed.). London: Prentice Hall Europe. This book provides a useful introduction to evaluation.

SHNEIDERMAN, B. (1998) ***Designing the User Interface: Strategies for Effective Human-Computer Interaction*** (3rd ed.). Reading, MA: Addison-Wesley. This text provides an alternative way of categorizing evaluation techniques and offers a good overview.

ROBSON, C. (1993) ***Real World Research.*** Oxford, UK: Blackwell. This book offers a practical introduction to applied research and evaluation. It is very readable.

WHITESIDE, J., BENNETT, J., AND HOLTZBLATT, K. (1998) Usability engineering: our experience and evolution. In M. Helander (ed.), ***Handbook of Human-Computer Interaction.*** Amsterdam: North Holland. This chapter reviews the strengths and weakness of usability engineering and explains why ethnographic techniques can provide a useful alternative in some circumstances, 791–817.

Chapter 12

Observing users

12.1 Introduction

Observation involves watching and listening to users. Observing users interacting with software, even casual observing, can tell you an enormous amount about what they do, the context in which they do it, how well technology supports them, and what other support is needed. In Chapter 9 we discussed the role of observation and ethnography in informing design, particularly early in the process. In this chapter we describe how to observe and do ethnography and discuss their role in evaluation.

Users can be observed in controlled laboratory-like conditions, as in usability testing, or in the natural environments in which the products are used—i.e., the field. How the observation is done depends on why it is being done and the approach adopted. There is a variety of structured, less structured, and descriptive observation techniques for evaluators to choose from. Which they select and how their findings are interpreted will depend upon the evaluation goals, the specific questions being addressed, and practical constraints. This chapter focuses on how to select appropriate observation techniques, how to do observation, and how to analyze the data and present findings from it. We also discuss the benefits and practicalities associated with each technique. An interview with interaction design consultant Sara Bly at the end of the chapter discusses how she uses observation in her work.

The main aims of this chapter are to:

- Discuss the benefits and challenges of different types of observation.
- Describe how to observe as an on-looker, a participant, and an ethnographer.
- Discuss how to collect, analyze and present data from observational evaluation.
- Examine key issues for doing think-aloud evaluation, diary studies and interaction logging.
- Give you experience in selecting and doing observational evaluation.

In general, observing and talking to users usually go together, but we leave the details of interview techniques until Chapter 13.

12.2 Goals, questions, and paradigms

Goals and questions provide a focus for observation, as the DECIDE framework points out. Even studies that use "quick and dirty" observations have a goal; for example, to identify or confirm usability and user experience goals in a prototype. *Goals and questions should guide all evaluation studies.* Just because some evaluators do not make their goals obvious does not mean that they don't have goals. Expert evaluators sometimes don't articulate their goals, but as you will read in Sara Bly's interview they do have them. Even in field studies and ethnography there is a careful balance between being guided by goals and being open to modifying, shaping, or refocusing the study as you learn about the situation. Being able to keep this balance is a skill that develops with experience.

ACTIVITY 12.1

(a) Find a small group of people who are using any kind of technology (e.g., computers, household or entertainment appliances, etc.) and try to answer the question, "What are these people doing?" Watch for three to five minutes and write down what you observe. When you have finished, note how you felt doing this.

(b) If you were to repeat the exercise what would you look for when you next observe the group? How would you refine your goals?

Comment

(a) What was the group doing? Were they talking, working, playing or something else? How were you able to decide? Did you feel awkward or embarrassed watching? Did you wonder whether you should tell them that you were observing them? What problems did you encounter doing this exercise? Was it hard to watch everything and remember what happened? What were the most important things? Did you wonder if you should be trying to identify and remember just those things? Was remembering the order of events tricky? Perhaps you naturally picked up a pen and paper and took notes. If so, was it difficult to record fast enough? How do you think the people being watched felt? Did they know they were being watched? Did knowing affect the way they behaved? Perhaps some of them objected and walked away. If you didn't tell them, do you think you should have?

(b) Your questions should be more focused. For example, you might ask, what are the people specifically trying to do and how is the technology being used? Is everyone in the group using the technology? Is it supporting or hindering the users' goals?

Having a goal, even a very general goal, helps to guide the observation because there is always so much going on.

12.2.1 What and when to observe

Observing is useful at any time during product development. Early in design, observation helps designers understand users' needs. Other types of observation are done later to examine whether the developing prototype meets users' needs.

Depending on the type of study, evaluators may be onlookers, participant observers, or ethnographers. Remember Christian Heath's and Paul Luff's ethnographic study of the London Underground discussed in Chapter 4 (Heath and Luff, 1992)? This study demonstrates the power of insightful observation to improve the redesign of a system. However, in order to understand how London Underground workers do their jobs the authors needed "insider" knowledge. The degree of immersion that evaluators adopt varies across a broad outsider-insider spectrum. Where a particular study falls along this spectrum depends on its goal and on the practical and ethical issues that constrain and shape it.

ACTIVITY 12.2 To understand this notion of an outsider-insider spectrum better, read the scenarios below and answer the questions that follow.

Scenario 1. A usability consultant joins a group who have been given WAP phones to test on a visit to Washington, DC. Not knowing the restaurants in the area, they use the WAP phone to find a list of restaurants within a five-mile radius of their hotel. Several are listed and while the group waits for a taxi, they find the telephone numbers of a couple, call them to ask about their menus, select one, make a booking, and head off to the restaurant. The usability consultant observes some problems keying instructions because the buttons seem small. She also notices that the screen seems rather small, but the person using it is able to get the information needed and call the restaurant, etc. Discussion with the group supports the evaluator's impression that there are problems with the interface, but on balance the device is useful and the group is pleased to get a table at a good restaurant nearby.

***Scenario* 2.** A usability consultant observes how participants perform a pre-planned task using the WAP phone in a usability laboratory. The task requires the participants to find the telephone number of a restaurant called Matisse. It takes them several minutes to do this and they appear to have problems. The video recording and interaction log suggest that the screen is too small for the amount of information they need to access and this is supported by participants' answers on a user satisfaction questionnaire.

(a) In which situation does the observer take the most control?

(b) What are the advantages and disadvantages of these two types of observation?

(c) When might each type of observation be useful?

Comment

(a) The observer takes most control in the second study. The task is predetermined, the participant is instructed what to do, and she is located in a controlled laboratory environment.

(b) The advantages of the field study are that the observer got to see how the device could be used in a real situation to solve a real problem. She experienced the delight expressed with the overall concept and the frustration with the interface. By watching how the group used the device "on the move," she gained an understanding of what they liked and needed. The disadvantage is that the observer was an "insider" in the group, so how objective could she be? The data is qualitative and while anecdotes can be very persuasive, how useful are they in evaluation? Maybe she was having such a good time that her judgment was clouded and she missed hearing negative comments and didn't notice some people's annoyance. Another study could be done to find out more, but it is not possible to replicate the exact situation, whereas the laboratory study is easier to replicate.

The advantages of the laboratory are that several users performed the same task, so different users' performance could be compared and averages calculated. The observer could also be more objective because she was more of an outsider. The disadvantage is that the study is artificial and says nothing about how the device would be used in the real environment.

(c) Both types of studies have merits. Which is better depends on the goals of the study. The laboratory study is useful for examining details of the interaction style to make sure that usability problems with the interface and button design are diagnosed and corrected. The field study reveals how the phone is used in a real world context and how it integrates with or changes users' behavior. Without this study, it is possible that developers **might** not have discovered the enthusiasm for the phone because the reward for doing laboratory tasks is not as compelling as a good meal!

Table 12.1 Type of observation

Observation	**Controlled environment (i.e., lab-like)**	**Field environment (i.e., natural)**
Outsider looking on	"Quick and dirty" In usability testing	"Quick and dirty" In field studies
Insider	(Not applicable)	Participant observation (e.g., in ethnography)

Table 12.1 summarizes this insider-outsider discussion, how it relates to different types of environments, and how much control evaluators take over the evaluation process.

12.2.2 Approaches to observation

Observers can be outsiders in the field and in the controlled environments, but they can't be insiders in a controlled environment. In the field it is possible to have varying degrees of "insider-outsiderness." In practice these distinctions are more difficult to describe than to experience!

"Quick and dirty" observation

"Quick and dirty" observations can occur anywhere, anytime. For example, evaluators often go into a school, home, or office to watch and talk to users in a casual way to get immediate feedback about a prototype or product. Evaluators can also join a group for a short time, which gives them a slightly more insider role. Quick and dirty observations are just that, ways of finding out what is happening quickly and with little formality.

Observation in usability testing

Video and interaction logs capture everything that the user does during a usability test including keystrokes, mouse clicks, and their conversations. In addition, observers can watch through a one-way mirror or via a remote TV screen. The observational data is used to see and analyze what users do and how long they spend on different aspects of the task. It also provides insights into users' affective reactions. For example, sighs, tense shoulders, frowns, and scowls speak of users' dissatisfaction and frustrations. The environment is controlled but users often forget that they are being observed. In addition, many evaluators also supplement findings from the laboratory with observations in the field.

Observation in field studies

In field studies, as we have said, observers may be anywhere along the outsider-insider spectrum. Looking on as an outsider, being a participant observer, or being an ethnographer brings a philosophy and practices that influence what data is collected, how data collection is done, and how the data is analyzed and reported. Colin Robson (1993) summarizes the possible levels of participation as: complete participants, more marginal participants, observers who also participate, and people who observe from the outside and do not participate.

Whether and in what ways observers influence those being observed depends on the type of observation and the observer's skills. The goal is to cause as little disruption as possible. An example of outsider observation is when an observer is interested only in the presence of certain types of behavior. For instance, in a study of the time spent by boys and girls using technology in the classroom, an observer may go into the classroom to note when technology is used by boys and when by girls. She could do this by standing at the back of the room with a data sheet on which she notes the gender of the children who use the computer and how long they spend using it. In contrast, if the goal is to understand how the computer integrates with other artifacts and social interactions in the classroom, a more holistic approach would be better. In this situation the evaluator might take more of an insider perspective in which she talks to participants as well as observes. The observer mixes and integrates with participants more, but there is no illusion that she is anything other than an observer.

Inside observers may be participant observers or ethnographers. In participant observation evaluators participate with users in order to learn what they do and how and why they do it. A fully participant observer observes from the inside as a member of the group, which means she must not only be present to share experiences, but also learn the social conventions of the group, including beliefs and protocols, dress codes, communication conventions, use of language, and non-verbal communication. "Participant observation combines participation in the lives of the people under study with maintenance of a professional distance that allows adequate observation and recording of data" (Fetterman, 1998, p. 34–35).

Ethnographers can be thought of as participant observers or not, depending on your point of view. Ethnographers themselves debate this issue. Some see participant observation as virtually synonymous with ethnography (Atkinson and Hammersley, 1994). Others view participant observation as a technique that is used in ethnography along with informants from the community, interviews with community members, and the study of community artifacts (Fetterman, 1998). Ethnographic evaluation is derived from ethnography. Ethnographic studies typically take weeks, months, or even longer to gain an "inside" understanding of what is going on in a community. Much shorter studies are usual in interaction design because of the time constraints imposed by development schedules.

As in any evaluation study, goals and questions determine whether the observation will be "quick and dirty," in a controlled environment or in the field, and the extent to which the observers are outsiders or insiders. Determining goals, exploring questions, and choosing techniques are necessary steps in the DECIDE framework. Practical and ethical issues also have to be identified and decisions made about how to handle them.

12.3 How to observe

The same basic data-collection tools are used for laboratory and field studies (i.e., direct observation, taking notes, collecting video, etc.) but the way in which they are used is different. In the laboratory the emphasis is on the details of what individuals do, while in the field the context is important and the focus is on how people interact with each other, the technology, and their environment. Furthermore, the equipment in the laboratory is usually set up in advance and is relatively static, whereas in the field it usually must be moved around. In this section we discuss how to observe, and then examine the practicalities and compare data-collection tools.

12.3.1 In controlled environments

The role of the observer is to first collect and then make sense of the stream of data on video, audiotapes, or notes made while watching users in a controlled environment. Many practical issues have to be thought about in advance, including the following.

- It is necessary to decide where users will be located so that the equipment can be set up. Many usability laboratories, for example, have two or three wall-mounted, adjustable cameras to record users' activities while they work on test tasks. One camera might record facial expressions, another might focus on mouse and keyboard activity, and another might record a broad view of the participant and capture body language. The stream of data from the cameras is fed into a video editing and analysis suite where it is annotated and partially edited. Another form of data that can be collected is an interaction log. This records all the user's key presses. Mobile usability laboratories, as the name suggests, are intended to be moved around, but the equipment can be bulky. Usually it is taken to a customer's site where a temporary laboratory environment is created.
- The equipment needs testing to make sure that it is set up and works as expected, e.g., it is advisable that the audio is set at the right level to record the user's voice.
- An informed consent form should be available for users to read and sign at the beginning of the study. A script is also needed to guide how users are greeted, and to tell them the goals of the study, how long it will last, and to explain their rights. It is also important to make users feel comfortable and at ease.

Whether in a real or make-do laboratory one of the problems with this type of observation is that the observer doesn't know what users are thinking, and can only guess from what she sees.

Think-aloud technique Imagine observing someone who has been asked to evaluate the interface of the web search engine Northernlight. The user, who has used the web only once before, is told to find a list of the books written by the well-known biologist Stephen Jay Gould. He is told to type http://www.northernlight.com and then proceed however he thinks best. He types the URL and gets a screen similar to the one in Figure 12.1.

Next he goes to the search box but types Stephen Jay Gouild without realizing that he has made a typing error and added an 'i'. He presses return and gets a screen similar to the one in Figure 12.2.

He is silent. What is going on, you wonder? What is he thinking? One way around this problem is to collect a think-aloud protocol, using a technique developed by Erikson and Simon for examining people's problem-solving strategies (Erikson and Simon, 1985). The technique requires people to say out loud everything that they are thinking and trying to do, so that their thought processes are externalized.

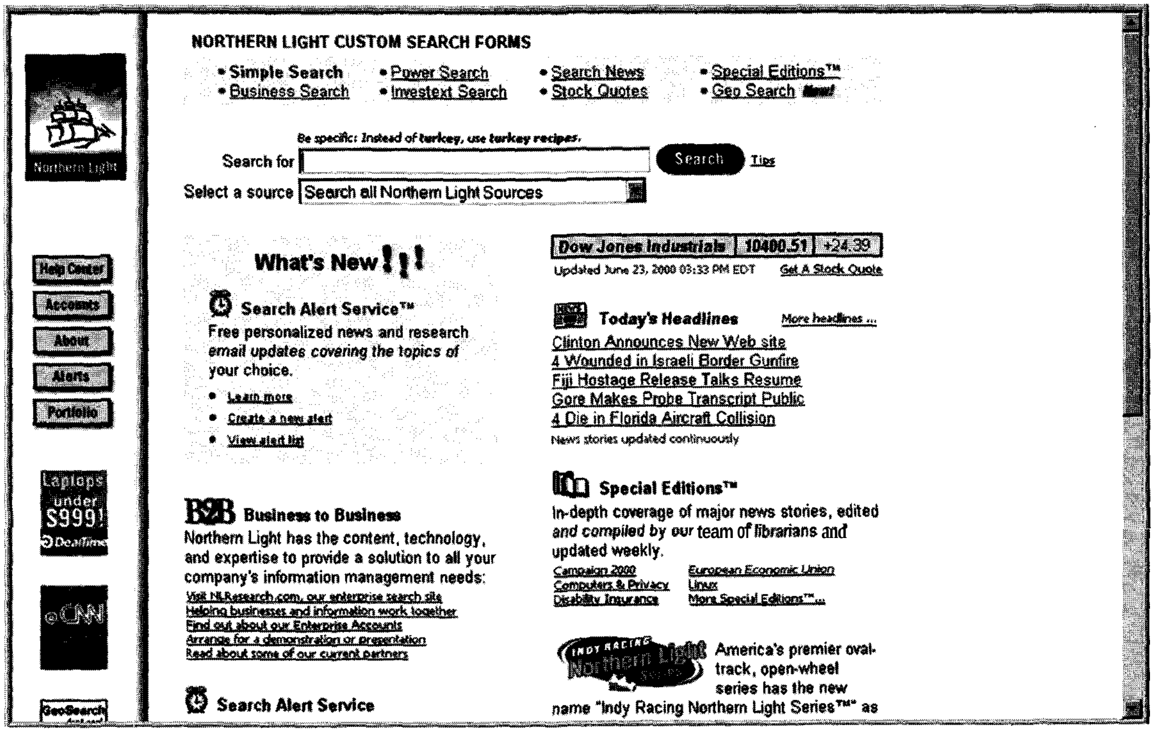


Figure 12.1 Home page of Northernlight search engine (www.northernlight.com).

So, let's imagine an action replay of the situation just described, but this time the user has been instructed to think aloud:

> *I'm typing in http://www.northernlight.comas you told me.* (types)
> *Now I press the enter key, right?* (presses enter key)
> (pause and silence)
> *It's taking a few moments to respond.*
> *Oh! Here it is.* (Figure 12.1 appears)
> *Gosh, there's a lot of stuff on this screen, hmmm, I wonder what I do next.* (pauses and looks at the screen) *Probably a simple search. What's apower search and there's all these others too?*
> *I just want to find Stephen Jay Gould, right, and then it's bound to have a list of his books?* (pause) *Well, it looks like I should type his name in this box here.* (moves cursor towards the search box. Positions cursor. Types 'Stephen Jay Gouild'. Pauses, but does not notice that he has incorrectly included an "i" in Gould, then clicks the search button.) *Well, something seems to be happening...* (Watches) *something is happening. Ah! What's this...* (Looks at screen and Figure 12.2 appears)
> *Silence...*

Now you know more about what the user is trying to achieve but he is silent again. You can see that he has spelled Gould incorrectly and that he doesn't realize that he has typed Gouild. What you don't know is what he is thinking now or what is he

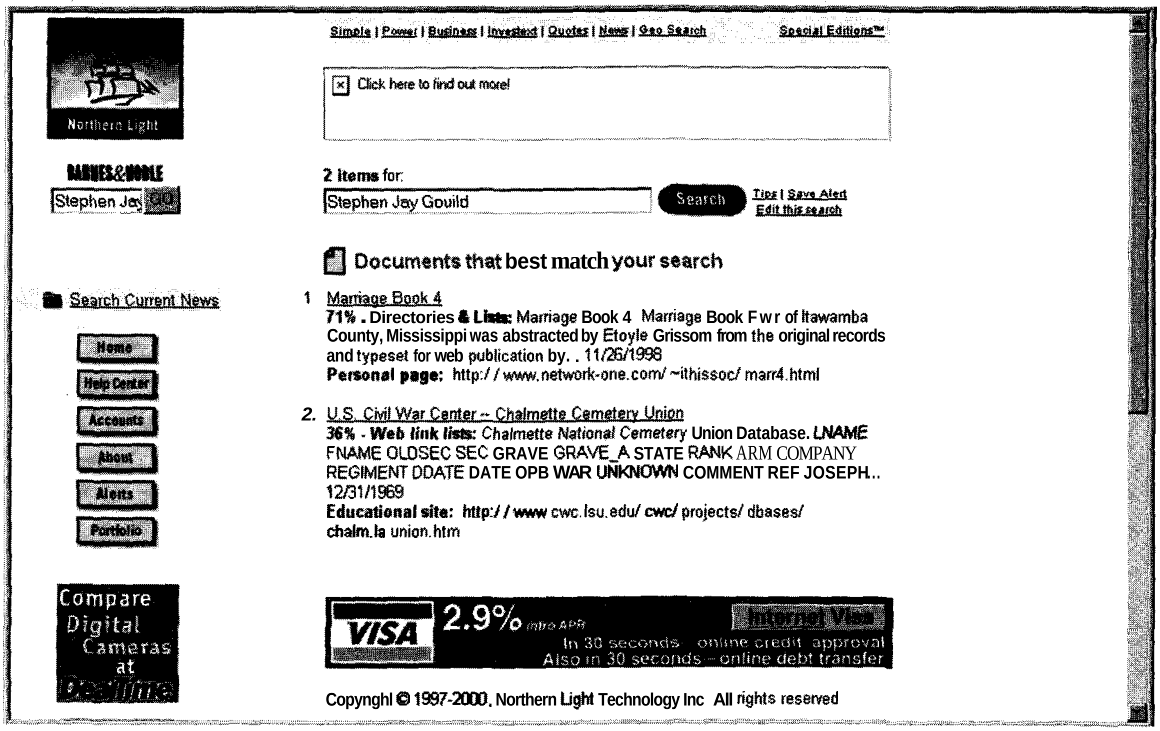


Figure 12.2 The screen that appears in response to searching for Stephen Jay Gouild.

looking at. Has he noticed his typing error or the Barnes and Noble box at the top left that says "Stephen Jay"?

ACTIVITY 12.3 Try a think-aloud exercise yourself. Go to an e-commerce website, such as Amazon.com or BarnesandNoble.com, and look for something that you want to buy. Think aloud as you search and notice how you feel and behave. Did you find it difficult to keep speaking all the way through the task? Did you feel awkward? Did you stop when you got stuck?

Comment You probably felt self-conscious and awkward doing this. Some people say they feel really embarrassed. At times you may also have started to forget to speak out loud because it feels like talking to yourself, which most of us don't do. You may also have found it difficult to think aloud when the task got difficult. In fact, you probably stopped speaking when the task became demanding, and that is exactly the time when an evaluator is most eager to hear your comments.

The occurrence of these silences is one of the biggest problems with the think-aloud technique.

If a user is silent during a think-aloud protocol, the evaluator could interrupt and remind him to think out loud, but that would be intrusive. Another solution is to have two people work together so that they talk to each other. Working with another person is often more natural and revealing because they talk in order to help each other along. This technique has been found particularly successful with children. It is also very effective when evaluating systems intended to be used synchronously by groups of users, e.g., shared whiteboards.

12.3.2 In the field

Whether the observer sets out to be an outsider or an insider, events in the field can be complex and rapidly changing. There is a lot for evaluators to think about, so many experts have a framework to structure and focus their observation. The framework can be quite simple. For example, this is a practitioner's framework that focuses on just three easy-to-remember items to look for:

- ***The person.*** Who is using the technology at any particular time?
- ***The place.*** Where are they using it?
- ***The thing.*** What are they doing with it?

Frameworks like the one above help observers to keep their goals and questions in sight. Experienced observers may, however, prefer more detailed frameworks, such as the one suggested by Goetz and LeCompte (1984) below, which encourages observers to pay greater attention to the context of events, the people and the technology:

- ***Who*** is present? How would you characterize them? What is their role?
- ***What*** is happening? What are people doing and saying and how are they behaving? Does any of this behavior appear routine? What is their tone and body language?
- ***When*** does the activity occur? How is it related to other activities?
- ***Where*** is it happening? Do physical conditions play a role?
- ***Why*** is it happening? What precipitated the event or interaction? Do people have different perspectives?
- How is the activity organized? What rules or norms influence behavior?

Colin Robson (1993) suggests a slightly longer but similar set of items:

- ***Space.*** What is the physical space like and how is it laid out?
- ***Actors.*** What are the names and relevant details of the people involved?
- ***Activities.*** What are the actors doing and why?
- ***Objects.*** What physical objects are present, such as furniture?
- ***Acts.*** What are specific individuals doing?
- ***Events.*** Is what you observe part of a special event?
- ***Goals.*** What are the actors trying to accomplish?
- ***Feelings.*** What is the mood of the group and of individuals?

ACTIVITY 12.4

(a) Look at Goetz's and LeCompte's framework. Apart from there being more items than in the first framework, what is the other main difference?

(b) Now compare this framework with Robson's. What does Robson's attend to that is not obvious in Goetz's and LeCompte's framework?

(c) Which of the three frameworks do you think would be easiest to remember and why?

Comment

(a) The Goetz and LeCompte framework pays much more attention to the context of the observation.

(b) There is considerable overlap between the two frameworks despite differences in wording. The main difference is that Robson's framework pays attention to the mood of the group.

(c) The three-item framework is likely to be easy, but so is the Goetz and LeCompte framework because it adopts the much used organizing principle "who, what, when, where, why, how." Robson's framework has two extra items and no obvious way of remembering them. However, having said that, to me it is more explicit. Which is used for a particular study depends on the study goals and how much detail is needed, and to a degree, it is also a matter of personal preference.

These frameworks are useful not only for providing focus but also for organizing the observation and data-collection activity. Below is a checklist of things to plan before going into the field:

- State the initial study goal and questions clearly.
- Select a framework to guide your activity in the field.

  Decide how to record events—i.e., as notes, on audio, or on video, or using a combination of all three. Make sure you have the appropriate equipment and that it works. You need a suitable notebook and pens. A laptop computer might be useful but could be cumbersome. Although this is called observation, photographs, video, interview transcripts and the like will help to explain what you see and are useful for reporting the story to others.
- Be prepared to go through your notes and other records as soon as possible after each evaluation session to flesh out detail and check ambiguities with other observers or with the people being observed. This should be done routinely because human memory is unreliable. A basic rule is to do it within 24 hours, but sooner is better!
- As you make and review your notes, try to highlight and separate personal opinion from what happens. Also clearly note anything you want to go back to. Data collection and analysis go hand in hand to a large extent in fieldwork.
- Be prepared to refocus your study as you analyze and reflect upon what you see. Having observed for a while, you will start to identify interesting

phenomena that seem relevant. Gradually you will sharpen your ideas into questions that guide further observation, either with the same group or with a new but similar group.

- Think about how you will gain the acceptance and trust of those you observe. Adopting a similar style of dress and finding out what interests the group and showing enthusiasm for what they do will help. Allow time to develop relationships. Fixing regular times and venues to meet is also helpful, so everyone knows what to expect. Also, be aware that it will be easier to relate to some people than others, and it will be tempting to pay attention to those who receive you well, so make sure you attend to everyone in the group.
- Think about how to handle sensitive issues, such as negotiating where you can go. For example, imagine you are observing the usability of a portable home communication device. Observing in the living room, study, and kitchen is likely to be acceptable, but bedrooms and bathrooms are probably out of bounds. Take time to check what participants are comfortable with and be accommodating and flexible. Your choice of equipment for data collection will also influence how intrusive you are in people's lives.
- Consider working as a team. This can have several benefits; for instance, you can compare your observations. Alternatively, you can agree to focus on different people or different parts of the context. Working as a team is also likely to generate more reliable data because you can compare notes among different evaluators.

  Consider checking your notes with an informant or members of the group to ensure that you are understanding what is happening and that you are making good interpretations.
- Plan to look at the situation from different perspectives. For example, you may focus on particular activities or people. If the situation has a hierarchical structure, as in many companies, you will get different perspectives from different layers of management—e.g., end-users, marketing, product developers, product managers, etc.

12.3.3 Participant observation and ethnography

Being a participant observer or an ethnographer involves all the practical steps just mentioned, but especially that the evaluator must be accepted into the group. An interesting example of participant observation is provided by Nancy Baym's work (1997) in which she joined an online community interested in soap operas for over a year in order to understand how the community functioned. She told the community what she was doing and offered to share her findings with them. This honest approach gained her their trust, and they offered support and helpful comments. As Baym participated she learned about the community, who the key characters were, how people interacted, their values, and the types of discussions that were generated. She kept all the messages as data to be referred to later. She also adapted interviewing and questionnaire techniques to collect additional information. She summarizes her data collection as follows (Baym, 1997, p. 104):

> ***The data for this study were obtained from three sources. In October 1991, I saved all the messages that appeared. . . . I collected more messages in 1993. Eighteen participants responded to a questionnaire Iposted. . . . Personal email correspondence with 10 other . . . participants provided further information. I posted two notices to the group explaining the project and offeringto exclude posts by those who preferred not to be involved. No one declined to participate.***

Using this data, Baym examined the group's technical and participatory structure, its emergent traditions, and its usage with the technology. As the work evolved, she shared its progress with the group members, who were supportive and helpful.

ACTIVITY 12.5 Drawing on your experience of using email, bulletin boards, UseNet News, or chat rooms, how might participant observation online differ from face-to-face participant observation?

Comment In online participant observation you don't have to look people in the eye, deal with their skepticism, or wonder what they think of you, as you do in face-to-face situations. What you wear, how you look, or the tone of your voice don't matter. However, what you say or don't say and how you say it are central to the way others will respond to you. Online you only see part of people's context. You usually can't see how they behave off line, how they present themselves, their body language, how they spend their day, their personalities, who is present but not participating, etc.

As we said the distinction between ethnography and participant observation is blurred. Some ethnographers believe that ethnography is an open interpretivist approach in which evaluators keep an open mind about what they will see. Others, such as David Fetterman from Stanford University, see a stronger role for a theoretical underpinning: "before asking the first question in the field the ethnographer begins with a problem, a theory or model, a research design, specific data collection techniques, tools for analysis, and a specific writing style" (Fetterman, 1998, p. 1). This may sound as if ethnographers have biases, but by making assumptions explicit and moving between different perspectives, biases are at least reduced. Ethnographic study allows ***multiple*** interpretations of reality; it is ***interpretivist.*** Data collection and analysis often occur simultaneously in ethnography, with analysis happening at many different levels throughout the study. The question being investigated is refined as more understanding about the situation is gained.

The checklist below (Fetterman, 1998) for doing ethnography is similar to the general list just mentioned:

- Identify a problem or goal and then ask good questions to be answered by the study, which may or may not invoke theory depending on your philosophy of ethnography. The observation framework such as those mentioned above can help to focus the study and stimulate questions.

- The most important part of fieldwork is just being there to observe, ask questions, and record what is seen and heard. You need to be aware of people's feelings and sensitive to where you should not go.
- Collect a variety of data, if possible, such as notes, still pictures, audio and video, and artifacts as appropriate. Interviews are one of the most important data-gathering techniques and can be structured, semi-structured, or open. So-called ***retrospective interviews*** are used after the fact to check that interpretations are correct.
- As you work in the field, be prepared to move backwards and forwards between the broad picture and specific questions. Look at the situation holistically and then from the perspectives of different stakeholder groups and participants. Early questions are likely to be broad, but as you get to know the situation ask more specific questions.
- Analyze the data using a ***holistic*** approach in which observations are understood within the broad context—i.e., they are *contextualized.* To do this, first synthesize your notes, which is best done at the end of each day, and then check with someone from the community that you have described the situation accurately. Analysis is usually iterative, building on ideas with each pass.

ACTIVITY 12.6 Look at the steps listed for doing ethnography and compare them with the earlier generic set for field observation (see Section 12.3.2). What is the main difference?

Comment Both sets of steps involve structuring observations and refining goals and questions through knowledge gained during the study. Both use similar data collection techniques and rely on the trust and cooperation of those being observed. Ethnographers tend to be deeply immersed in the group, whereas not everyone doing field studies takes this approach. Some ethnographers, such as David Fetterman, are guided by theory; others are strongly against this and believe that ethnography should be approached open-mindedly.

During the last ten years ethnography has gained credibility in interaction design because if products are to be used in a wide variety of environments designers must know the context and ecology of those environments (Nardi and O'Day, 1999). However, for those unfamiliar with ethnography and general field observation there are two dilemmas. The first dilemma is, "When have I observed

DILEMMA When Should I Stop Observing?

Knowing when to stop doing *any* type of evaluation can be difficult for novice evaluators, but it is particularly tricky in observational studies and ethnography because there is no obvious ending. Schedules often dictate when your study ends. Otherwise, stop when you stop learning new things. Two indications of having done enough are when you start to see similar patterns of behavior being repeated, or when you have listened to all the main stakeholder groups and understand their perspectives.

DILEMMA How Can I Adapt Ethnography to Fit the Development Process?

Many developers are unsure how to integrate ethnographic evaluation into development cycles. In addition, most developers have a technical training that does not encourage them to value qualitative data. We discussed the use of ethnography to inform design in Chapter 9. Here is an example where it has been adapted for evaluation.

In a project for the Department of Juvenile Justice, Ann Rose and her colleagues developed a procedure to be used by technical design teams with limited ethnographic training (Rose et al., 1995). This applied form of ethnography acknowledges the comparatively small amounts of time available for any kind of user study. By making the process more structured the amount of time needed for the study can be reduced. It also emphasizes that taking time to become familiar with the intricacies of a system enhances the evaluator's credibility during the field study and promotes productive fieldwork. The procedures this group advocates are highly structured, and while they may seem contrary to ethnographic practice, this structure helps to make it possible for some development teams to benefit from an applied ethnographic approach. There are four stages, as follows:

1 Preparation

Understand organization policies and work culture.

Familiarize yourself with the system and its history.

Set initial goals and prepare questions.

Gain access and permission to observe and interview.

2 Field study

Establish a rapport with managers and users.

Observe and interview users in their workplace and collect data.

Follow any leads that emerge from the visits.

Record your visits.

3 Analysis

Compile the collected data in numerical, textual, and multimedia databases.

Quantify data and compile statistics.

Reduce and interpret the data.

Refine the goals and processes used.

4 Reporting

Consider multiple audiences and goals.

Prepare a report and present the findings.

enough?" The second dilemma is, "How can I adapt ethnography so that it better fits the short development cycles and the mindset of the developers?"

ACTIVITY 12.7 What are the main differences between the stages that Rose et al. (1995) describe and the steps suggested by Fetterman (1998)?

Comment The list in the "How Can I Adapt Ethnography" dilemma suggests that the evaluators are not as immersed in the study as Fetterman's process suggests. One aim of the Rose procedure is radically to reduce the time needed to do a study so that it is compatible with system development. Another aim is to reduce the data to a quantifiable form so that it is familiar and acceptable to the developers.

12.4 Data collection

Data collection techniques (i.e., taking notes, audio recording, and video recording) are used individually or in combination and are often supplemented with photos from a still camera. When different kinds of data are collected, evaluators have to coordinate them; this requires additional effort but has the advantage of providing more information and different perspectives. Interaction logging and participant diary studies are also used, as we discuss later in Section 12.5. Which techniques are used will depend on the context, time available, and the sensitivity of what is being observed. In most settings, audio, photos, and notes will be sufficient. In others it is essential to collect video data so as to observe in detail the intricacies of what is going on.

12.4.1 Notes plus still camera

Taking notes is the least technical way of collecting data, but it can be difficult and tiring to write and observe at the same time. Observers also get bored and the speed at which they write is limited. Working with another person solves some of these problems and provides another perspective. Handwritten notes are flexible in the field but must be transcribed. However, this transcription can be the first step in data analysis, as the evaluator must go through the data and organize it. A laptop computer can be a useful alternative but it is more obtrusive and cumbersome, and its batteries need recharging every few hours. If a record of images is needed, photographs, digital images, or sketches are easily collected.

12.4.2 Audio recording plus still camera

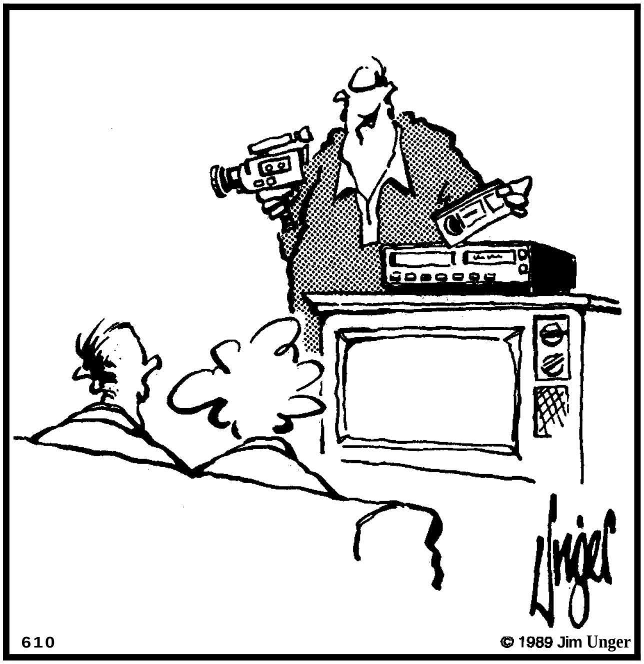
Audio can be a useful alternative to note taking and is less intrusive than video. It allows evaluators to be more mobile than with even the lightest, battery-driven video cameras, and so is very flexible. Tapes, batteries, and the recorder are now relatively inexpensive but there are two main problems with audio recording. One is the lack of a visual record, although this can be dealt with by carrying a small camera. The second drawback is transcribing the data, which can be onerous if the contents of many hours of recording have to be transcribed; often, however, only sections are needed. Using a headset with foot control makes transcribing less onerous. Many studies do not need this level of detail; instead, evaluators use the recording to remind them about important details and as a source of anecdotes for reports.

12.4.3 Video

Video has the advantage of capturing both visual and audio data but can be intrusive. However, the small, handheld, battery-driven digicams are fairly mobile, inexpensive and are commonly used.

A problem with using video is that attention becomes focused on what is seen through the lens. It is easy to miss other things going on outside of the camera view. When recording in noisy conditions, e.g., in rooms with many computers running or outside when it is windy, the sound may get muffled.

Analysis of video data can be very time-consuming as there is so much to take note of. Over 100 hours of analysis time for one hour of video recording is common for detailed analyses in which every gesture and utterance is analyzed. However, this



"This is a video of you two watching the video of our vacation."

level of detail is usually not needed because evaluators often focus on particular episodes and use the whole recording only for contextual information and reference.

In Table 12.2 we summarize the key features, advantages and drawbacks of these three combinations of data collection techniques.

ACTIVITY 12.8 Imagine you are a consultant who is employed to help develop a new computerized garden-planning tool to be used by amateur and professional garden designers. Your goal is to find out how garden designers use an early prototype as they walk around their clients' gardens sketching design ideas, taking notes, and asking the clients about what they like and how they and their families use the garden. What are the advantages and disadvantages of the three types of data-collection techniques in this environment?

Comment Handwritten notes do not require specialist equipment. They are unobtrusive and very flexible but difficult to do while walking around a garden. If it starts to rain there is no equipment to get wet, but taking notes is tiring, people lose concentration, biases creep in, and handwriting can be difficult to decipher. Video captures more information (e.g., the landscape, where the designers are looking, sketches, comments, etc.) but it is more intrusive, you must also carry equipment and film and what happens if it starts to rain? You also need access to

Table 12.2 Comparison of the three main data-collection techniques used in observation

Criterion	Notes *plus* camera	Audio *plus* camera	Video
Equipment	Paper, pencil and camera are easily available.	Inexpensive, handheld recorder with a good microphone. Headset useful for easy transcription.	More expensive. Editing, mixing and analysis equipment needed.
Flexibility of use	Very flexible. Unobtrusive.	Flexible. Relatively unobtrusive.	Needs positioning and focusing camera lens. Even portable versions can be bulky.
Completeness of data	Only get what note-taker thinks is important and can record in the time available. Problem with inexperienced evaluators.	Can obtain complete audio recording but visual data is missing. Notes, photographs, sketches can augment recording but need coordinating with the recording.	Most complete method of data collecting, especially if more than one camera used, but coordination of video material is needed.
Disturbance to users	Very low.	Low but cassette must be changed and microphone positioned.	Can be very obtrusive. Care needed to avoid Hawthorne effect.
Reliability of data	May be low. Relies on humans making a good record and knowing what to record.	High but external noise, e.g. fans in computers can muffle what is said.	Can be high but depends on what camera is focused on.
Analysis	Relatively easy to transcribe. Rich descriptions can be produced. Transcribing data can be onerous or a useful first step in data analysis.	Critical discussions can be identified. Transcription needed for detailed analysis. Permanent original record that can be revisited.	Critical incidents can be identified and tagged. Automated support needed for detailed analysis. Permanent original record that can be revisited.
Feedback to design team	Relies strongly on the authority of the evaluator.	Material captured on tape is more convincing than notes but feedback relies on authority of evaluator.	Hard to dispute material captured on video. Video clips are very powerful for communicating ideas.

playback and editing facilities. Audio could be a good compromise, but integrating sketches and other artifacts later can be a burden and garden planning is a highly visual, aesthetic activity. You could also supplement notes and audio with a still camera.

12.5 Indirect observation: tracking users' activities

Sometimes direct observation is not possible because it is obtrusive or evaluators cannot be present over the duration of the study, and so users' activities are tracked indirectly. Diaries and interaction logs are two techniques for doing this. From the records collected evaluators reconstruct what happened and look for usability and user experience problems.

12.5.1 Diaries

Diaries provide a record of what users did, when they did it, and what they thought about their interactions with the technology. They are useful when users are scattered and unreachable in person, as in many Internet and web evaluations. Diaries are inexpensive, require no special equipment or expertise, and are suitable. for long-term studies. Templates can also be created online to standardize entry format and enable the data to go straight into a database for analysis. These templates are like those used in open-ended online questionnaires. However, diary studies rely on participants being reliable and remembering to complete them, so incentives are needed and the process has to be straightforward and quick. Another problem is that participants often remember events as being better or worse than they really were, or taking more or less time than they actually did.

Robinson and Godbey (1997) asked participants in their study to record how much time Americans spent on various activities. These diaries were completed at the end of each day and the data was later analyzed to investigate the impact of television on people's lives. In another diary study, Barry Brown and his colleagues from Hewlett Packard collected diaries form 22 people to examine when, how, and why they capture different types of information, such as notes, marks on paper, scenes, sounds, moving images, etc. (Brown, et al., 2000). The participants were each given a small handheld camera and told to take a picture every time they captured information in any form. The study lasted for seven days and the pictures were used as memory joggers in a subsequent semi-structured interview used to get participants to elaborate on their activities. Three hundred and eighty-one activities were recorded. The pictures provided useful contextual information. From this data the evaluators constructed a framework to inform the design of new digital cameras and handheld scanners.

12.5.2 Interaction logging

Interaction logging in which key presses, mouse or other device movements are recorded has been used in usability testing for many years. Collecting this data is usually synchronized with video and audio logs to help evaluators analyze users' behavior and understand how users worked on the tasks they set. Specialist software tools are used to collect and analyze the data. The log is also time-stamped so it can be used to calculate how long a user spends on a particular task or lingered in a certain part of a website or software application.

Explicit counters that record visits to a website were once a familiar sight. Recording the number of visitors to a site can be used to justify maintenance and upgrades to it. For example, if you want to find out whether adding a bulletin board to an e-commerce website increases the number of visits, being able to compare traffic before and after the addition of the bulletin board is useful. You can also track how long people stayed at the site, which areas they visited, where they came from, and where they went next by tracking their Internet Service Provider (I.S.P.) address. For example, in a study of an interactive art museum by researchers at the University of Southern California, server logs were analyzed by tracking visitors in this way (McLaughlin et al., 1999). Records of when people came to the site, what they requested, how long they looked at each page, what browser they were using, and what country they were from, etc., were collected over a seven-month period. The data was analyzed using Webtrends, a commercial analysis tool, and the evaluators discovered that the site was busiest on weekday evenings. In another study that investigated lurking behavior in listserver discussion groups, the number of messages posted was compared with list membership over a three-month period to see how lurking behavior differed among groups (Nonnecke and Preece, 2000).

An advantage of logging user activity is that it is unobtrusive, but this also raises ethical concerns that need careful consideration (see the dilemma about observing without being seen). Another advantage is that large volumes of data can be logged automatically. However, powerful tools are needed to explore and analyze this data quantitatively and qualitatively. An increasing number of visualization tools are being developed for this purpose; one example is WebLog, which dynamically shows visits to websites, as illustrated in Figure 12.3 (Hochheiser and Shneiderman, 2000).

DILEMMA They Don't Know We Are Watching. Shall We Tell Them?

If you have appropriate algorithms and sufficient computer storage, large quantities of data about Internet usage can be collected and users need never know. Furthermore, if we tell users that we are logging their behavior they may react or change their behavior. So, what should we do? It depends on the context, how much personal information is collected, and how the information will be used. Many companies now tell you that your computer activity and phone calls may be logged for quality assurance and other purposes. Most people do not object to this practice. However, should we be concerned about logging personal information (e.g., discussions about health or financial information)? Should users be worried? How can we exploit the ability to log user behavior when visiting websites without overstepping a person's civil rights? Where should we draw the line?

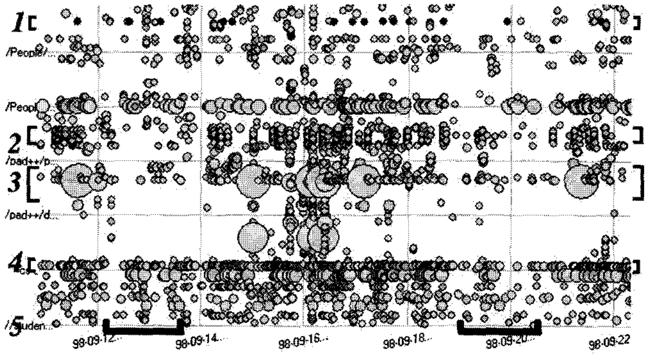


Figure 12.3 A display from WebLog, time vs. URL (Hochheiser and Shneiderman, 2001). The requested URL is on the y-axis, with the date and time on the x-axis. The dark lines on the x-axis correspond to weekends. Each circle represents a request for a single page, and the size of the circle indicates the number of bytes delivered for a given request. (Color, which is not shown here, indicates the Http status response.)

12.6 Analyzing, interpreting, and presenting the data

By now you should know that many, indeed most observational evaluations generate a lot of data in the form of notes, sketches, photographs, audio and video records of interviews and events, various artifacts, diaries, and logs. Most observational data is qualitative and analysis often involves interpreting what users were doing or saying by looking for patterns in the data. Sometimes qualitative data is categorized so that it can be quantified and in some studies events are counted.

Dealing with large volumes of data, such as several hours of video, is daunting, which is why it is particularly important to plan observation studies very carefully before starting them. The DECIDE framework suggests identifying goals and questions first before selecting techniques for the study, because the goals and questions help determine which data is collected and how it will be analyzed.

When analyzing any kind of data, the first thing to do is to "eyeball" the data to see what stands out. Are there patterns or significant events? Is there obvious evidence that appears to answer a question or support a theory? Then proceed to analyze it according to the goals and questions. The discussion that follows focuses on three types of data:

- *Qualitative data* that is *interpreted* and used to tell "the story" about what was observed.
- *Qualitative data* that is *categorized* using techniques such as content analysis.
- *Quantitative data* that is collected from interaction and video logs and presented as values, tables, charts and graphs and is treated statistically.

12.6.1 Qualitative analysis to tell a story

Much of the power of analyzing descriptive data lies in being able to tell a convincing story, illustrated with powerful examples that help to confirm the main points and will be credible to the development team. It is hard to argue with well-chosen video excerpts of users interacting with technology or anecdotes from transcripts.

In the interview with Sara Bly you will read about how she and her colleagues use data from several sources. At the end of each observation period they review their data, discuss what they observed, and construct a story from the data. This story evolves as more data is collected and more insights are generated. Teamwork plays an important role in this process because it provides different perspectives that can be compared. A large part of the analysis involves making "collections" of incidents or anecdotes that illustrate similar issues. For example, if several people comment at different times that it is hard to track down a manager in a particular work setting, these examples are powerful evidence of the need for better communication.

To summarize, the main activities involved in working with qualitative data to tell a story are:

- Review the data after each observation session to synthesize and identify key themes and make collections.
- Record the themes in a coherent yet flexible form, with examples. While post-its enable you to move ideas around and group similar ones, they can fall off and get lost and are not easily transported, so capture the main points in another form, either on paper or on a laptop, or make an audio recording.
- Record the date and time of each data analysis session. (The raw data should already be systematically logged with dates.)
- As themes emerge, you may want to check your understanding with the people you observe or your informants.
- Iterate this process until you are sure that your story faithfully represents what you observed and that you have illustrated it with appropriate examples from the data.
- Report your findings to the development team, preferably in an oral presentation as well as in a written report. Reports vary in form, but it is always helpful to have a clear, concise overview of the main findings presented at the beginning.

Analyzing and reporting ethnographic dafa Ethnographers work in a similar way but emphasize understanding events within the context in which they happen. Data is collected from participant observation, interviews, and artifacts, and analysis is continuous with great attention to detail. Ethnographers reconstruct knowledge to produce detailed descriptions known as ***rich*** or ***thick descriptions.*** In these descriptions, quotes, pictures, and anecdotes play a convincing role in communicating the findings to others. The main activities in analyzing ethnographic data are similar to those just mentioned but notice the emphasis on detail (Fetterman, 1998):

- Look for key events within a group that speak about what drives the group's activity.
- Look for patterns of behavior in various situations and among different players. With experience, ethnographers build up sets of knowledge from various sources, asking questions, listening, probing, comparing and contrasting, synthesizing, and evaluating information.
- Compare sources of data against each other to provide consistent explanations.

  Finally, report your findings in a convincing and honest way. Writing is part of the analysis since it helps to crystallize ideas.

Software tools, such as NUDIST and Ethnograph, allow ethnographers to code their notes and artifact descriptions so that they can be sorted, searched, and retrieved. For example, using NUDIST, field notes can be searched for key words or phrases and a report printed listing every occasion the word or phrase is used. The information can also be printed out as a tree showing the relationship of occurrences. Similarly, NUDIST can be used to search a body of text to identify specific predetermined categories or words for content analysis. The more copious the notes, the more useful tools like NUDIST are. Furthermore, many exploratory searches can be done to test hypotheses among different categories of data.

Other computerized tools support basic statistical analysis. For example, some data can be analyzed using statistical tests (such as chi-square contingency table analysis or rank correlation) to determine whether particular trends are significant.

12.6.2 Qualitative analysis for categorization

Data from think-aloud protocols, video, or audio transcripts can be analyzed in different ways. These can be coarse-grained or detailed analyses of excerpts from a protocol in which each word, phrase, utterance, or gesture is analyzed. Sometimes examining the comment or action in the context of other behavior is sufficient. In this section we discuss a selection of techniques. Some are used more often in research while others are used more for product development.

Looking for incidents or patterns

Analyzing even a short half-hour videotape would be very time-consuming if evaluators studied every comment or action in detail. Furthermore, such fine-grained analyses are often not necessary. A common strategy is to look for critical incidents, such as times when users were obviously stuck. Such incidents are usually marked by a comment, silence, looks of puzzlement, etc. Evaluators focus on these incidents and review them in detail, using the rest of the video as context to inform their analysis. For example, Jurgen Koenemann-Belliveau et al. (1994) used this approach to compare the efficacy of two versions of a Smalltalk programming manual for supporting novice programmers. They used a form of critical incident analysis to examine breakdowns or problems in achieving a programming task and also to identify possible threats of incidents. This enabled them to identify specific problems that might otherwise have been overlooked. Taking this approach, they were able to trace through a sequence of incidents and achieve a more holistic understanding of the problem. For example they found that they needed to emphasize how objects interact in teaching object-oriented programming.

Theory may also be used to guide the study. Wendy Mackay et al. (2000) took this approach in analyzing a four-minute excerpt from a video of users working with a new software tool. Using Activity Theory to guide their analysis, they identified 19 shifts in attention between different parts of the tool interface and the task at hand. (In fact, some users spent so much time engaged in these shifts that they lost track of their original task.) Using the theory helped the evaluators to focus on relevant incidents.

Whether your analysis is coarse-grained or finer, whether you are guided by theory or are just looking for incidents and patterns of behavior, you need a way of handling your data and recording your analysis. For example, in another part of their study, Wendy Mackay et al. (2000) collected and analyzed video excerpts of users interacting with their tool and constructed a form of paper storyboards. The series of images taken from the video illustrated the changes made through the task, while the accompanying text descriptions provided details about the precise operations performed and the difficulties encountered.

A variety of tools are available to record, manipulate and search the data. NUDIST was mentioned above and Box 12.1 briefly describes the Observer Video-Pro tool. Typically reports from these analyses are fed back to the development team, often accompanied by video clips.

BOX 12.1 The Observer Video-Pro: An Automated Data Analysis Tool

The Observer Video-Pro provides the following features (Noldus, 2000):

- During preparation of a video tape recording, a *time code generator* adds an invisible time code to each video frame.
- During a data-collection session, a *time code reader* retrieves the time code from the tape, allowing frame-accurate event timing independent of the playback speed of the video cassette recorder (VCR).
- Each keyboard entry is firmly anchored to the video frame displayed at the instant the evaluator presses the first key of a behavior code or free-format note. The evaluator can also use a mouse to score events.
- Observational data can be reviewed and edited, with synchronized display of the corresponding video images.
- For optimal visual feedback during coding, the evaluator can display the video image in a window on the computer screen.
- The VCR can be controlled by the computer, allowing software-controlled "jog", "shuttle", and "search" functions.
- Video images can be captured and saved as disk files for use as illustrations in documents, slides for presentations, etc.
- Marked video episodes can be copied to an Edit Decision List for easy creation of highlight tapes.

ACTIVITY 12.9 What does the Observer Video-Pro tool allow you to search for in the data collected?

Comment Depending on how the logs have been annotated, using the Observer Video-Pro product, you can search the data for various things including the following:

Video time—A specific time, e.g., 02:24:36.04 (hh:mm: ss.dd).

Marker—A previously entered free-format annotation.

Event—A combination of actor, behavior, and modifiers, with optional wildcards (e.g., the first occurrence of "glazed look" or "Sarah approaches Janice").

Text—Any word or alphanumeric text string occurring in the coded event records or free-format notes.

Analyzing data into categories

Content analysis provides another fine grain way of analyzing video data. It is a systematic, reliable way of coding content into a meaningful set of mutually exclusive categories (Williams et al., 1988). The content categories are determined by the evaluation questions and one of its most challenging aspects is determining meaningful categories that are orthogonal—i.e., do not overlap each other in any way.

Deciding on the appropriate granularity is another issue to be addressed. The content categories must also be reliable so that the analysis can be replicated. This can be demonstrated by training a second person to use the categories. When training is complete, both researchers analyze the same data sample. If there is a large discrepancy between the two analyses, either training was inadequate or the categorization is not working and needs to be refined. By talking to the researchers you can determine the source of the problem, which is usually with the categorization. If so, then a better categorization scheme needs to be devised and re-tested by doing more inter-researcher reliability tests. However, if the researchers do not seem to know how to carry out the process then they probably need more training.

When a high level of reliability is reached, it can be quantified by calculating an ***inter-research reliability rating.*** This is the percentage of agreement between the two researchers, defined as the number of items that both categorized in the same way expressed as a percentage of the total number of items examined. It provides a measure of the efficacy of the technique and the categories.

Content analysis ***per se*** is not used very often in evaluations because it is very labor-intensive and time-consuming but a study by Maria Ebling and Bonnie John (2000) showed how useful it can be. They developed a hierarchical content classification for analyzing data when evaluating a graphical interface for a distributed file system.

Analyzing discourse

Another approach to video, and audio analysis is to focus on the dialog, i.e., the meaning of what is said, rather than the content. Discourse analysis is strongly interpretive, pays great attention to context, and views language not only as reflecting psychological and social aspects but also as constructing it (Coyle, 1995). An underlying assumption of discourse analysis is that there is no objective scientific truth. Language is a form of social reality that is open to interpretation from different perspectives. In this sense, the underlying philosophy of discourse analysis is similar to that of ethnography. Language is viewed as a constructive tool and discourse analysis provides a way of focusing upon how people use language to construct versions of their worlds (Fiske, **1994).**

Small changes in wording can change meaning, as the following excerpts indicate (Coyle, **1995):**

> ***Discourse analysis is what you do when you are saying that you are doing discourse analysis. . . .***
>
> ***According to Coyle, discourse analysis is what you do when you are saying that you are doing discourse analysis. . . .***

By adding just three words "According to Coyle," the sense of authority changes, depending on what the reader knows about Coyle's work and reputation. Some analysts also suggest that a useful approach is to look for variability either within or between individuals.

Analyzing discourse on the Internet (e.g., in chatrooms, bulletin boards, and virtual worlds) has started to influence designers' understanding about users' needs in these environments. Conversation analysis is a very fine-grained form of discourse analysis that can be used for this purpose. In conversational analysis the semantics of the discourse are examined in fine detail. The focus is on how conversations are conducted. This technique is used in sociological studies and examines how conversations start, how turntaking is structured, and other rules of conversation. It can also be very useful when comparing conversations that take place during video-mediated sessions or in computer-mediated communication such as chatrooms as discussed in Chapter **4.**

12.6.3 Quantitative data analysis

Video data collected in usability laboratories is usually annotated as it is observed. Small teams of evaluators watch monitors showing what is being recorded in a control room out of the users' sight. As they see errors or unusual behavior, one of the evaluators marks the video and records a brief remark. When the test is finished evaluators can use the annotated recording to calculate performance times so they can compared users' performance on different prototypes. The data stream from the interaction log is used in a similar way to calculate performance times. Typically this data is further analyzed using simple statistics such as means, standard deviations, T-tests, etc. Categorized data may also be quantified and analyzed statistically, as we have said.

12.6.4 Feeding the findings back into design

The results from an evaluation can be reported to the design team in several ways, as we have indicated. Clearly written reports with an overview at the beginning and detailed content list make for easy reading and a good reference document. Includ-

ing anecdotes, quotations, pictures, and video clips helps to bring the study to life, stimulate interest, and make the written description more meaningful. Some teams like quantitative data, but its value depends on the type of study and its goals. Verbal presentations that include video clips can also be very powerful. Often both qualitative and quantitative data analysis are useful becuase they provide alternative perspectives.

Assignment

The aim of this assignment is for you to learn to do field obsewation. To do the assignment you will need to find a group of people or a single individual engaged in using one of the following: a mobile phone, a VCR, a photocopying machine, computer software, or some other type of technology that interests you. Assume that you have been employed to improve the product, either by doing a redesign or by creating a completely new product. You can observe people in your family, your friends, or people in your class or local community group.

For this assignment you should:

(a) Consider what the basic goal of "improving the product" means. What initial questions might you ask?

(b) Watch the group (or person) casually to get an understanding of issues that might create challenges for you doing this assignment and information that might enable you to refine your questions.

(c) Then plan your study:

- (i) Think again about what questions will help direct your observation. What are you evaluating?
- (ii) Decide where on the outsider-insider spectrum of observers you wish to be.
- (iii) Prepare an informed consent form and any scripts that you need to introduce yourself and your study.
- (iv) Decide how you will collect data and prepare any data-collection sheets needed; acquire and test any equipment needed.
- (v) Decide how you will analyze the data that you collect.
- (vi) Think through the DECIDE framework. Is everything covered?
- (vii) If so, do a pilot study to check your preparation.

(d) Carry out your study but limit its scope. For example, plan two half-hour observation periods.

(e) Now analyze your data using the method chosen above.

(f) Write a report about what you did and why; describe your data, how you analyzed it, and your findings.

(g) Suggest some ways in which the product might be improved.

Summary

Observing users in the field enables designers to see how technology is used in context. It is valuable for confirming designers' understanding of users' needs and for exploring new design ideas. Various amounts of control, intervention, and involvement with users are possible.

At one end of the spectrum, laboratory studies offer a strongly controlled environment with little evaluator involvement; at the other, participant observation and ethnography require deeper involvement with users and understanding of context. Diaries and data-logging techniques provide a way of tracking user activity without intruding.

Key points

- Observation in usability testing tends to be objective, from the outside. The observer watches and analyzes what happens.
- In contrast, in participant observation the evaluator works with users to understand their activities, beliefs and feelings within the context in which the technology is used.
- Ethnography uses a set of techniques that include participant observation and interviews. Ethnographers immerse themselves in the culture that they study.
- The way that observational data is collected and analyzed depends on the paradigm in which it is used: quick and dirty, user testing, or field studies.
- Combinations of video, audio and paper records, data logging, and diaries can be used to collect observation data.
- In participant observation, collections of comments, incidents, and artifacts are made during the observation period. Evaluators are advised to discuss and summarize their findings as soon after the observation session as possible.

  Analyzing video and data logs can be difficult because of the sheer volume of data. It is important to have clearly specified questions to guide the process and also access to appropriate tools.
- Evaluators often flag events in real time and return to examine them in more detail later. Identifying key events is an effective approach. Fine-grained analyses can be very time-consuming.

Fuither reading

BLY, S. (1997) Field work: Is it product work? ***Interactions,*** January and February, 25–30. This article provides additional information to supplement the interview with Sara Bly. It gives a broad perspective on the role of participant observation in product development.

BOGDEWIC, S. P. (1992) Participant observation. In B. F. Crabtree and W. L. Miller (eds.), ***Doing Qualitative Research.*** Newbury Park, CA: Sage, 45-69. This chapter provides an introduction to participant observation.

BROWN, B. A., SELLEN, A. J., AND O'HARA, K. P. (2000). ***A diary study of information capture in working life.*** In the Proceedings of CHI2000, The Hague, Holland, 438–445. This paper discusses how cameras were used in a diary study, followed by semi-structured interviews, to inform the design of handheld storage devices.

FETTERMAN, D. M. (1998). ***Ethnography: Step by Step*** (2nd ed.). (Vol. 17). Thousand Oaks, CA: SAGE. This book provides an introduction to the theory and practice of ethnography and is an excellent guide for beginners. In addition, it has a useful section on computerized tools for ethnography.

ROBSON, C. (1993). ***Real World Research.*** Oxford, ***UK:*** Blackwell. Chapter 8 discusses a range of observation methods. There is a section on doing participant observation and also on observing from the outside using coding schemes.

INTERVIEW with Sara Bly

Sara Bly is a user-centered design consultant who specializes in the design and evaluation of distributed group technologies and practices. As well as having a Ph.D. in computer science, Sara pioneers the development of rich, qualitative observational techniques for analyzing group interactions and activities that inform technology design. Prior to becoming a consultant, Sara managed the Collaborative Systems Group at Xerox Palo Alto Research Center (PARC). While at PARC, Sara also contributed to ground-breaking work on shared drawing, awareness systems, and systems that used non-speech audio to represent information, and to the PARC Media Space project, in which video, audio, and computing technologies are uniquely combined to create a trans-geographical laboratory.

JP: Sara, tell us about your work and what especially interests you.

SB: I'm interested in the ways that qualitative studies, particularly based on ethnographic methods, can inform design and development of technologies. My work spans the full gamut of user-centered design, from early conceptual design through iterative prototypes to final product deployment. I've worked on a wide range of projects from complex collaborative systems to straightforward desktop applications, and a variety of new technologies. My recent projects include a cell phone enhancement, a web-based video application, and the integration of text-based virtual environments with documents.

JP: Why do you think qualitative methods are so important for evaluating usability?

SB: I strongly believe that technical systems are closely bound with the social setting in which they are used. An important part of evaluation is to look "beyond the task." Too often we think of computer systems in isolation from the rest of the activities in which the people are involved. It's important to be able to see the interface in the context of ongoing practice. Usually the complexities and "messiness" of everyday life do not lend themselves to constraining the evaluation to only a few variables for testing. Qualitative methods are particularly helpful for evaluating complex systems that involve several tasks, embedded in other activities that include multiple users.

JP: Can you give me an example?

SB: Recently I was asked to design and evaluate an application for setting up personal preferences and purchasing services on the web. I was told it would be hard to test the interface "in the field" because it was difficult to get a 45–60 minute test period when the user wasn't being interrupted. When I pointed out that interruptions were normal in the environment in which the product would be used and therefore should occur in the evaluation too, the client looked aghast. There was a moment of silence as he realized, for the first time, that this hadn't been taken into account in the design and that the interface timed out after 60 seconds. It was unusable because the user would have to start all over again after each timeout. This should have been noticed at the requirements stage. So why wasn't it? It sounds like such an obvious thing, but the team was so busy with the intricacies of the design that they failed to realize what the real world would be like in which the system would be used. This might sound extreme, but you'd be surprised how often such things happen.

JP: Collaborative applications seem particularly difficult to evaluate out of context.

SB: Yes, you have to evaluate collaborative systems integrated within an organizational culture in which working relationships are taken into account. We know that work practice impacts system design and that the introduction of a new system impacts work practice. Consequently, the system and the practice have to evolve together. Understanding the task or the interface is impossible without understanding the environment in which the system will be used.

JP: Much of what you've described involves various forms of observation. How do you collect and analyze this data?

SB: It's important that qualitative methods are not seen as just *watching*. Any method we use has at least three critical phases. First, there is the initial assessment of the domain and/or technology and the determination of the questions to address in the evaluation. Second is the data collection, analysis, and representation, and third, the communication of the findings with the development team. I try to start with a clear understanding of what I need to focus on in the field. However, I also try hard not to start with assumptions about what will be true. So, 1 start with a ***well-defined*** focus but ***not*** a hypothesis. In the field (or even in the lab), I primarily use interviews and observations with some self-reporting that often takes the form of diaries, etc. The data typically consists of my notes, the audio and/or videotapes from interviews and observation time, still pictures, and as many artifacts as I can appropriately gather (e.g., a work document covered with post-its, a page from an old calendar). I also prefer to work with at least one other colleague so that there is a minimum of two perspectives on the events and data.

JP: It sounds like keeping track of all this data could be a problem. How do you organize and analyze it?

SB: Obviously it's critical not to end with the data collection. Whenever possible, I do immediate debriefs after each session in the field with my colleague, noting individually and collectively whatever jumped out at us. Subsequently, I use the interview notes (from everyone involved) and the tapes and artifacts to construct as much of a picture of what happened as possible, without putting any judgment on it. For example, in a recent study six of us were involved in interviews and observations. We worked in pairs and tried to vary the pairings as often as possible. Thus. we had lots of conversations about the data and the situations before we ever came together. First, we wrote up the notes from each session (something I try to do as soon as possible). Next we got together and began looking across the data. That is, we created representations of important events (tables, maps, charts) together. Because we collectively had observed all the events and because we could draw upon our notes, we could feed the data from each observation into each finding. Oftentimes, we create collections, looking for common behaviors or events across multiple sessions. A collection will highlight activities that are crucial to the design of the system being evaluated. Whatever techniques we use, we always come back to the data as a reality and validity check.

JP: Is it difficult to get development teams and managers to listen to you? How do you feed your findings back?

SB: As often as possible, development teams are involved in the process along the way. They participate in setting the initial goals of the evaluation, occasionally in observation sessions, and as recipients of a final report. My goal with any project is to ensure that the final report is not a handoff but rather an interaction that offers a chance to work together on what we've found.

JP: What are the main challenges you face?

SB: It's always difficult to conduct a field study with as much time and participation as would be ideal. Most product cycles are short and the evaluation is just one of many necessary steps. So it's always a challenge to do an evaluation that is timely, useful, and yet based on solid methodology.

A gnawing question for me is how to evaluate a system in the context of the customer's own environment and experience when the system is not fully developed and ready to deploy? If we can't bring a product to the field, can we bring the field to the product? For example, a client recently had a prototype interface for a system that was intended to provide a new approach to person-to-person calls. But using the interface made sense only in the context of actual real-world interactions. So, while we certainly could do a standard usability study of the interface, this approach wouldn't get at the questions of how well the product would fit into an actual work situation.

JP: Finally, what about the future? Any comments?

SB: I think the explosion of computing technology is both exciting and overwhelming. We now have so much new information constantly available and so many new devices to master that it's hard to keep up. Evaluation is going to become ever more critical and complex and we should use all the techniques at our disposal as appropriate. I think an increasingly important aspect of new interfaces will be not only how well they support performance, satisfaction, and experience, but the way in which a user is able to grasp a conceptual model that is compatible with, but does not overwhelm their ongoing practice.

Chapter 13

Asking users and experts

13.1 Introduction

In the last chapter we looked at observing users. Another way of finding out what users do, what they want to do, like, or don't like is to ask them. Interviews and questionnaires are well-established techniques in social science research, market research, and human-computer interaction. They are used in "quick and dirty" evaluation, in usability testing, and in field studies to ask about facts, behavior, beliefs, and attitudes. Interviews and questionnaires can be structured (as in the HutchWorld case study in Chapter 10), or flexible and more like a discussion, as in field studies. Often interviews and observation go together in field studies, but in this chapter we focus specifically on interviewing techniques.

The first part of this chapter discusses interviews and questionnaires. As with observation, these techniques can be used in the requirements activity (as we described in Chapter 7), but in this chapter we focus on their use in evaluation. Another way of finding out how well a system is designed is by asking experts for their opinions. In the second part of the chapter, we look at the techniques of heuristic evaluation and cognitive walkthrough. These methods involve predicting how usable interfaces are (or are not). As in the previous chapter, we draw on the DECIDE framework from Chapter 11 to help structure studies that use these techniques.

The main aims of this chapter are to:

- Discuss when it is appropriate to use different types of interviews and questionnaires.
- Teach you the basics of questionnaire design.
- Describe how to do interviews, heuristic evaluation, and walkthroughs.
- Describe how to collect, analyze, and present data collected by the techniques mentioned above.
- Enable you to discuss the strengths and limitations of the techniques and select appropriate ones for your own use.

13.2 Asking users: interviews

Interviews can be thought of as a "conversation with a purpose" (Kahn and Cannell, 1957). How like an ordinary conversation the interview is depends on the questions to be answered and the type of interview method used. There are four main types of interviews: *open-ended or unstructured, structured, semi-structured,* and *group* interviews (Fontana and Frey, 1994). The first three types are named according to how much control the interviewer imposes on the conversation by following a *predetermined set of questions.* The fourth involves a small group guided by an interviewer who facilitates discussion of a specified set of topics.

The most appropriate approach to interviewing depends on the evaluation goals, the questions to be addressed, and the paradigm adopted. For example, if the goal is to gain first impressions about how users react to a new design idea, such as an interactive sign, then an informal, open-ended interview is often the best approach. But if the goal is to get feedback about a particular design feature, such as the layout of a new web browser, then a structured interview or questionnaire is often better. This is because the goals and questions are more specific in the latter case.

13.2.1 Developing questions and planning an interview

When developing interview questions, plan to keep them short, straightforward and avoid asking too many. Here are some guidelines (Robson, 1993):

- Avoid long questions because they are difficult to remember.
- Avoid compound sentences by splitting them into two separate questions. For example, instead of, "How do you like this cell phone compared with

previous ones that you have owned?" Say, "How do you like this cell phone? Have you owned other cell phones? If so, How did you like it?" This is easier for the interviewee and easier for the interviewer to record.

- Avoid using jargon and language that the interviewee may not understand but would be too embarrassed to admit.
- Avoid leading questions such as, "Why do you like this style of interaction?" If used on its own, this question assumes that the person did like it.
- Be alert to unconscious biases. Be sensitive to your own biases and strive for neutrality in your questions.

Asking colleagues to review the questions and running a pilot study will help to identify problems in advance and gain practice in interviewing.

When planning an interview, think about interviewees who may be reticent to answer questions or who are in a hurry. They are doing *you* a favor, so try to make it as pleasant for them as possible and try to make the interviewee feel comfortable. Including the following steps will help you to achieve this (Robson, 1993):

1. An ***Introduction*** in which the interviewer introduces himself and explains why he is doing the interview, reassures **interviewees about the ethical issues**, and asks if they mind being recorded, if appropriate. This should be exactly the same for each interviewee.
2. A *warmup* session where easy, non-threatening questions come first. These may include questions about demographic information, such as "Where do you live?"
3. A ***main*** session in which the questions are presented in a logical sequence, with the more difficult ones at the end.
4. A ***cool-off period*** consisting of a few easy questions (to defuse tension if it has arisen).
5. A ***closing*** session in which the interviewer thanks the interviewee and switches off the recorder or puts her notebook away, signaling that the interview has ended.

The golden rule is to be professional. Here is some further advice about conducting interviews (Robson, 1993):

- Dress in a similar way to the interviewees if possible. If in doubt, dress neatly and avoid standing out.
- Prepare an informed consent form and ask the interviewee to sign it.
- If you are recording the interview, which is advisable, make sure your equipment works in advance and you know how to use it.
- Record answers exactly; do not make cosmetic adjustments, correct, or change answers in any way.

13.2.2 Unstructured interviews

Open-ended or unstructured interviews are at one end of a spectrum of how much control the interviewer has on the process. They are more like conversations that focus on a particular topic and may often go into considerable depth. Questions posed by the interviewer are open, meaning that the format and content of answers is not predetermined. The interviewee is free to answer as fully or as briefly as she wishes. Both interviewer and interviewee can steer the interview. Thus one of the skills necessary for this type of interviewing is to make sure that answers to relevant questions are obtained. It is therefore advisable to be organized and have a plan of the main things to be covered. Going in without an agenda to accomplish a goal is not advisable, and should not to be confused with being open to new information and ideas.

A benefit of unstructured interviews is that they generate rich data. Interviewees often mention things that the interviewer may not have considered and can be further explored. But this benefit often comes at a cost. A lot of unstructured data is generated, which can be very time-consuming and difficult to analyze. It is also impossible to replicate the process, since each interview takes on its own format. Typically in evaluation, there is no attempt to analyze these interviews in detail. Instead, the evaluator makes notes or records the session and then goes back later to note the main issues of interest.

The main points to remember when conducting an unstructured interview are:

- Make sure you have an interview agenda that supports the study goals and questions (identified through the DECIDE framework).
- Be prepared to follow new lines of enquiry that contribute to your agenda.
- Pay attention to ethical issues, particularly the need to get informed consent.
- Work on gaining acceptance and putting the interviewees at ease. For example, dress as they do and take the time to learn about their world.
- Respond with sympathy if appropriate, but be careful not to put ideas into the heads of respondents.
- Always indicate to the interviewee the beginning and end of the interview session.
- Start to order and analyze your data as soon as possible after the interview.

ACTIVITY 13.1 Ananova is a virtual news reporter created by the British Press Association on the website www.ananova.com, which is similar to the picture in Figure 13.1. Viewers who wish to hear Ananova report the news must select from the menu beneath her picture and must have downloaded software that enables them to receive streaming video. Those who wish to read text may do so.

The idea is that Ananova is a life-like, i.e., an 'anthropomorphic' news presenter. She is designed to speak, move her lips, and blink, and she has some human facial expressions. She reads news edited from news reports. Ananova's face, her voice tone, her hair, in fact everything about her was tested with users before the site was launched so that she would appeal to as many users as possible. She is fashionable and looks as though she is in her twenties or

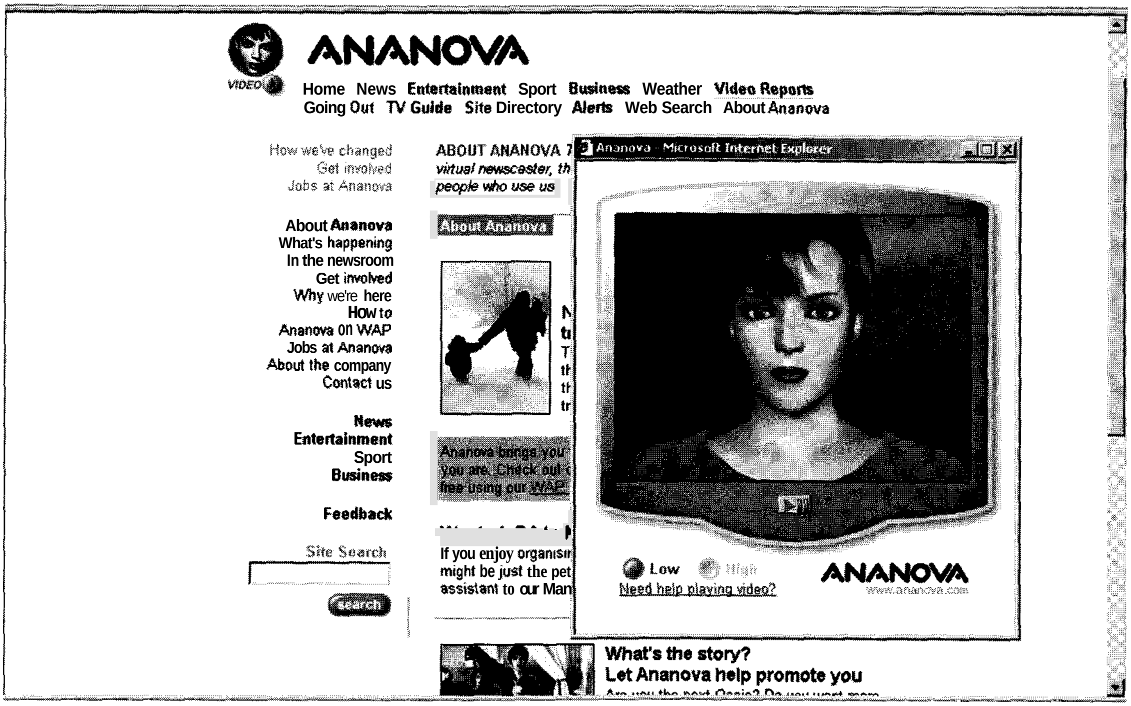


Figure 13.1 Ananova.com showing Ananova, a virtual news presenter.

early thirties — presumably the age that market researchers determined fits the profile of the majority of users — and she is also designed to appeal to older people too.

To see Ananova in action, go to the website (www.annanova.com) and follow the directions for downloading the software. Alternatively you can do the activity by just looking at the figure and thinking about the questions.

(a) Suggest unstructured interview questions that seek opinions about whether Ananova improves the quality of the news service.

(b) Suggest ways of collecting the interview data.

(c) Identify practical and ethical issues that need to be considered.

Comment

(a) Possible questions include: Do you think Ananova reading the news is good? Is it better than having to read it yourself from a news bulletin? In what ways does having Ananova read the news influence your satisfaction with the service?

(b) Taking notes might be cumbersome and distracting to the interviewee, and it would be easy to miss important points. An alternative is to audio record the session. Video recording is not needed as it isn't necessary to see the interviewee. However, it would be useful to have a camera at hand to take shots of the interface in case the interviewee wanted to refer to aspects of Ananova.

(c) The obvious practical issues are obtaining a cassette recorder, finding participants, scheduling times for the interviews and finding a quiet place to conduct them. Having a computer available for the interviewee to refer to is important. The ethical issues include telling the interviewees why you are doing the interviews and what you will do with the information, and guaranteeing them anonymity. An informed consent form may be needed.

13.2.3 Structured interviews

Structured interviews pose predetermined questions similar to those in a questionnaire (see Section 13.3). Structured interviews are useful when the study's goals are clearly understood and specific questions can be identified. To work best, the questions need to be short and clearly worded., Responses may involve selecting from a set of options that are read aloud or presented on paper. The questions should be refined by asking another evaluator to review them and by running a small pilot study. Typically the questions are ***closed,*** which means that they require a precise answer. The same questions are used with each participant so the study is standardized.

13.2.4 Semi-structured interviews

Semi-structured interviews combine features of structured and unstructured interviews and use both closed and open questions. For consistency the interviewer has a basic script for guidance, so that the same topics are covered with each interviewee. The interviewer starts with preplanned questions and then probes the interviewee to say more until no new relevant information is forthcoming. For example:

> Which websites do you visit most frequently? ***<Answer>*** Why? ***<Answer mentions several but stresses that she prefers hottestmusic.com>*** And why do you like it? ***<Answer>*** Tell me more about x? ***<silence, followed by an answer>*** Anything else? ***<Answer>*** Thanks. Are there any other reasons that you haven't mentioned?

It is important not to preempt an answer by phrasing a question to suggest that a particular answer is expected. For example, "You seemed to like this use of color . . ." assumes that this is the case and will probably encourage the interviewee to answer that this is true so as not to offend the interviewer. Children are particularly prone to behave in this way. The body language of the interviewer, for example, whether she is smiling, scowling, looking disapproving, etc., can have a strong influence.

Also the interviewer needs to accommodate ***silences*** and not to move on too quickly. Give the person time to speak. ***Probes*** are a device for getting more information, especially neutral probes such as, "Do you want to tell me anything else?" You may also ***prompt*** the person to help her along. For example, if the interviewee is talking about a computer interface but has forgotten the name of a key menu item, you might want to remind her so that the interview can proceed productively. However, semi-structured interviews are intended to be broadly replicable, so probing and prompting should aim to help the interview along without introducing bias.

ACTIVITY 13.2 Write a semi-structured interview script to evaluate whether receiving news from Ananova is appealing and whether Ananova's presentation is realistic. Show two of your peers the

Ananova.com website or Figure 13.1. Then ask them to comment on your interview script. Refine the questions based on their comments.

Comment

You can use questions that have a predetermined set of answer choices. These work well for fast interviews when the range of answers is known, as in the airport studies where people tend to be in a rush. Alternatively, open-ended questions can also be used if you want to explore the range of opinions.

Some questions that you might ask include:

- Have you seen Ananova before?
- Would you like to receive news from Ananova?
- Why?
- In your opinion, does Ananova look like a real person?

Some of the questions in Exercise 13.2 have a predetermined range of answers, such as "yes," "no," "maybe." Others, such as the one about interviewees' attitudes, do not have an easily predicted range of responses. But it would help us in collecting answers if we list possible responses together with boxes that can just be checked (i.e., ticked). Here's how we could convert the questions from Activity 13.2.

- Have you seen Ananova before? (Explore previous knowledge)
  Interviewer checks box ☐ *Yes* ☐ *No* ☐ *Don't remember/know*
- Would you like to receive news from Ananova? (Explore initial reaction, then explore the response)
  Interviewer checks box ☐ *Yes* ☐ *No* ☐ *Don't know*
- Why?
  If response is "Yes" or "No," interviewer says, "Which of the following statements represents your feelings best?"
  For "Yes," Interviewer checks the box
  - ☐ *I don't like typing*
  - ☐ *This is fun/cool*
  - ☐ *I've never seen a system like this before*
  - ☐ *It's going to be the way of the future*
  - ☐ *Another reason (Interviewer notes the reason)*

  For "No," Interviewer checks the box
  - ☐ *I don't like speech systems*
  - ☐ *I don't like systems that pretend to be people*
  - ☐ *It's faster to read*
  - ☐ *I can't control the pace of presentation*
  - ☐ *I can't be bothered to download the software*
  - ☐ *Another reason (Interviewer notes the reason)*
- In your opinion, does Ananova look like a real person?
  Interviewer checks box
  - ☐ *Yes, she looks like a real person*
  - ☐ *No, she doesn't look like a real person*

As you can probably guess, there are problems deciding on the range of possible answers. Maybe you thought of other ones. In order to get a good range of answers for the second question, a large number of people would have to be interviewed before the questionnaire is constructed to identify all the possible answers and then those could be used to determine what should be offered.

ACTIVITY 13.3 Write three or four semi-structured interview questions to find out if Ananova is popular with your friends. Make the questions general.

Comment Here are some suggestions:

(a) Would you listen to the news using Ananova?
If yes, then ask, why?
If no, then ask, why not?

(b) Is Ananova's appearance attractive to you?
If yes, then say, Tell me more, what did you like?
If no, then *say, What* don't you find attractive?

(c) Is there anything else you want to say about Ananova?

ACTIVITY 13.4 Prepare the full interview script to evaluate Ananova, including a description of why you are doing the interview, and an informed consent form, and the exact questions. Use the DECIDE framework for guidance. Practice the interview on your own, audiotape yourself, and then listen to it and review your performance. Then interview two peers and be reflective. What did you learn from the experience?

Comment You probably found it harder than you thought to interview smoothly and consistently. Did you notice an improvement when you did the second interview? Were some of the questions poorly worded. Piloting your interview often reveals poor or ambiguous questions that you then have a chance to refine before holding the first proper interview.

13.2.5 Group interviews

One form of group interview is the focus group that is frequently used in marketing, political campaigning, and social sciences research. Normally three to 10 people are involved. Participants are selected to provide a representative sample of typical users; they normally share certain characteristics. For example, in an evaluation of a university website, a group of administrators, faculty, and students may be called to form three separate focus groups because they use the web for different purposes.

The benefit of a focus group is that it allows diverse or sensitive issues to be raised that would otherwise be missed. The method assumes that individuals develop opinions within a social context by talking with others. Often questions posed to focus groups seem deceptively simple but the idea is to enable people to put forward their own opinions in a supportive environment. A preset agenda is developed to guide the discussion but there is sufficient flexibility for a facilitator to follow unanticipated issues as they are raised. The facilitator guides and prompts discussion and skillfully encourages quiet people to participate and stops verbose ones from dominating the discussion. The discussion is usually recorded for later analysis in which participants my be invited to explain their comments more fully.

Focus groups appear to have high validity because the method is readily understood and findings appear believable (Marshall and Rossman, 1999). Focus groups are also attractive because they are low-cost, provide quick results, and can easily be scaled to gather more data. Disadvantages are that the facilitator needs to be skillful so that time is not wasted on irrelevant issues. It can also be difficult to get people together in a suitable location. Getting time with any interviewees can be difficult, but the problem is compounded with focus groups because of the number of people involved. For example, in a study to evaluate a university website the evaluators did not expect that getting participants would be a problem. However, the study was scheduled near the end of a semester when students had to hand in their work, so strong incentives were needed to entice the students to participate in the study. It took an increase in the participation fee and a good lunch to convince students to participate.

13.2.6 Other sources of interview-like feedback

Telephone interviews are a good way of interviewing people with whom you cannot meet. You cannot see body language, but apart from this telephone interviews have much in common with face-to-face interviews.

Online interviews, using either asynchronous communication as in email or synchronous communication as in chats, can also be used. For interviews that involve sensitive issues, answering questions anonymously may be preferable to meeting face to face. If, however, face to face meetings are desirable but impossible because of geographical distance, video-conferencing systems can be used (but remember the drawbacks discussed in Chapter 4). Feedback about a product can also be obtained from customer help lines, consumer groups, and online customer communities that provide help and support.

At various stages of design, it is useful to get quick feedback from a few users. These short interviews are often more like conversations in which users are asked their opinions. Retrospective interviews can be done when doing field studies to check with participants that the interviewer has correctly understood what was happening.

DILEMMA What They Say and What They Do!

What users say isn't always what they do. People sometimes give the answers that they think show them in the best light, or they may just forget what happened or how long they spent on a particular activity.

So, can evaluators believe all the responses they get? Are the respondents giving "the truth" or are they simply giving the answers that they think the evaluator wants to hear?

It isn't possible to avoid this behavior, but it is important to be aware of it and to reduce such biases by getting a large number of participants or by using a combination of techniques. Questions that suggest particular responses should also be avoided.

13.2.7 Data analysis and interpretation

Analysis of unstructured interviews can be time-consuming, though their contents can be rich. Typically each interview question is examined in depth in a similar way to observation data discussed in Chapter 12. **A** coding form may be developed, which may be predetermined or may be developed during data collection as evaluators are exposed to the range of issues and learn about their relative importance. Alternatively, comments may be clustered along themes and anonymous quotes used to illustrate points of interest. Tools such a NUDIST and Ethnograph can be useful for qualitative analyses as mentioned in Chapter 12. Which type of analysis is done depends on the goals of the study, as does whether the whole interview is transcribed, only part of it, or none of it. Data from structured interviews is usually analyzed quantitatively as in questionnaires which we discuss next.

13.3 Asking users: questionnaires

Questionnaires are a well-established technique for collecting demographic data and users' opinions. They are similar to interviews and can have ***closed*** or open questions. Effort and skill are needed to ensure that questions are clearly worded and the data collected can be analyzed efficiently. Questionnaires can be used on their own or in conjunction with other methods to clarify or deepen understanding. In the HutchWorld study discussed in Chapter 10, for example, you read how questionnaires were used along with observation and usability testing. The methods and questions used depends on the context, interviewees and so on.

The questions asked in a questionnaire, and those used in a structured interview are similar, so how do you know when to use which technique? One advantage of questionnaires is that they can be distributed to a large number of people. Used in this way, they provide evidence of wide general opinion. On the other hand, structured interviews are easy and quick to conduct in situations in which people will not stop to complete a questionnaire.

13.3.1 Designing questionnaires

Many questionnaires start by asking for basic demographic information (e.g., gender, age) and details of user experience (e.g., the time or number of years spent using computers, level of expertise, etc.). This background information is useful in finding out the range within the sample group. For instance, a group of people who are using the web for the first time are likely to express different opinions to another group with five years of web experience. From knowing the sample range, a **designer might develop two different versions or veer towards the needs of one of** the groups more because it represents the target audience.

Following the general questions, specific questions that contribute to the evaluation goal are asked. If the questionnaire is long, the questions may be subdivided into related topics to make it easier and more logical to complete.

Box 13.1 contains an excerpt from a paper questionnaire designed to evaluate users' satisfaction with some specific features of a prototype website for career changers aged 34–59 years.

BOX 13.1 An Excerpt from A User Satisfaction Questionnaire Used to Evaluate A Website for Career Changers *(Andrews et al., 2001)*

Notice that in the following excerpt users are asked to circle appropriate responses, and check the box that most closely describes their opinion: these are commonly used techniques. Fewer than fifty participants were involved in this study, so inviting them to write on open-ended comment suggesting recommendations for change was manageable. It would have been difficult to collect this information with closed questions, since good suggestions would undoubtedly have been missed because the evaluator would probably not have thought to ask about them.

Participant #: ___________

Please circle the most appropriate selection:

Age Range:	34–39	40–49	50–59	
Gender:	Male	Female		
Career-Change Status:	Exploring	In-Progress	Completed	
Internet/Web Experience				
Research, Information Gathering	Daily	Weekly	Monthly	Never
Bulletin Board Posting	Daily	Weekly	Monthly	Never
Chat Room Usage	Daily	Weekly	Monthly	Never

Please rate (i.e., check the box to show) agreement or disagreement with the following statements:

Question	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The navigation language on the links is clear and easy to understand					
The website site contains information that would be useful to me					
Information on the website is easy to find					
The "Center Design" presents information in an aesthetically pleasing manner					
The website pages are confusing and difficult to read					
I prefer darker colors to lighter colors for display					
It is apparent from the first website page (homepage) what the purpose of the website is.					

Please add any recommendations for changes to the overall design, language or navigation of the website on the back of this paper.

Thanks for your participation in the testing of this prototype.

The following is a checklist of general advice for designing a questionnaire:

- Make questions clear and specific.
- When possible, ask closed questions and offer a range of answers.
- Consider including a "no-opinion" option for questions that seek opinions.
- Think about the ordering of questions. The impact of a question can be influenced by question order. General questions should precede specific ones.
- Avoid complex multiple questions.
- When scales are used, make sure the range is appropriate and does not overlap.
- Make sure that the ordering of scales (discussed below) is intuitive and consistent, and be careful with using negatives. For example, it is more intuitive in a scale of 1 to 5 for 1 to indicate low agreement and 5 to indicate high agreement. Also be consistent. For example, avoid using 1 as low on some scales and then as high on others. A subtler problem occurs when most questions are phrased as positive statements and a few are phrased as negatives. However, advice on this issue is more controversial as some evaluators argue that changing the direction of questions helps to check the users' intentions. Scales such as those used in Box 13.1 are also preferred by some evaluators.
- Avoid jargon and consider whether you need different versions of the questionnaire for different populations.
- Provide clear instructions on how to complete the questionnaire. For example, if you want a check put in one of the boxes, then say so. Questionnaires can make their message clear with careful wording and good typography.
- A balance must be struck between using white space and the need to keep the questionnaire as compact as possible. Long questionnaires cost more and deter participation.

13.3.2 Question and response format

Different types of questions require different types of responses. Sometimes discrete responses are required, such as "Yes" or "No." For other questions it is better to ask users to locate themselves within a range. Still others require a single preferred opinion. selecting the most appropriate makes it easier for respondents to be able to answer. Furthermore, questions that accept a specific answer can be categorized more easily. Some commonly used formats are described below.

Check boxes and ranges

The range of answers to demographic questionnaires is predictable. Gender, for example, has two options, male or female, so providing two boxes and asking respondents to check the appropriate one, or circle a response, makes sense for collecting this information (as in Box 13.1). A similar approach can be adopted if details of age are needed. But since some people do not like to give their exact age, many questionnaires ask respondents to specify their age as a range (Box 13.1). A common design error arises when the ranges overlap. For example, specifying two ranges as 15–20, 20–25 will cause confusion: which box do people who are 20 years old check? Making the ranges 14–19, 20–24 avoids this problem.

A frequently asked question about ranges is whether the interval must be equal in all cases. The answer is that it depends on what you want to know. For example, if you want to collect information for the design of an e-commerce site to sell life insurance, the target population is going to be mostly people with jobs in the age range of, say, 21-65 years. You could, therefore, have just three ranges: under 21, 21–65 and over 65. In contrast, if you are interested in looking at ten-year cohort groups for people over 21 the following ranges would be best: under 21, 22–31, 32–41, etc.

There are a number of different types of rating scales that can be used, each with its own purpose (see Oppenheim, 1992). Here we describe two commonly used scales, Likert and semantic differential scales.

The purpose of these is to elicit a range of responses to a question that can be compared across respondents. They are good for getting people to make judgments about things, e.g. how easy, how usable etc., and therefore are important for usability studies.

Likert scales rely on identifying a set of statements representing a range of possible opinions, while semantic differential scales rely on choosing pairs of words that represent the range of possible opinions. Likert scales are the most commonly used scales because identifying suitable statements that respondents will understand is easier than identifying semantic pairs that respondents interpret as intended.

Likert Scales

Likert scales are used for measuring opinions, attitudes, and beliefs, and consequently they are widely used for evaluating user satisfaction with products as in the HutchWorld evaluation described in Chapter 10. For example, users' opinions about the use of color in a website could be evaluated with a Likert scale using a range of numbers (1) or with words (2):

(1) The use of color is excellent: (where 1 represents strongly agree and 5 represents strongly disagree)

1	2	3	4	5
☐	☐	☐	☐	17

(2) The use of color is excellent:

strongly agree	agree	OK	disagree	strongly disagree
☐	☐	☐	☐	☐

Below are some steps for designing Likert scales:

- Gather a pool of short statements about the features of the product that are to be evaluated e.g., "This control panel is easy to use." A brainstorming session with peers in which you examine the product to be evaluated is a good way of doing this.

- Divide the items into groups with about the same number of positive and negative statements in each group. Some evaluators prefer to have all negative or all positive questions, while others use a mix of positive and negative questions, as we have suggested here. Deciding whether to phrase the questionnaire positively or negatively depends partly on the complexity of the questionnaire and partly on the evaluator's preferences. The designers of **QUIS** (Box 13.2) (Chin et al., 1988), for example, decided not to mix negative and positive statements because the questionnaire was already complex enough without forcing participants to pay attention to the direction of the argument.
- Decide on the scale. **QUIS** (Box 13.2) uses a 9-point scale, and because it is a general questionnaire that will be used with a wide variety of products it also includes N/A (not applicable,) as a category. Many questionnaires use 7- or 5-point scales and there are also 3-point scales. Arguments for the number of points go both ways. Advocates of long scales argue that they help to show discrimination, as advocated by the **QUIS** team (Chin et al., 1988). Rating features on an interface is more difficult for most people than, say, selecting among different flavors of ice cream, and when the task is difficult there is evidence to show that people "hedge their bets." Rather than selecting the poles of the scales if there is no right or wrong, respondents tend to select values nearer the center. The counter-argument is that people cannot be expected to discern accurately among points on a large scale, so any scale of more than five points is unnecessarily difficult to use.

  Another aspect to consider is whether the scale should have an even or odd number of points. An odd number provides a clear central point. On the

BOX 13.2 QUIS, Questionnaire for User Interaction Satisfaction

The Questionnaire for User Interaction Satisfaction (QUIS) developed by the University of Maryland Human-Computer Interaction Laboratory is one of the most widely used questionnaires for evaluating interfaces (Chin et al., 1988; Shneiderman, 1998a). Although developed for evaluating user satisfaction, it is frequently applied to other aspects of interaction design. An advantage of this questionnaire is that it has gone through many cycles of refinement and has been used for hundreds of evaluation studies, so it is well tried and tested. The questionnaire consists of the following 12 parts that can be used in total or in parts:

- system experience (i.e., time spent on this system)
- past experience (i.e., experience with other systems)
- overall user reactions
- screen design
- terminology and system information
- learning (i.e., to operate the system)
- system capabilities (i.e., the time it takes to perform operations)
- technical manuals and online help
- online tutorials
- multimedia
- teleconferencing
- software installation

Notice that the third part of QUIS assesses users' overall reactions. Evaluators often use this part on its own because it is short so people are likely to respond.

other hand, an even number forces participants to make a decision and prevents them from sitting on the fence.

- Select items for the final questionnaire and reword as necessary to make them clear.

Semantic differential scales

Semantic differential scales are used less frequently than Likert scales. They explore a range of bipolar attitudes about a particular item. Each pair of attitudes is represented as a pair of adjectives. The participant is asked to place a cross in one of a number of positions between the two extremes to indicate agreement with the poles, as shown in Figure 13.2. The score for the evaluation is found by summing the scores for each bipolar pair. Scores can then be computed across groups of participants. Notice that in this example the poles are mixed so that good and bad features are distributed on the right and the left. In this example there are seven positions on the scale.

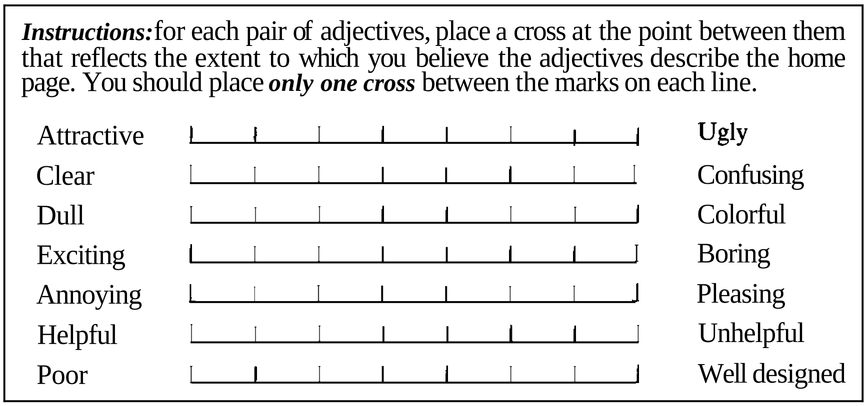


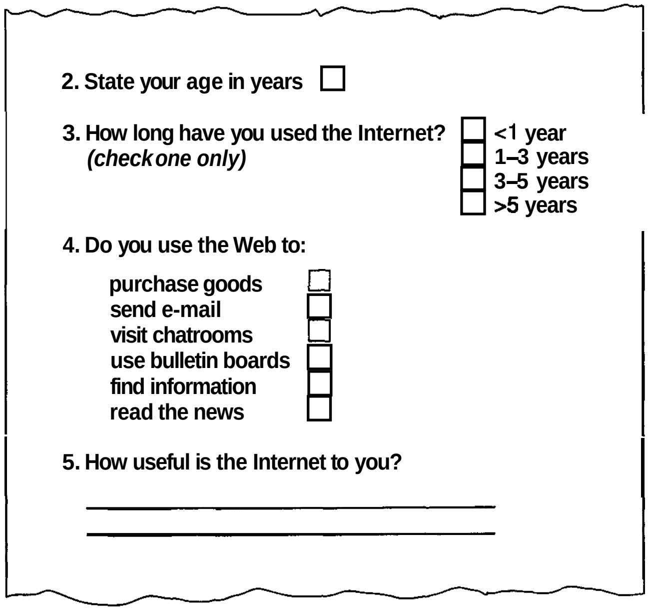
Figure 13.2 An example of a semantic differential scale.

ACTIVITY 13.5 Spot the four poorly designed features in Figure **13.3.**

Comment

Some of the features that could be improved include:

- Request for exact age. Many people prefer not to give this information and would rather position themselves in a range.
- Years of experience is indicated with overlapping scales, **i.e.**, <1, 1–3, 3–5, etc. How do you answer if you have 1, 3, or 5 years of experience?
- The questionnaire doesn't tell you whether you should check one, two, or as many boxes as you wish.
- The space left for people to write their own information is too small, and this will annoy them and deter them from giving their opinions.


2. State your age in years ☐

3. How long have you used the Internet? *(check one only)*
☐ <1 year
☐ 1–3 years
☐ 3–5 years
☐ >5 years

4. Do you use the Web to:
purchase goods ☐
send e-mail ☐
visit chatrooms ☐
use bulletin boards ☐
find information ☐
read the news ☐

5. How useful is the Internet to you?

Figure 13.3 A questionnaire with poorly designed features.

13.3.3 Administering questionnaires

Two important issues when using questionnaires are reaching a representative sample of participants and ensuring a reasonable response rate. For large surveys, potential respondents need to be selected using a sampling technique. However, interaction designers tend to use small numbers of participants, often fewer than twenty users. One hundred percent completion rates often are achieved with these small samples, but with larger, more remote populations, ensuring that surveys are returned is a well-known problem. Forty percent return is generally acceptable for many surveys but much lower rates are common.

Some ways of encouraging a good response include:

- Ensuring the questionnaire is well designed so that participants do not get annoyed and give up.
- Providing a short overview section, as in QUIS (Box 13.2), and telling respondents to complete just the short version if they do not have time to complete the whole thing. This ensures that you get something useful returned.
- Including a stamped, self-addressed envelope for its return.
- Explaining why you need the questionnaire to be completed and assuring anonymity.
- Contacting respondents through a follow-up letter, phone call or email.
- Offering incentives such as payments.

13.3.4 Online questionnaires

Online questionnaires are becoming increasingly common because they are effective for reaching large numbers of people quickly and easily. There are two types: email and web-based. The main advantage of email is that you can target specific users. However, email questionnaires are usually limited to text, whereas web-based questionnaires are more flexible and can include check boxes, pull-down and pop-up menus, help screens, and graphics (Figure 13.4). web-based questionnaires can also provide immediate data validation and can enforce rules such as select only one response, or certain types of answers such as numerical, which cannot be done in email or with paper. Other advantages of online questionnaires include (Lazar and Preece, 1999):

- Responses are usually received quickly.
- Copying and postage costs are lower than for paper surveys or often non-existent.
- Data can be transferred immediately into a database for analysis.
- The time required for data analysis is reduced.
- Errors in questionnaire design can be corrected easily (though it is better to avoid them in the first place).

A big problem with web-based questionnaires is obtaining a random sample of respondents. Few other disadvantages have been reported with online questionnaires, but there is some evidence suggesting that response rates may be lower online than with paper questionnaires (Witmer et al., 1999).

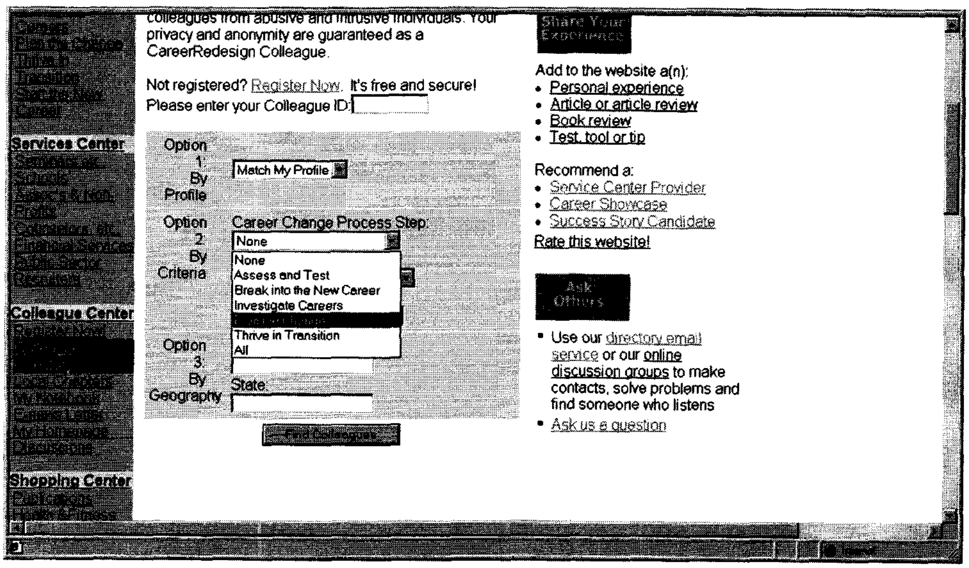


Figure 13.4 An excerpt from a web-based questionnaire showing pull-down menus.

Developing a web-based questionnaire

Developing a successful web-based questionnaire involves designing it on paper, developing strategies for reaching the target population, and then turning the paper version into a web-based version (Lazar and Preece, 1999).

It is important to devise the questionnaire on paper first, following the general guidelines introduced above, such as paying attention to the clarity and consistency of the questions, questionnaire layout, and so on. Only once the questionnaire has been reviewed and the questions refined adequately should it be translated into a web-based version. If reaching your target population is an issue, e.g., if some of them may not have access to the web, the paper version may be administered to them, but be careful to maintain consistency between the web-based version and the original paper version.

Identifying a random sample of a population so that the results are indicative of the whole population may be difficult, if not impossible, to achieve especially if the size and demography of the population is not known, as is often the case in Internet research. This has been a criticism of several online surveys including Georgia Tech's GVU survey, one of the first online surveys. This survey collects demographic and activity information from Internet users and has been distributed twice yearly since 1994. The policy that GVU employs to deal with this difficult sampling issue is to make as many people aware of the GVU survey as possible so that a wide variety of participants are encouraged to participate. However, even these efforts do not avoid biased sampling, since participants are self-selecting. Indeed, some survey experts are vehemently opposed to such methods and instead propose using national census records to sample offline (Nie & Ebring, 2000). In some countries, web-based questionnaires are used in conjunction with television to elicit viewers' opinions of programs and political events, and many such questionnaires now say that their results are "not scientific" when they cite them, meaning that unbiased sampling was not done. A term that is gaining popularity is convenience sampling, which is another way of saying that the sample includes those who were available rather than those selected using scientific sampling.

Turning the paper questionnaire into a web-based version requires four steps.

1. Produce an error-free interactive electronic version from the original paper-based one. This version should provide clear instructions and be free of input errors. For example, if just one box should be checked, the other attempts should be rejected automatically. It may also be useful to embed feedback and pop-up help within the questionnaire.
2. Make the questionnaire accessible from all common browsers and readable from different-size monitors and different network locations. Specialized software or hardware should be avoided. The need to download software also deters novice users and should be avoided.
3. Make sure information identifying each respondent will be captured and stored confidentially because the same person may submit several completed surveys. This can be done by recording the Internet domain name or the IP address of the respondent, which can then be transferred directly to a

database. However, this action could infringe people's privacy and the legal situation should be checked. Another way is to access the transfer and referrer logs from the web server, which provide information about the domains from which the web-based questionnaire was accessed. Unfortunately, people can still send from different accounts with different IP addresses, so additional identifying information may also be needed.

4. User-test the survey with pilot studies before distributing.

Commercial questionnaires are becoming available via the Internet. Two examples are SUMI and MUMMS, which are briefly discussed in Box 13.3.

BOX 13.3 Questionnaire Tools

SUMI (Software Usability Measurement Inventory) was developed in the early 1990s as part of a European project. The aim was to develop a standardized tool to evaluate users' reactions to a piece of software. More recently a new version has been developed, known as MUMMS (Measuring the Usability of Multi-Media Systems), that, as the name implies, is geared more towards current software in which multimedia is assumed to be a component. This questionnaire focuses on five concepts:

- how much the product captures the user's emotional responses
- how much the user feels in control of the software
- the degree to which the users can achieve their goals using the software
- the extent to which the product seems to assist the user
- the ease with which the user can learn to use the product

The developers are also planning to include a new concept that they are calling "excitement." This would address two kinds of user experience goals (emotional responses and excitement) and four usability goals. More information about SUMI and MUMMS can be found at www.ucc.ie/hfrg/questionnaires/

13.3.5 Analyzing questionnaire data

Having collected a set of questionnaire responses, you need to know what to do with the data. The first step is to identify any trends or patterns. Using a spreadsheet like Excel to hold the data can help in this initial analysis. Often only simple statistics are needed such as the number or percentage of responses in a particular category. If the number of participants is small, under ten for example, giving actual numbers is more honest, but for larger numbers of responses percentages are useful for standardizing the data, particularly if you want to compare two or more sets of responses. Bar charts can also be used to display data graphically. More advanced statistical techniques such as cluster analysis can also be used to show whether there is a relationship between question responses.

13.4 Asking experts: inspections

Sometimes users are not easily accessible or involving them is too expensive or takes too long. In such circumstances, experts or combinations of experts and users can provide feedback. Various inspection techniques began to be developed as alternatives to usability testing in the early 1990s. These included various kinds of expert evaluations or ***reviews,*** such as heuristic evaluations and walkthroughs, in which experts inspect the human-computer interface and predict problems users would have when interacting with it. Typically these techniques are relatively inexpensive and easy to learn as well as being effective, which makes them appealing. They are similar to some software engineering practices where code and other types of inspections have been conducted for years. In addition, they can be used at any stage of a design project, including early design before well-developed prototypes are available.

13.4.1 Heuristic evaluation

Heuristic evaluation is an informal usability inspection technique developed by Jakob Nielsen and his colleagues (Nielsen, 1994a) in which experts, guided by a set of usability principles known as ***heuristics,*** evaluate whether user-interface elements, such as dialog boxes, menus, navigation structure, online help, etc., conform to the principles. These heuristics closely resemble the high-level design principles and guidelines discussed in Chapters 1 and 8, e.g., making designs consistent, reducing memory load, and using terms that users understand. When used in evaluation, they are called heuristics. The original set of heuristics was derived empirically from an analysis of 249 usability problems (Nielsen, 1994b). We list the latest here (also in Chapter 1), this time expanding them to include some of the questions addressed when doing evaluation:

- ***Visibility of system status***
  Are users kept informed about what is going on?
  Is appropriate feedback provided within reasonable time about a user's action?
- ***Match between system and the real world***
  Is the language used at the interface simple?
  Are the words, phrases and concepts used familiar to the user?
- ***User control and freedom***
  Are there ways of allowing users to easily escape from places they unexpectedly find themselves in?
- ***Consistency and standards***
  Are the ways of performing similar actions consistent?
- ***Help users recognize, diagnose, and recover from errors***
  Are error messages helpful?
  Do they use plain language to describe the nature of the problem and suggest a way of solving it?
- ***Error prevention***
  Is it easy to make errors?
  If so where and why?
- ***Recognition rather than recall***
  Are objects, actions and options always visible?

- ***Flexibility and efficiency of use***
  Have accelerators (i.e., shortcuts) been provided that allow more experienced users to carry out tasks more quickly?
- ***Aesthetic and minimalist design***
  Is any unnecessary and irrelevant information provided?
- ***Help and documentation***
  Is help information provided that can be easily searched and easily followed?

However, some of these core heuristics are too general for evaluating new products coming onto the market and there is a strong need for heuristics that are more closely tailored to specific products. For example, Nielsen (1999) suggests that the following heuristics are more useful for evaluating commercial websites, and makes them memorable by introducing the acronym H O M E R U N:

- High-quality content
- Often updated
- Minimal download time
- Ease of use
- Relevant to users' needs
- Unique to the online medium
- Netcentric corporate culture

Different sets of heuristics for evaluating toys, WAP devices, online communities, wearable computers, and other devices are needed, so evaluators must develop their own by tailoring Nielsen's heuristics and by referring to design guidelines, market research, and requirements documents. Exactly which heuristics are the best and how many are needed are debatable and depend on the product.

Using a set of heuristics, expert evaluators work with the product role-playing typical users and noting the problems they encounter. Although other numbers of experts can be used, empirical evidence suggests that five evaluators usually identify around 75% of the total usability problems, as shown in Figure 13.5 (Nielsen,

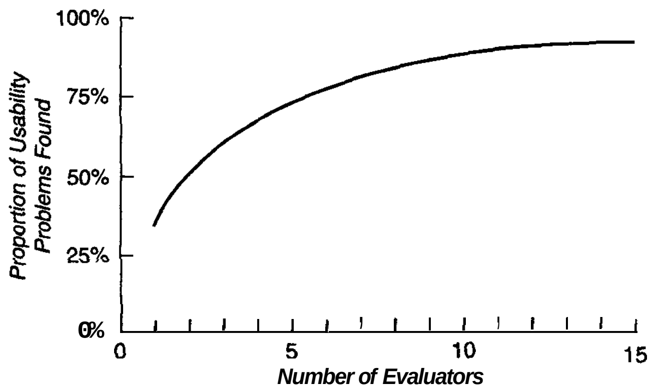
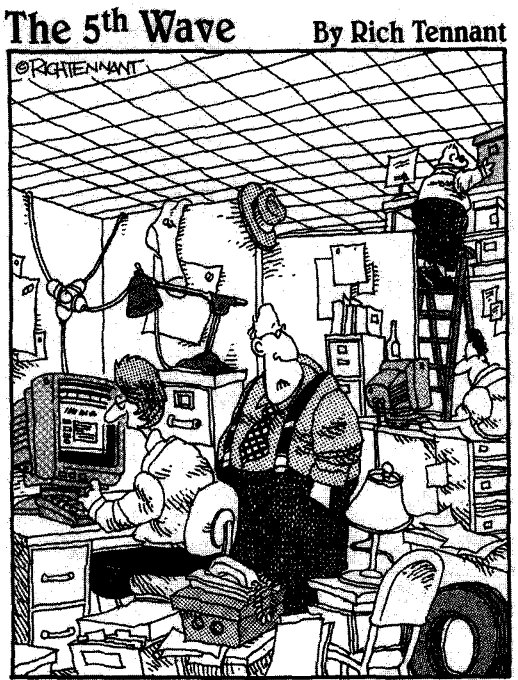


Figure 13.5 Curve showing the proportion of usability problems in an interface found by heuristic evaluation using various numbers of evaluators. The curve represents the average of six case studies of heuristic evaluation.



"Just how accurately should my Web site reflect my place of business?"

1994a). However, skillful experts can capture many of the usability problems by themselves, and many consultants now use this technique as the basis for critiquing interactive devices—a process that has become know as an ***expert crit*** in some countries. Because users and special facilities are not needed for heuristic evaluation and it is comparatively inexpensive and quick, it is also known as ***discount evaluation.***

13.4.2 Doing heuristic evaluation

Heuristic evaluation is one of the most straightforward evaluation methods. The evaluation has three stages:

1. ***The briefing session*** in which the experts are told what to do. **A** prepared script is useful as a guide and to ensure each person receives the same briefing.
2. The ***evaluation period*** in which each expert typically spends 1–2 hours ***independently*** inspecting the product, using the heuristics for guidance. The experts need to take ***at least two*** passes through the interface. The ***first pass*** gives a feel for the flow of the interaction and the product's scope. The ***second pass*** allows the evaluator to focus on specific interface ele-

ments in the context of the whole product, and to identify potential usability problems.

If the evaluation is for a functioning product, the evaluators need to have some specific user tasks in mind so that exploration is focused. Suggesting tasks may be helpful but many experts do this automatically. However, this approach is less easy if the evaluation is done early in design when there are only screen mockups or a specification; the approach needs to be adapted to the evaluation circumstances. While working through the interface, specification or mockups, a second person may record the problems identified, or the evaluator may think aloud. Alternatively, she may take notes herself. Experts should be encouraged to be as specific as possible and to record each problem clearly.

3. The *debriefing session* in which the experts come together to discuss their findings and to prioritize the problems they found and suggest solutions.

The heuristics focus the experts' attention on particular issues, so selecting appropriate heuristics is therefore critically important. Even so, there is sometimes less agreement among experts than is desirable, as discussed in the dilemma below.

There are fewer practical and ethical issues in heuristic evaluation than for other techniques because users are not involved. A week is often cited as the time needed to train experts to be evaluators (Nielsen and Mack, 1994), but this of course depends on the person's expertise. The best experts will have expertise in both interaction design and the product domain. Typical users can be taught to do

DILEMMA Problems or False Alarms?

You might think that heuristic evaluation is a panacea for designers, and that it can reveal all that is wrong with a design. However, it has problems. Several independent studies compare heuristic evaluation with other techniques, particularly user testing, indicating that the different approaches often identify *different* problems and that sometimes heuristic evaluation misses severe problems (Karat, 1994). This argues for using complementary techniques. Furthermore, heuristic evaluation should not be thought of as a replacement for user testing.

Another problem that Bill Bailey (2001) warns about is of experts reporting problems that don't exist. In other words, some of the experts' predictions are wrong. Bailey cites analyses from three published sources showing that about 33% of the problems reported were real usability problems, some of which were serious, others trivial. However, the heuristic evaluators missed about 21% of users' problems. Furthermore, about 43% of the problems identified by the experts were *not* problems at all; they were false alarms! Bailey points out that if we do the arithmetic and round up the numbers, what this comes down to is that only about *half* the problems identified are true problems. "More specifically, for every true usability problem identified, there will be a little over one false alarm (1.2) and about one half of one missed problem (0.6). If this analysis is true, heuristic evaluators tend to identify more false alarms and miss more problems than they have true hits."

How can the number of false alarms or missed serious problems be reduced? Checking that experts really have the expertise that they claim would help but how can you do this? One way to over come biases is to have several evaluators. This helps to reduce the impact of one person's bias of poor performance. Using heuristic evaluation along with user testing and other techniques is also a good idea.

heuristic evaluation, although there have been claims that it is not very successful (Nielsen, 1994a). However, some closely related methods take a team approach that involves users (Bias, 1994).

13.4.3 Heuristic evaluation of websites

In this section we examine heuristics for evaluating websites. We begin by discussing MEDLINEplus, a medical information website created by the National Library of Medicine (NLM) to provide health information for patients, doctors, and researchers (Cogdill, 1999). The home page and two other screens are shown in Figures 13.6–13.8.

In 1999 usability consultant Keith Cogdill was commissioned by NLM to evaluate MEDLINEplus. Using a combination of his own knowledge of the users' tasks, problems that had already been reported by users, and advice from documented sources (Shneiderman, 1998a; Nielsen, 1993; Dumas and Redish, 1999), Cogdill identified the seven heuristics listed below. Some of the heuristics resemble Nielsen's original set, but have been tailored for evaluating MEDLINEplus.

- ***Internal consistency.***
  The user should not have to speculate about whether different phrases or actions carry the same meaning.

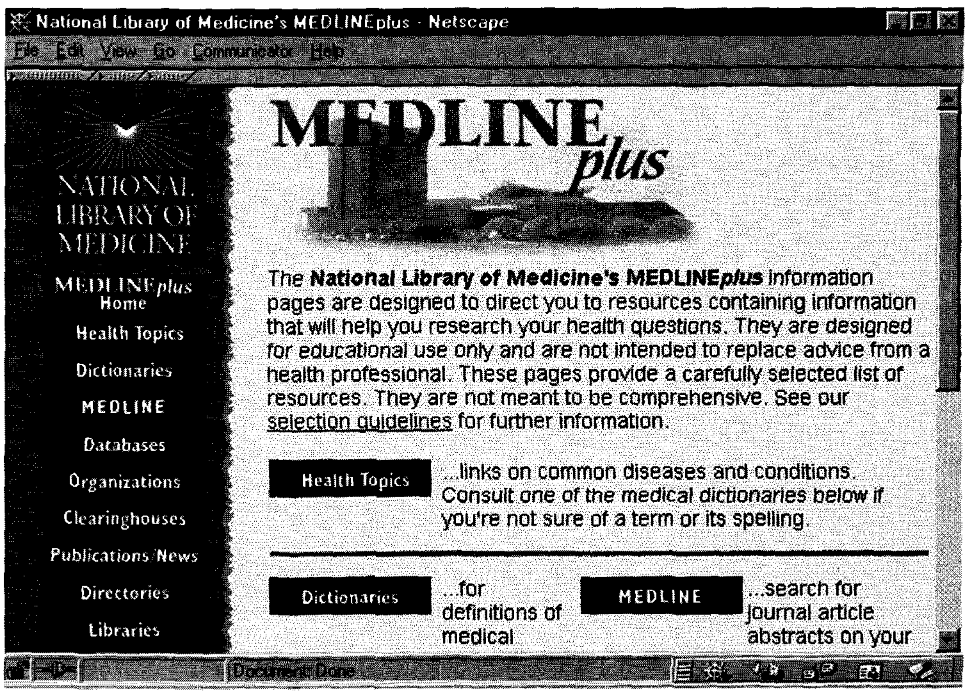


Figure 13.6 Home **page** of **MEDLINEplus.**

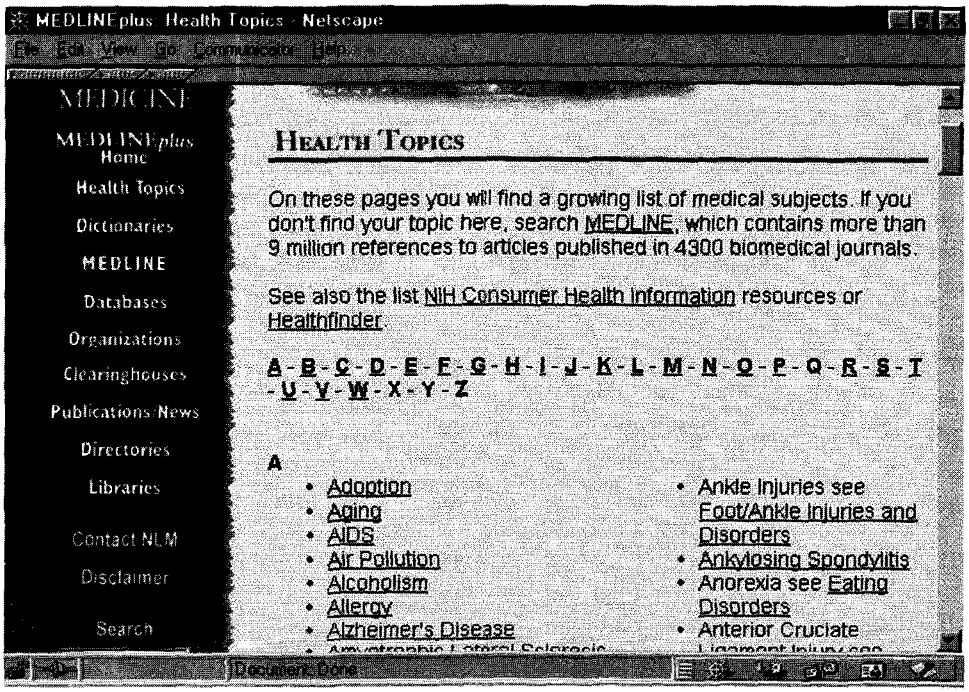


Figure 13.7 Clicking Health Topics on the home page produced this page.

- ***Simple dialog.***
  The dialog with the user should not include information that is irrelevant, unnecessary, or rarely needed. The dialog should be presented in terms familiar to the user and not be system-oriented.
- ***Shortcuts.***
  The interface should accommodate both novice and experienced users.
- ***Minimizing the user's memory load.***
  The interface should not require the user to remember information from one part of the dialog to another.
- ***Preventing errors.***
  The interface should prevent errors from occurring.
- ***Feedback.***
  The system should keep the user informed about what is taking place.
- ***Internal locus of control.***
  Users who choose system functions by mistake should have an "emergency exit" that lets them leave the unwanted state without having to engage in an extended dialog with the system,

These heuristics were given to three expert evaluators who independently evaluated MEDLINEplus. Their comments were then compiled and a meeting was

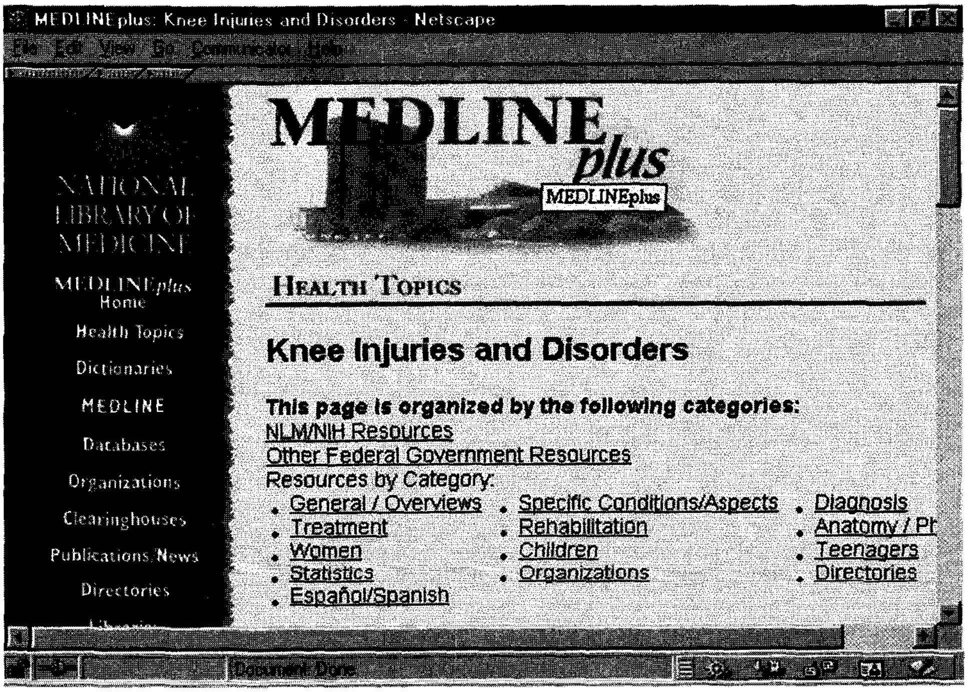


Figure 13.8 Categories of links within Health Topics for knee injuries.

called to discuss their findings and suggest strategies for addressing problems. The following points were among their findings:

- ***Layout.***
  All pages within MEDLINEplus have a relatively uncomplicated vertical design. The home page is particularly compact, and all pages are well suited for printing. The use of graphics is conservative, minimizing the time needed to download pages.
- ***Internal consistency.***
  The formatting of pages and presentation of the logo are consistent across the website. Justification of text, fonts, font sizes, font colors, use of terms, and links labels are also consistent.

The experts also suggested improvements, including:

- ***Arrangement of health topics.***
  Topics should be arranged alphabetically as well as in categories. For example, health topics related to cardiovascular conditions could appear together.
- ***Depth of navigation menu.***
  Having a higher "fan-out" in the navigation menu in the left margin would enhance usability. By this they mean that more topics should be listed on the

surface, giving many short menus rather than a few deep ones (see the experiment on breadth versus depth in Chapter 14 which provides evidence to justify this.)

Turning design guidelines into heuristics for the web

The following list of guidelines for evaluating websites was compiled from several sources and grouped into three categories: *navigation, access,* and *information design* (Preece, 2000). These guidelines provide a basis for developing heuristics by converting them into questions.

Navigation One of the biggest problems for users of large websites is navigating around the site. The phrase "lost in cyberspace" is understood by every web user. The following six guidelines (from Nielsen (1998) and others) are intended to encourage good navigation design:

- *Avoid orphan pages i.e. pages that are not connected to the home page, because they lead users into dead ends.*
  Are there any orphan pages? Where do they go to?
- *Avoid long pages with excessive white space that force scrolling.*
  Are there any long pages? Do they have lots of white space or are they full of texts or lists?
- *Provide navigation support, such as a strong site map that is always present* (Shneiderman, 1998b).
  Is there any guidance, e.g. maps, navigation bar, menus, to help users find their way around the site?
- *Avoid narrow, deep, hierarchical menus that force users to burrow deep into the menu structure.*
  Empirical evidence indicates that broad shallow menus have better usability than a few deep menus (Larson and Czerwinski, 1998; Shneiderman, 1998b).
- *Avoid non-standard link colors.*
  What color is used for links? Is it blue or another color? If it is another color, then is it obvious to the user that it is a hyperlink?
- *Provide consistent look and feel for navigation and information design.*
  Are menus used, named, and positioned consistently? Are links used consistently?

Access Accessing many websites can be a problem for people with slow Internet connections and limited processing power. In addition, browsers are often not sensitive to errors in URLs. Nielsen (1998) suggests the following guidelines:

- *Avoid complex URLs.*
  Are the URLs complex? Is it easy to make typing mistakes when entering them?

- ***Avoid long download times that annoy users.***
  Are there pages with lots of graphics? How long does it take to download each page?

Information design **Information design (i.e., content comprehension and aesthetics) contributes to users' understanding and impressions of the site as you can see in Activity 13.6.**

ACTIVITY 13.6 Consider the following design guidelines for information design and for each one suggest a question that could be used in heuristic evaluation:

- ***Outdated or incomplete information is to be avoided*** (Nielsen, 1998). It creates a poor impression with users.
- ***Good graphical design is important.*** Reading long sentences, paragraphs, and documents is difficult on screen, so break material into discrete, meaningful chunks to give the website structure (Lynch and Horton, 1999).
- ***Avoid excessive use of color.*** Color is useful for indicating different kinds of information, i.e., cueing (Preece et al., 1994).
- ***Avoid gratuitous use of graphics and animation.*** In addition to increasing download time, graphics and animation soon become boring and annoying (Lynch and Horton, 1999).
- Be ***consistent.*** Consistency both within pages (e.g., use of fonts, numbering, terminology, etc.) and within the site (e.g., navigation, menu names, etc.) is important for usability and for aesthetically pleasing designs.

Comment We suggest the following questions; you may have identified others:

- ***Outdated or incomplete information.***
  Do the pages have dates on them? How many pages are old and provide outdated information?
- ***Good graphical design is important.***
  Is the page layout structured meaningfully? Is there too much text on each page?
- ***Avoid excessive use of color.***
  How is color used? Is it used as a form of coding? Is it used to make the site bright and cheerful? Is it excessive and garish?
- ***Avoid gratuitous use of graphics and animation.***
  Are there any flashing banners? Are there complex introduction sequences? Can they be short-circuited? Do the graphics add to the site?
- ***Be Consistent.***
  Are the same buttons, fonts, numbers, menu styles, etc. used across the site? Are they used in the same way?

ACTIVITY 13.7 Look at the heuristics above and consider how you would use them to evaluate a website for purchasing clothes (e.g., REI.com, which has a home page similar to that in Figure 13.9).

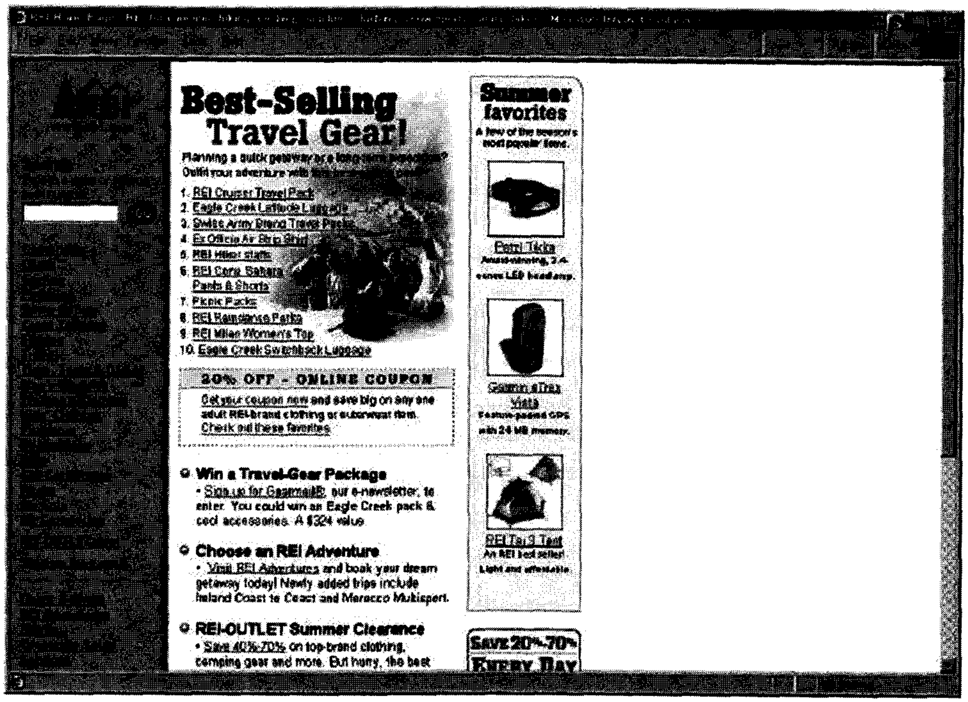


Figure 13.9 The home page is similar to that of REI.com.

While you are doing this activity think about whether the grouping into three categories is useful.

(a) Does it help you focus on what is being evaluated?

(b) Might fewer heuristics be better? Which might be combined and what are the trade-offs?

Comment

(a) Informal evaluation in which the heuristics were categorized suggests that the three categories help evaluators to focus. However, 13 heuristics is still a lot.

(b) Some heuristics can be combined and given a more general description. For example, ***providing navigation support*** and ***avoiding narrow, deep, hierarchical menus*** could be replaced with "help users develop a good mental model," but this is a more abstract statement and some evaluators might not know what is packed into it. Producing questions suitable for heuristic evaluation often results in more of them, so there is a trade-off. An argument for keeping the detail is that it reminds evaluators of the issues to consider. At present, since the web is relatively new, we can argue that such reminders are needed. Perhaps in five years they will not be.

Heuristics for online communities

As we have already mentioned, different combinations and types of heuristics are needed to evaluate different types of applications and interactive products. Another

kind of web application to which heuristics must be tailored is online communities. Here, a key concern is how to evaluate not merely usability but also how well social interaction (i.e., sociability) is supported. This topic has received less attention than the web but the following nine sets of example questions can be used as a starting point for developing heuristics to evaluate online communities (Preece, 2000):

- Sociability: Why should I join this community? (What are the benefits for me? Does the description of the group, its name, its location in the website, the graphics, etc., tell me about the purpose of the group?)
- Usability: How do I join (or leave) the community? (What do I do? Do I have to register or can I just post, and is this a good thing?)
- Sociability: What are the rules? (Is there anything I shouldn't do? Are the expectations for communal behavior made clear? Is there someone who checks that people are behaving reasonably?)
- Usability: How do I get, read and send messages? (Is there support for newcomers? IS it clear what I should do? Are templates provided? Can I send private messages?)
- Usability: Can I do what I want to do easily? (Can I navigate the site? Do I feel comfortable interacting with the software? Can I find the information and people I want?)
- Sociability: Is the community safe? (Are my comments treated with respect? Is my personal information secure? Do people make aggressive or unacceptable remarks to each other?)
- Sociability: Can I express myself as I wish? (Is there a way of expressing emotions, such as using emoticons? Can I show people what I look like or reveal aspects of my character? Can I see others? Can I determine who else is present—perhaps people are looking on but not sending messages?)
- Sociability: Do people reciprocate? (If I contribute will others contribute comments, support and answer my questions?)

  Sociability: Why should I come back? (What makes the experience worthwhile? What's in it for me? Do I feel part of a thriving community? Are there interesting people with whom to communicate? Are there interesting events?)

ACTIVITY 13.8 Go to the communities in REI.com or to another site that has bulletin boards to which customers can send comments. Social interaction **was discussed** in Chapter **4,** and **this exercise** involves picking up some of the concepts discussed there and developing heuristics to evaluate online communities. Before starting you will find it useful to familiarize yourself by carrying out the following:

- read some of the messages
- send a message
- reply to a message
- search for information
- notice how many messages have been sent and how recently

- notice whether you can see the physical relationship between messages easily
- notice whether you can post to people privately using email
- notice whether you can gain a sense of what the other people are like and the emotional content of their messages
- notice whether there is a sense of community and of individuals being present, etc.

Then use the nine questions above as heuristics to evaluate the site:

(a) How well do the questions work as heuristics for evaluating the online community for both usability and sociability issues?

(b) Could these questions form the basis for heuristics for other online communities such as HutchWorld discussed in Chapter 10?

Comment

(a) You probably found that these questions helped focus your attention on the main issues of concern. You may also have noticed that some communities are more like ghost towns than communities; they get very few visitors. Unlike the website evaluation it is therefore important to pay attention to social interaction. A community without people is not a community no matter how good the software is that supports it.

(b) HutchWorld is designed to support social interaction and offers many additional features such as support for social presence by allowing participants to represent themselves as avatars, show pictures of themselves, tell stories, etc. The nine questions above are useful but may need adapting.

13.4.4 Heuristics for other devices

The examples in the previous activities start to show how heuristics can be tailored for specific applications. However, some products are even more different than those from the desktop world of the early 1990s that gave rise to Nielsen's original heuristics. For example, computerized toys are being developed that motivate, entice and challenge, in innovative ways. Handheld devices sell partly on size, color and other aesthetic qualities—features that can have a big impact on the user experience but are not covered by traditional heuristics. Little research has been done on developing heuristics for these products, but Activity 13.9 will start you thinking about them.

ACTIVITY 13.9

Allison Druin works with children to develop web applications and computerized toys (Druin, 1999). From doing this work Allison and her team know that children like to:

- be in control and not to be controlled
- create things
- express themselves
- be social
- collaborate with other children

(a) What kind of tasks should be considered in evaluating a fluffy robot toy dog that can be programmed to move and to tell personalized stories about itself and children? The target age group for the toy is 7–9 years.

(b) Suggest heuristics to evaluate the toy.

Comment

(a) Tasks that you could consider: making the toy tell a story about the owner and two friends, making the toy move across the room, turn, and speak. You probably thought of others.

(b) The heuristics could be written to cover: being in control, being flexible, supporting expression, being motivating, supporting collaboration and being engaging. These are based on the issues raised by Druin, but the last one is aesthetic and tactile. Several of the heuristics needed would be more concerned with user experience (e.g., motivating, engaging, etc.) than with usability.

13.5 Asking experts: walkthroughs

Walkthroughs are an alternative approach to heuristic evaluation for predicting users' problems without doing user testing. As the name suggests, they involve walking through a task with the system and noting problematic usability features. Most walkthrough techniques do not involve users. Others, such as pluralistic walkthroughs, involve a team that includes users, developers, and usability specialists.

In this section we consider cognitive and pluralistic walkthroughs. Both were originally developed for desktop systems but can be applied to web-based systems, handheld devices, and products such as VCRs.

13.5.1 Cognitive walkthroughs

"Cognitive walkthroughs involve simulating a user's problem-solving process at each step in the human-computer dialog, checking to see if the user's goals and memory for actions can be assumed to lead to the next correct action." (Nielsen and Mack, 1994, p. 6). The defining feature is that they focus on evaluating designs for ease of learning—a focus that is motivated by observations that users learn by exploration (Wharton et al., 1994). The steps involved in cognitive walkthroughs are:

1. The characteristics of typical users are identified and documented and sample tasks are developed that focus on the aspects of the design to be evaluated. A description or prototype of the interface to be developed is also produced, along with a clear sequence of the actions needed for the users to complete the task.
2. A designer and one or more expert evaluators then come together to do the analysis.
3. The evaluators walk through the action sequences for each task, placing it within the context of a typical scenario, and as they do this they try to answer the following questions:
   - Will the correct action be sufficiently evident to the user? (Will the user know what to do to achieve the task?)
   - Will the user notice that the correct action is available? (Can users see the button or menu item that they should use for the next action? Is it apparent when it is needed?)

- Will the user associate and interpret the response from the action correctly? (Will users know from the feedback that they have made a correct or incorrect choice of action?)

   In other words: will users know what to do, see how to do it, and understand from feedback whether the action was correct or not?

4. As the walkthrough is being done, a record of critical information is compiled in which:
   - The assumptions about what would cause problems and why are recorded. This involves explaining why users would face difficulties,
   - Notes about side issues and design changes are made.
   - A summary of the results is compiled.
5. The design is then revised to fix the problems presented.

It is important to document the cognitive walkthrough, keeping account of what works and what doesn't. **A** standardized feedback form can be used in which answers are recorded to the three bulleted questions in step **(3)** above. The form can also record the details outlined in points 1-4 as well as the date of the evaluation. Negative answers to any of the questions are carefully documented on a separate form, along with details of the system, its version number, the date of the evaluation, and the evaluators' names. It is also useful to document the severity of the problems, for example, how likely a problem is to occur and how serious it will be for users.

The strengths of this technique are that it focuses on users' problems in detail, yet users do not need to be present, nor is a working prototype necessary. However, it is very time-consuming and laborious to do. Furthermore the technique has a narrow focus that can be useful for certain types of system but not others.

Example: Find a book at Amazon.com

This example shows a cognitive walkthrough of buying this book at Amazon.com.

Task: to buy a copy of this book from Amazon.com
Typical users: students who use the web regularly

The steps to complete the task are given below. Note that the interface for Amazon.com may have changed since we did our evaluation.

Step 1. Selecting the correct category of goods on the home page

Q. Will users know what to do?

Answer: Yes—they know that they must find "books."

Q. Will users see how to do it?

Answer: Yes—they have seen menus before and will know to select the appropriate item and click go.

Q. Will users understand from feedback whether the action was correct or not?

Answer: Yes—their action takes them to a form that they need to complete to search for the book.

Step 2. Completing the form

Q. Will users know what to do?

Answer: Yes—the online form is like a paper form so they know they have to complete it.

Answer: No—they may not realize that the form has defaults to prevent inappropriate answers because this is different from a paper form.

Q. Will users see how to do it?

Answer: Yes—it is clear where the information goes and there is a button to tell the system to search for the book.

Q. Will users understand from feedback whether the action was correct or not?

Answer: Yes—they are taken to a picture of the book, a description, and purchase details.

ACTIVITY 13.10 Activity 13.7 was about doing a heuristic evaluation of REI.com or a similar e-commerce retail site. Now go back to that site and do a cognitive walkthrough to buy something, say a pair of skis. When you have completed the evaluation, compare your findings from the cognitive walkthrough technique with those from heuristic evaluation.

Comment You probably found that the cognitive walkthrough took longer than the heuristic evaluation for evaluating the same part of the site because it examines each step of a task. Consequently, you probably did not see as much of the website. It's likely that you also got much more detailed findings from the cognitive walkthrough. Cognitive walkthrough is a useful technique for examining a small part of a system in detail, whereas heuristic evaluation is useful for examining whole or parts of systems.

Variation of the cognitive walkthrough

A useful variation on this theme is provided by Rick Spencer of Microsoft, who adapted the cognitive walkthrough technique to make it more effective with a team who were developing an interactive development environment (IDE) (Spencer, 2000). When used in its original state, there were two major problems. First, answering the three questions in step (3) and discussing the answers took too long. Second, designers tended to be defensive, often invoking long explanations of cognitive theory to justify their designs. This second problem was particularly difficult because it undermined the efficacy of the technique and the social relationships of team members. In order to cope with these problems Rick Spencer adapted the technique by reducing the number of questions and curtailing discussion. This meant that the analysis was more coarse-grained but could be completed in much less time (about 2.5 hours). He also identified a leader, the usability specialist, and set strong ground rules for the session, including a ban on defending a design, debating cognitive theory, or doing designs on the fly.

These adaptations made the technique more usable, despite losing some of the detail from the analysis. Perhaps most important of all, he directed the social interactions of the design team so that they achieved their goal.

13.5.2 Pluralistic walkthroughs

"Pluralistic walkthroughs are another type of walkthrough in which users, developers and usability experts work together to step through a [task] scenario, discussing usability issues associated with dialog elements involved in the scenario steps" (Nielsen and Mack, **1994,** p. 5). Each group of experts is asked to assume the role of typical users. The walkthroughs are then done by following a sequence of steps (Bias, **1994):**

1. Scenarios are developed in the form of a series of hard-copy screens representing a single path through the interface. Often just two or a few screens are developed.
2. The scenarios are presented to the panel of evaluators and the panelists are asked to write down the sequence of actions they would take to move from one screen to another. They do this individually without conferring with one another.
3. When everyone has written down their actions, the panelists discuss the actions that they suggested for that round of the review. Usually, the representative users go first so that they are not influenced by the other panel members and are not deterred from speaking. Then the usability experts present their findings, and finally the developers offer their comments.
4. Then the panel moves on to the next round of screens. This process continues until all the scenarios have been evaluated.

The benefits of pluralistic walkthroughs include a strong focus on users' tasks. Performance data is produced and many designers like the apparent clarity of working with quantitative data. The approach also lends itself well to participatory design practices by involving a multidisciplinary team in which users play a key role. Limitations include having to get all the experts together at once and then proceed at the rate of the slowest. Furthermore, only a limited number of scenarios, and hence paths through the interface, can usually be explored because of time constraints.

Assignment

This assignment continues the work you did on the web-based ticketing system at the end of Chapters 7 and 8. The aim of this assignment is to evaluate the prototypes produced in the assignment of Chapter 8. The assignment takes an iterative form in which we ask you to evaluate and redesign your prototypes, following the iterative path in the interaction design process described in Chapter 6.

(a) For each prototype, return to the feedback you collected in Chapter 8 but this time perform open-ended interviews with a couple of potential users.

(b) Based on the feedback from this first evaluation, redesign the software/HTML prototype to take comments on all three prototypes into account.

(c) Decide on an appropriate set of heuristics and perform a heuristic evaluation of the redesigned prototype.

(d) Based on this evaluation, redesign the prototype to overcome the problems you encountered.

(e) Design a questionnaire to evaluate the system. The questionnaire may be paper-based or electronic. If it is electronic, make your software prototype and the questionnaire available to others and ask a selection of people to evaluate the system.

Summary

Techniques for asking users for their opinions vary from being unstructured and open-ended to tightly structured. The former enable exploration of concepts, while the latter provide structured information and can be replicated with large numbers of users, as in surveys. Predictive evaluation is done by experts who inspect the designs and offer their opinions. The value of these techniques is that they structure the evaluation process, which can in turn help to prevent problems from being overlooked. In practice, interviews and observations often go hand in hand, as part of a design process.

Key points

- There are three styles of interviews: structured, semi-structured and unstructured.
- Interview questions can be open or closed. Closed questions require the interviewee to select from a limited range of options. Open questions accept a free-range response.
- Many interviews are semi-structured. The evaluator has a predetermined agenda but will probe and follow interesting, relevant directions suggested by the interviewee. A few structured questions may also be included, for example to collect demographic information.
- Structured and semi-structured interviews are designed to be replicated.
- Focus groups are a form of group interview.
- Questionnaires are a comparatively low-cost, quick way of reaching large numbers of people.
- Various rating scales exist including selection boxes, Likert, and semantic scales.
- Inspections can be used for evaluating requirements, mockups, functional prototypes, or systems.
- Five experts typically find around 75% of the usability problems.
- Compared to user testing, heuristic evaluation is less expensive and more flexible.
- User testing and heuristic evaluation often reveal different usability problems.
- Other types of inspections include pluralistic and cognitive walkthroughs.
- Walkthroughs are very focused and so are suitable for evaluating small parts of systems.

Further reading

NIELSEN, J., AND MACK, R. L. (eds.) (1994) ***Usability Inspection Methods.*** New York: John Wiley & Sons. This book contains an edited collection of chapters on a variety of usability inspection methods. There is a detailed description of heuristic evaluation and walkthroughs and comparisons of these techniques with other evaluation techniques, particularly user testing. Jakob Nielsen's website *useit.com* provides additional information and advice on website design.

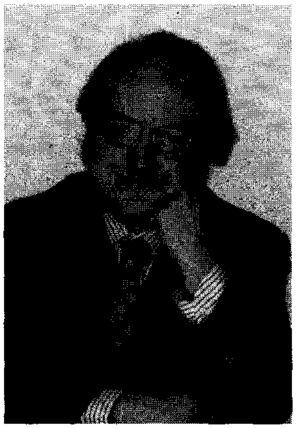
OPPENHEIM, A. N. (1992) ***Questionnaire Design, Interviewing and Attitude Measurement.*** London: Pinter Publishers. This text is useful for reference. It provides a detailed account of all aspects of questionnaire design, illustrated with many examples.

PREECE, J. ***(2000) Online Communities: Designing Usability, Supporting Sociability.*** Chichester, UK: John Wiley & Sons. This book is about the design of web-based online communities. It suggests guidelines for evaluating for sociability and usability that can be used as a basis for heuristics.

ROBSON, C. (1993) ***Real World Research.*** Blackwell. Oxford, UK. Chapter 9 provides basic practical guidance on how to interview and design questionnaires. It also contains many examples.

SHNEIDERMAN, B. (1998) ***Designing the User Interface: Strategies for Effective Human-Computer Interaction (3rd Edition)*** Reading, ***MA.:*** Addison-Wesley. Chapter 4 contains a discussion of the QUIS questionnaire.

INTERVIEW with Jakob Nielsen



Jakob Nielsen is a pioneer of heuristic evaluation. He is currently principal of the Nielsen Norman Consultancy Group and the author of numerous articles and books, including his recent book, *Designing Web Usability* (New Riders Publishing). He is well-known for his regular sound bites on usability which for many years have appeared at useit.com. In this interview Jakob talks about heuristic evaluation, why he developed the technique, and how it can be applied to the web.

JP: Jakob, why did you create heuristic evaluation?

JN: It is part of a larger mission I was on in the mid-'80s, which was to simplify usability engineering, to get more people using what I call "discount usability engineering." The idea was to come up with several simplified methods that would be very easy and fast to use. Heuristic evaluation can be used for any design project or any stage in the design process, without budgetary constraints. To succeed it had to be fast, cheap, and useful.

JP: How can it be adapted for the web?

JN: I think it applies just as much to the web, actually if anything more, because a typical website will have tens of thousands of pages. A big one may have hundreds of thousands of pages, much too much to be assessed using traditional usability evaluation methods such as user testing. User testing is good for testing the home page or the main navigation system. But if you look at the individual pages, there is no way that you can really test them. Even with the discount approach, which would involve five users, it would still be hard to test all the pages. So all you are left with is the notion of doing a heuristic evaluation, where you just have a few people look at the majority of pages and judge them according to the heuristics. Now the heuristics are somewhat different, because people behave differently on the web. They are more ruthless about getting a very quick glance at what is on a page and if they don't understand it then leaving it. Typically application users work a little harder at learning an application. The basic heuristics that I developed a long time ago are universal, so they apply to the web as well. But as well as these global heuristics that are always true, for example "consistency," there can be specialized heuristics that apply to particular systems. But most evaluators use the general heuristics because the web is still evolving and we are still in the process of determining what the web-specific heuristics should be.

JP: So how do you advise designers to go about evaluating a really large website?

JN: Well, you cannot actually test every page. Also, there is another problem: developing a large website is incredibly collaborative and involves a lot of different people. There may be a central team in charge of things like the home page, the overall appearance, and the overall navigation system. But when it comes to making a product page, it is the product-marketing manager of, say, Kentucky who is in charge of that. The division in Kentucky knows about the product line and the people back at headquarters have no clue about the details. That's why they have to do their own evaluations in that department. The big thing right now is that this is not being done, developers are not evaluating enough. That's one of the reasons I want to push the heuristic evaluation method even further to get it out to all the website contributors. The uptake of usability methods has dramatically improved from five years ago, when many companies didn't have a clue, but the need today is still great because of the phenomenal development of the web.

JP: When should you start doing heuristic evaluation?

JN: You should start quite early, maybe not quite as early as testing a very rough mockup, but as soon as there is a slightly more substantial prototype. For example, if you are building a website that might eventually have ten thousand pages, it would be appropriate to do a heuristic evaluation of, say, the first ten to twenty pages. By doing this you would catch quite a lot of usability problems.

JP: How do you combine user testing and heuristic evaluation?

JN: I suggest a sandwich model where you layer them on top of each other. Do some early user testing of two or three drawings. Develop the ideas somewhat, then do a heuristic evaluation. Then evolve the design further, do some user tests, evolve it and do heuristic evaluation, and so on. When the design is nearing completion, heuristic evaluation is very useful particularly for a very large design.

JP: So, do you have a story to tell us about your consulting experiences, something that opened your eyes or amused you?

JN: Well, my most interesting project started when I received an email from a co-founder of a large company who wanted my opinion on a new idea. We met and he explained his idea and because I know a lot about usability, including research studies, I could warn him that it wouldn't work—it was doomed. This was very satisfying and seems like the true role for a usability consultant. I think usability consultants should have this level of insight. It is not enough to just clean up after somebody makes the mistake of starting the wrong project or produces a poor design. We really should help define which projects should be done in the first place. Our role is to help identify options for really improving people's lives, for developing products that are considerably more efficient, easier or faster to learn, or whatever the criteria are. That is the ultimate goal of our entire field.

JP: One last question—how do you think the web will develop? What will we see next, what do you expect the future to bring?

JN: I hope we will abandon the page metaphor and reach back to the earlier days of hypertext. There are other ideas that would help people navigate the web better. The web is really an "article-reading" interface. My website useit.com, for example, is mainly articles, but for many other things people need a different interface, the current interface just does not work. I hope we will evolve a more interesting, useful interface that I'll call the "Internet desktop," which would have a control panel for your own environment, or another metaphor would be "your personal secretary." Instead of the old goal where the computer spits out more information, the goal would be for the computer to protect you from too much information. You shouldn't have to actually go and read all those webpages. You should have something that would help you prioritize your time so you would get the most out of the web. But, pragmatically speaking, these are not going to come any time soon. My prediction has been that Explorer Version 8 will be the first good web browser and that is still my prediction, but there are still a few versions to come before we reach that level. The more short-term prediction is really that designers will take much more responsibility for content and usability of the web. We need to write webpages so that people can read them. For instance, we need headlines that make sense. Even something as simple as a headline is a user interface, because it's now being used interactively, not as in a magazine where you just look at it. So writing the headline, writing the content, designing the navigation are jobs for the individual website designers. In combination, such decisions are really defining the user experience of the network economy. That's why we really have an obligation, every one of us, because we are building the new world and if the new world turns out to be miserable, we have only ourselves to blame, not Bill Gates. We've got to design the web for the way users behave.

Chapter 14

Testing and modeling users

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

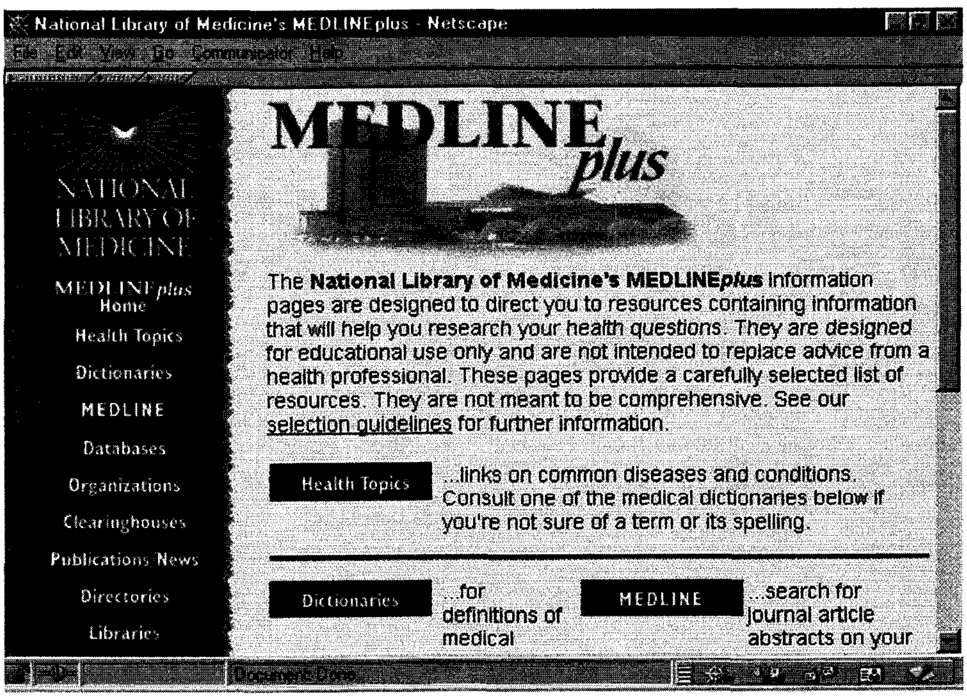


Figure 14.1 Home page of MEDLINEplus.

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 confidential in 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

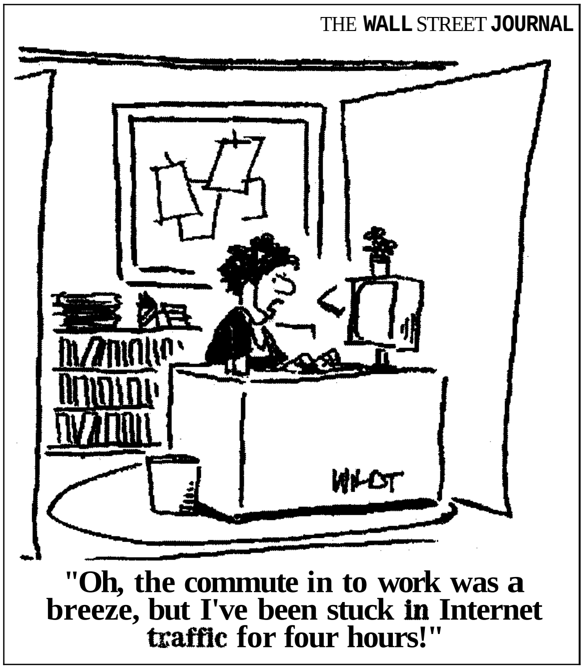
(a) This way of selecting participants was appropriate for user testing. The evaluator was careful to get a number of representative users across the user age range from both genders. Participants were screened to ensure that they were experienced web users. The evaluator decided to select from a local 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).

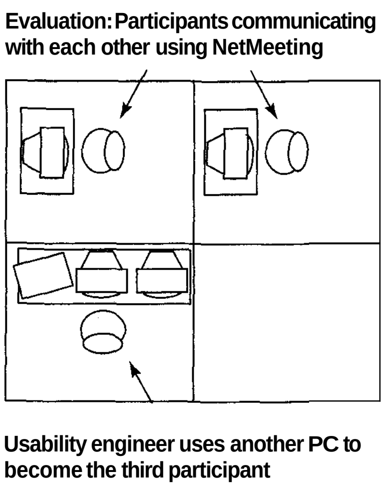


Figure 14.8 The testing arrangement used for NetMeeting videoconferencing system.

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 × 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 (see)
K	Pressing a single key or button Skilled typist (55 wpm) Average typist (40 wpm) User unfamiliar with the keyboard Pressing shift or control key	0.35 (average) 0.22 0.28 1.20 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:

$$T = k \log2(D/S + 0.5),\ k - 100 \text{ msec.}$$

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.

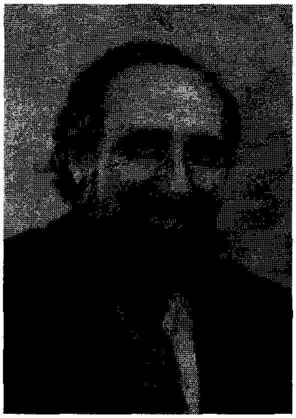
ROBSON, C. **(1994)** ***Experimental Design and Statistics in Psychology.*** Aylesbury, UK. Penguin Psychology. This book provides an introduction to experimental design and basic statistics.

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."

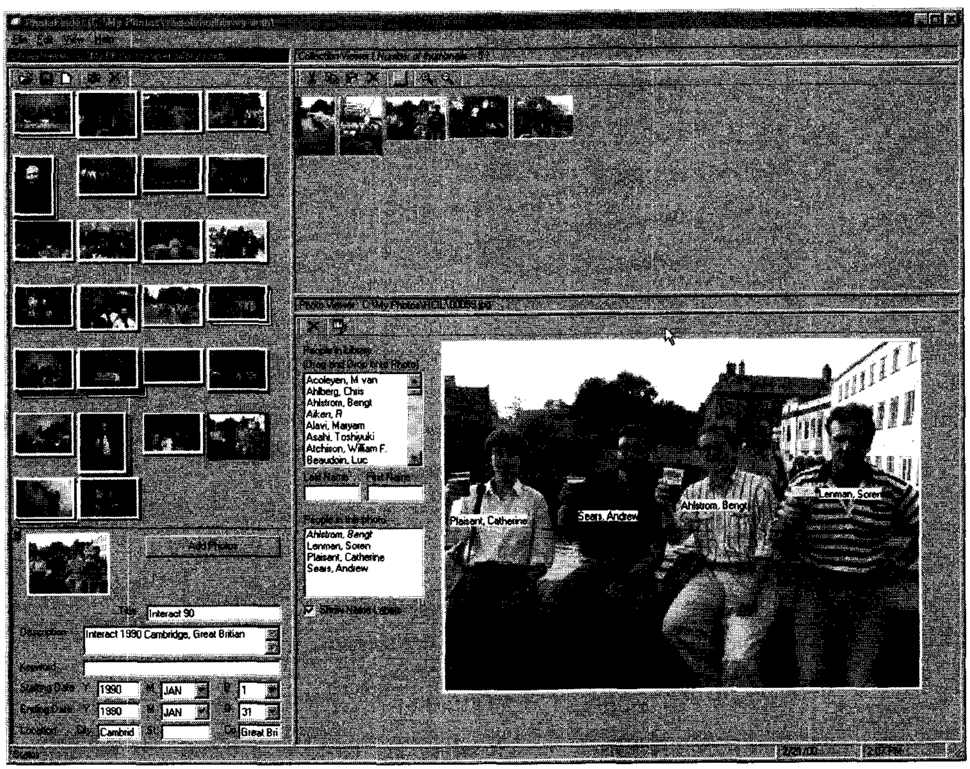
By contrast, a usability test studies a small number of users who carry out required tasks. Statistical results are less important. The goal is to refine a product as quickly as possible. The outcome of a usability test is a report to developers that identifies frequent problems and possibly suggests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

Chapter 15

Design and evaluation in the real world: communicators and advisory systems

15.1 Introduction

Textbooks about design and usability testing often make the processes sound straightforward and able to be followed in a step-by-step manner. However, in the real world bringing together all the different aspects of a design is far from straightforward. It is only when you become involved in an actual design project that the challenges and multitude of difficult decisions to be made become apparent. Iterative design often involves carrying out different parts of a project in parallel and under tremendous pressure. The need to deal with different sets of demands and trade-offs (e.g., the need for rigorous testing versus the very limited availability of time and resources) is a major influence on the way a design project is carried out.

The aim of this final chapter is to convey what interaction design is like in the real world by describing how others have dealt with the challenges of an actual design project. As you will have noticed, we have written primarily about design in Chapters 6–9 and evaluation in Chapters 10–14. This was to enable us to explain the different techniques and processes involved during a design project. It is important to realize that in the real world these two central aspects are closely integrated. You do not do one without the other. In particular, the main reason for doing an evaluation is to make progress on a design. Conversely, whenever you develop a design you need to evaluate it. Whether you are designing a small handheld device or a large air-traffic control system, a design that takes months to produce or one that spans years of effort, the two processes must be carried out together.

The chapter provides glimpses into the design and evaluation process for quite different types of interactive systems. The first two case studies discuss the design of mobile communicators for different groups of users, showing how the design issues differ for each group. The third case study examines the redesign of a large interactive voice response system. In the original design, the focus was on developing a system where the programmers used themselves as models of the users. Furthermore, the programmers were more concerned with developing elegant programs than with users' needs for easy interaction. As you will see, this caused a mismatch between their design and how users tried to find information. This is a common predicament and interaction designers are often brought in to fix already badly designed systems.

The main aims of this chapter are to:

- Show how design and evaluation are brought together in the development of interactive products.
- Show how different combinations of design and evaluation methods are used in practice.
- Describe the various design trade-offs and decisions made in the real world.

15.2 Key issues

As we have stressed throughout, user-centered approaches to interaction design involve iterative cycles of design-evaluate-redesign as development progresses from initial ideas through various prototypes to the final product. How many cycles need to take place depends on the constraints of the project (e.g., how many people are working on it, how much time is available, how secure the system has to be). To be good at working through these cycles requires a mix of skills involving multitasking, decision-making, team work and firefighting. Many practical issues and unexpected events also need to be dealt with (e.g., users not turning up at testing sessions, prototypes not working, budgets being cut, time to completion being reduced, designers leaving at crucial stages). A design team, therefore, must be creative, well organized, and knowledgeable about the range of techniques that can be brought into play when needed. ***Part*** of ***the challenge and*** excitement of interaction design is finding ways to cope with the diverse set of problems confronting a project.

A multitude of questions, concerns and decisions come up throughout a design project. No two projects are ever the same; each will face a different set of constraints, demands, and crises. Throughout the book we have raised what we consider to be general issues that are important in any project. These include how to involve users and take their needs into account, how to understand a problem space, how to design a conceptual model, and how to go about designing and evaluating interfaces. In the following case studies, we focus on some of the more practical problems and dilemmas that can arise when working on an actual project.

We present the case studies through a set of questions that draw out a number of key issues for each project. For example, mapping a large number of functions onto a much smaller number of buttons is key for mobile devices; understanding a child's world is key when designing for children; evaluating the current system is key when redesigning any large system.

15.3 Designing mobile communicators

The first two case studies are about the design of mobile communicators. They focus on some of the design decisions and trade-offs that need to be made. We describe example design practices at two companies, Nokia and Philips, highlighting the differences in requirements and design methods for what is seemingly a similar device.

15.3.1 Background

Mobile communicators often combine the functionality of a mobile telephone, a PDA, and a desktop computer. They allow the user to send and receive email and faxes, to make and receive telephone calls, and to keep contact details, diary entries, and other notes. They are an example of new devices that try to push technological boundaries while at the same time being accessible to a wide range of users. A key design challenge, therefore, is how to make such everyday devices usable and affordable to a heterogeneous set of users. Related to this set of usability goals is the decision about which design approach to use. As you are aware, there are many different approaches to choose from, ranging from ethnographic to more analytic methods. Here, we examine the different approaches of the two companies. To put you in a "design" frame of mind, we begin by asking you to consider the requirements for this kind of device.

ACTIVITY 15.1 In Chapter 7, we introduced a number of different kinds of requirements: functional, data, environmental, user, and usability requirements. Which of these is particularly relevant to the design of a mobile communicator?

Comment All these are relevant in the design of mobile communicators, but one that needs particular attention is environmental requirements. Because the device is aimed at users "on the move" in all kinds of places, the environment in which it should work or its "context of use" is very variable.

Core environmental issues include how to make the device small and light enough to be carried around in a pocket or small handbag. This means the device must be made of light materials and should be physically small, and also the software must be designed to work with a small screen and limited memory. The system must allow for a whole range of situations: noisy or quiet, well lit or poorly lit, hot or cold, wet or dry, vibrating or still, and so on. These constraints have implications for the use of audio, for the levels of display lighting, and for the physical robustness of the device, among other things.

Another consideration in the design of this kind of communication device is what the users are doing when using it. A typical user is likely to be doing something else at the same time as using the communicator. This may be walking around, avoiding obstacles, looking for traffic, etc., or it may be listening for a train announcement or a call from children. So users are trying to combine at least three things: communicating with the device (talking, typing, or whatever), performing the "external" activity (walking, listening, etc.), and operating the device. This creates quite a high cognitive load, so operating the device should occupy as little attention as possible.

Tasks are very likely to be interrupted by external events, so users need to know where in an interaction sequence they are at any time, and be able to restart the sequence after an interruption. For a mobile communicator designed to access the Internet, this raises an interesting design trade-off: how long should a communicator remain connected to the Internet after activity has apparently ceased? A balance is needed between disconnecting so as to minimize connection costs, and remaining connected in a stable state to allow the resumption of an interrupted task. The best option may be to let users set their own time-out period, but this adds to the complexity of operation.

Another implication of the fact that users are likely to be doing other things in parallel with operating the device is that the communicator may need to be operated with one hand, or indeed in a hands-free mode. For example, someone who is walking down the street carrying a bag when the phone rings needs to be able to respond without stopping and putting the bag down, i.e., the operation needs to be one-handed.

For mobile devices in particular, tasks tend to be time-critical, ad *hoc*, triggered by other people or events, relatively brief, low in terms of attention to be applied to the task, and very personal. Because of these characteristics, the flow among tasks must be smooth. It seems that easy transition between contact database, telephone, and calendar is particularly important for mobile devices. The nature of these tasks and the environmental requirements for mobile devices have implications for evaluation, as we discuss in section 15.3.2.

Because this device will be mobile it must be simple to use and not involve much training. It also needs to be robust and reliable, as the user is most likely to be away from any significant technical support.

15.3.2 Nokia's approach to developing a communicator

So how does Nokia deal with these kinds of requirements? And which design and evaluation methods do they use? Here, we look at an example approach of Nokia's, and some of the key decisions in mobile communicator design. **A** design example of an existing Nokia communicator is illustrated in Figure 15.1. This communicator weighs 244 g, is $158 \times 56 \times 27$ mm, and has a full-color screen. As well

Figure 15.1 The Nokia 9210 communicator.

as email and high-speed WAP connections, it also runs a variety of office applications including word processing, spreadsheets, and presentations.[1]

This case study is based on material from Vaananen-Vainio-Mattila and Ruuska (2000).

What kind of lifecycle does Nokia use? Nokia follows a user-centered approach to concept development that includes contextual design techniques. They point out that "one clear strength of the methodology is that it makes ethnographic research manageable in a business environment" (Vaananen-Vainio-Mattila and Ruuska, 2000, p. 197). As discussed in Chapter 9, the "rich" descriptions arising from an ethnographic study are often not in a form that can be readily translated into a design specification. Nokia tries to get around this problem by carrying out ethnographic studies in combination with other methods. This enables them to come up with a set of detailed requirements.

Figure 15.2 shows a top-level model of Nokia's approach. It has four main steps:

1. The cycle begins with data gathering. The data is collected through market research studies, data from previous projects, and contextual techniques.

[1]Description summarized from information on the Nokia website www.nokia.com, as of February 2001.

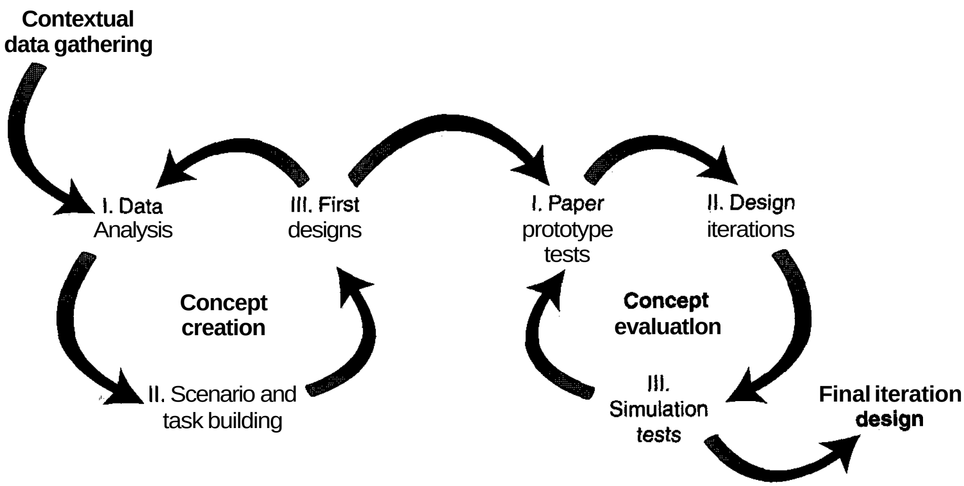


Figure 15.2 The user-centered concept and product development cycle.

2. Scenarios and then task models are built by analyzing the data collected, and initial designs are proposed.
3. Many iterations of design and evaluation are performed before the final design emerges. During this process, it may be found that more data is required, so further data gathering is conducted. The evaluation involves contextual interviews with paper-based prototypes to get feedback on first designs, and usability testing once the design is sufficiently advanced. Evaluation sessions emphasize the most important user tasks, as determined by the data gathering.
   - Once the design is advanced enough, high-fidelity simulations of the design are constructed.
   - Simulation tests are conducted with end users, and expert reviews are performed. Functional prototypes are tested with end users for feedback on long-term acceptability, efficiency, and utility of the concept.
4. During the last iteration phase, the final design is tested with end users and expert usability specialists.

ACTIVITY 15.2 How does this cycle of activities differ from the interaction design model introduced in Figure 6.7?

Comment This cycle also has a focus on iteration through prototyping and evaluation, which is the basis of the model in Chapter 6. However, this cycle distinguishes between concept creation and concept evaluation. scenarios and task modeling are used at the concept creation phase but simulation tests are used in the concept evaluation phase.

What challenges does this approach raise? Nokia is very conscious of the need for iterative design and evaluation in the development of mobile communicators. They also use participatory design to a degree, but they point out that users will not necessarily have the vision of future possibilities that would allow innovative design in the same way as they might if asked to help design a familiar application like a web browser. Nokia is also well aware of the challenges of evaluating an innovative product like a communicator. These include:

- The difficulty of testing in all possible scenarios.
- The difficulty of testing human communication practices, especially when developing innovative products that will encourage novel behavior.
- The difficulty of testing services that cannot all be known beforehand.

What happens when the product is new and there are no users to test? At Nokia, quick and effortless access to critical tasks is a key design driver, and usability tests are used to evaluate the flow of tasks that have been found critical for mobile devices.

In a competitive and innovative market, other evaluation challenges may also arise. For example, consider the original Nokia communicator (the N9000). This was the first of its kind on the market. This had implications for how it could be evaluated because the device could not be shown to people outside the development team for fear of losing the "first-in-the-market" advantage. Thus the first version on the market did not have the benefit of testing with real users. Although extensive paper-based prototyping and simulations were produced, the evaluations were limited to a small group of people.

What methods does Nokia use? Nokia uses a number of methods in its development cycle, in particular "usage scenarios." Usage scenarios are high-level descriptions of uses of the device, based on data collected from representative stakeholders. They differ from the generic scenarios described in Chapter 7 in that they focus specifically on concept creation and high-level design considerations. An example of a usage scenario developed by Nokia is given in Figure 15.3.

What do design teams do next once they have created a set of scenarios? At Nokia, the design teams use the usage scenarios they have developed to identify critical user tasks and their structure. These task descriptions, which are more detailed than the original descriptions provided in the usage scenarios, are then used to consider lower-level design issues. A sample critical user task is shown in Figure 15.4.

ACTIVITY 15.3 To create scenarios, appropriate tasks and stakeholders will need to be identified. Who would the stakeholders be, and what techniques might be used to investigate their needs?

Comment First, the tasks to be performed and the stakeholders who might be asked about requirements would have to be identified. Stakeholders for a mobile device include users, developers, telephone companies, computer hardware and software vendors, and their shareholders. At least in theory, a user may be almost any member of the population, but in practice, only certain sections of the population are likely to be users. Given the wide functionality of the communicator, the most likely users are professionals.

Example of a Usage Scenario

David works as a legal consultant in an international corporation. He uses a communicator daily for light note taking and communications as well as for his personal organization.

8 A.M. The working day starts with a multiparty conference call to Japan. He uses the communicator as a speakerphone to be able to type notes in it at the same time. At the end of the meeting, he sends everybody a copy of the notes via email directly from the communicator.

1 P.M. At the airport, he downloads all his new email messages to his communicator so that he can start working on them during the flight. On the plane there is always plenty of time to write answers to messages. While downloading, he views the communicator calendar for the day and remembers having promised to send his business card to a potential client. He does this while standing in line for boarding.

At his destination, he switches the communicator phone on, and it automatically starts sending the replies written on the plane. At the same time David can continue reading the rest of the messages.

2:30 P.M. His secretary back in London sends him a calendar reservation for the following week. David checks his calendar in the communicator and accepts the request. His communicator sends the confirmation automatically to the secretary and marks the appointment in David's calendar.

Figure 15.3 An example usage scenario.

If we assume that the user group is professional, then it is necessary to find out more about the tasks they perform. This could be done using questionnaires, interviews and observation, or focus groups, but there would be some other issues to consider. **A** professional who is constantly on the move will be difficult to track down. However, interviews and questionnaires can be administered in different settings such as at trade fairs where many professionals are all gathered in one place. This would potentially provide a ready audience, reduce travel expenses, and supply immediate responses.

Performing standard observations in an office has its problems, but observing someone on the move, in all the possible locations in which they might use the device, opens up a whole new set of issues. Mobile devices are intended to be used anywhere, so where are observations performed, and how closely can the participants be followed?

What usability and user experience goals are important in designing this kind of device? A mobile communicator would be expected to meet the normal usability goals that we have discussed before. But what about user experience goals? Personalization has been identified as significant in user satisfaction; however, a balance

User Tasks: Classification

(1) Done under pressure: very critical
(2) Done frequently: critical
(3) Medium frequency or medium pressure
(4) Not frequent or not done under pressure

Sample 1: User tasks in person-to-person voice communication

Call-making/in-call

(1) Making a call to an emergency number
(1) Answering a call
(1) Rejecting a call
(2) Making a call to frequently called numbers (usually 4–10 of them)
(2) Making a call by manually entering each digit
(2) Redialing a number/person
(2) Indication of being busy
(3) Making a call to semifrequently called numbers (e.g., a vet, hairdresser)
(4) Making a call to occasionally called numbers (i.e., numbers that are often called only once).

Phone book memory

(1/4) Saving a name and number [1 = very critical during a call]
(213) Recalling a name and number and dialing [2 = to a frequently called number]
(4) Editing a name and number
(4) Erasing a name and number
(4) Browsing the contents of a phone book, etc.

Sample 2: User tasks in text messaging

Sending

(4) Sending a text message to a contact in the phone book
(4) Setting a message center number, etc.

Receiving

(2) Reading and replying to a message
(2) Reading and calling back the sender
(3) Reading and erasing a message
(4) Reading and storing a message with a new name, etc.

Figure 15.4 Sample user tasks.

BOX 15.1 Designing an Interface with a Small Number of Keys

What would you do if you had to design a communication device that could accommodate a maximum of only 15 keys? The device has to support numerical and text input. How could you design the mapping of the 15 keys to the various kinds of operations proposed so as to support the range of user tasks identified?

As a minimum, the device will need an on/off switch, a switch for connecting and disconnecting to the network, and a mechanism for entering ten numbers, 26 characters, and the space. You may decide to omit punctuation and capital letters, although this will have implications for the usability of the device. One way in which the functionality can be achieved is described below.

One key is a dedicated on/off switch, one key is a dedicated connect/disconnect key, and one function key toggles the keys from numerical to character input. Ten of the keys represent the digits 0–9 and two or three characters each (26 characters plus the space bar means that seven of the keys must represent three characters and three of them must represent two). This uses 13 keys, which means that the keys for on/off, connect/disconnect, and the function button can be made larger to distinguish them from the others.

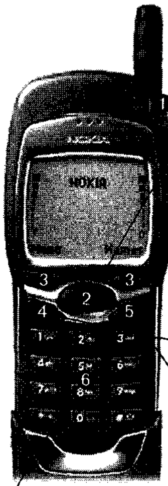
Alternatively, if you want to include punctuation, then all ten keys could have three characters each (giving room for four punctuation marks), and if you want to include capitals then a 14th key might be used as a "shift" key. Remember, though, that if the device is to be operated with one hand then this must operate more like a "caps lock" key than a shift key.

Some devices let you choose a character using only two keys. One key is repeatedly pressed as it toggles through the character list, and the second is used to accept the choice when the required character is displayed. This design choice requires more time from the user and its suitability depends on the functions the device is to support.

must be struck between allowing flexibility and providing sensible default values so that users don't have to customize settings unless they want to.

Mobile communicators are intended to support users wherever they are, so they must be compatible with the users' lifestyles. Designers must therefore understand the design characteristics that make the communicator attractive to different user groups, and those characteristics that will vary from group to group. If we consider the users as business people, then the important user experience goals are likely to include being helpful, motivating, aesthetically pleasing, and rewarding. If we consider children, then entertainment and fun are likely to be more important, while for teenagers its physical appearance might be more significant.

How does Nokia design a communicator's physical aspects? Deciding how many keys to have and how to map them onto a much larger set of functions is a difficult design challenge in any mobile device (see Box 15.1). For example, in the Nokia 7110 mobile phone, the problem of limited keys and limited space was dealt with by providing softkeys with context-sensitive functions that change depending on where the user is in the interaction sequence. This allows the keys to perform different functions depending on the other contextual issues. The softkeys allow the user to do a variety of things, such as make selections, enter, edit, or delete **text.** The current label for each softkey is displayed at the bottom of the screen, near the relevant key. There is,



1. Power key. Used for switching the phone on and off. When pressed briefly the user enters the list of profiles (user environments: e.g., Silent to turn off all the phone tones).

2. Navi Roller. Used for navigating the Menu and the Phonebook. Navi Roller allows scrolling up and down as well as selecting, saving, or sending the displayed item by clicking the roller.

3. Two **Softkeys**. The softkeys are assigned actions that enable the user to manipulate the user interface by making selections and entering, editing, and deleting text. The name of the action changes according to the state of the phone. Descriptive labels are shown in the lower corner of the display respective to the key underneath.

4. Send key (green receiver). Send key is used for call handling, that is, call creation, and also for bringing up the last-called numbers list.

5. End key (red receiver). End key is for call termination. It is also an Exit key that can be used as a panic key since it takes the user from any state of the phone to the idle state without saving changes.

6. Numeric keys, with an alphabet according to the ITU-T.161 standard. Used for number and character input. The 1 key also doubles as the Voice Mailbox speed dial key. The # key is used for changing the character case during editing. Nokia 7110 employs a predictive text input method: only one keypress per letter is required, and the entered text string is continually matched with the words in the built-in dictionary.

- The left softkey is basically used as a yes/positive key. It contains options that execute commands and go deeper into the menu structure. In the idle state the left softkey is Menu (the hierarchy of phone functions).
- The right softkey is basically used as a no/negative key. It contains options that cancel commands, delete text, and go higher in the menu structure. In the idle state the right softkey is Names (the Phonebook).

Figure 15.5 **The Nokia 7110 mobile phone.**

of course, a balance to be struck between having too many softkeys, each with limited functionality, and having only a few keys that can be overloaded with too many functions. In the end, the Nokia **7110** (Figure **15.5)** was designed with just two softkeys that performed multiple functions. (Vaananen-Vainio-Mattila and Ruuska, 2000).

Textual input becomes a major problem when the number of input keys is restricted by the design. Having only a small number means the users must constantly "peck" at a few keys, typically using their thumbs. Trying to place too many keys in a heavily constrained space means that the user is likely to press the wrong key or two keys at once. How was this problem handled by Nokia? They opted for a small number of keys but in combination with a way of speeding up the typing of words, through having the communicator guess what the user is writing. In particular, the Nokia **7110** introduced the T9 predictive text method that allows speedy input of words based on a dictionary. The phone proposes a likely word once the user has typed a few characters. The user then either selects the proposed word **and** moves on to the next word, or rejects it and continues to enter the current word.

Communicators have also been designed to include a function button to let the user customize the interface to a limited degree, for example by allowing a favorite application to be associated with one of the hard keys.

BOX 15.2 Designing Telephones for the Elderly and Disabled

The British Royal National Institute for the Blind (RNIB), together with the British Department of Trade and Industry and British Telecommunications, have compiled a brochure to explain the different impairments affecting many telephone user groups, together with a set of suggested telephone features that could greatly enhance the accessibility of devices for such user groups. They identify 15 impairments and 44 features that could be added to telephones to make their use more pleasant. The impairments include cognitive impairment, weak grip, limited dexterity, speech impairment, hearing impairment, and hand tremor (Gill and Shipley, 1999). Features that could make a difference to these user groups include:

- Guarded or recessed keys to help prevent pressing the wrong key by mistake.
- Sidetone reduction, which reduces the amount of noise picked up from the environment and mixed with incoming speech at the earpiece.
- Allowing the user to adjust the amount of pressure needed to select a key. Apart from the more obvious consequences of too much or too little pressure, unsuitable key pressure may produce muscle spasms in some users.
- Audio and tactile key feedback to indicate when a key has been pressed.

Is it possible to design consistent interfaces, given the physical constraints of a communicator? A particular problem when developing software for a small display with limited input controls is how to make the interface consistent.

The design dilemma of consistency was addressed in Chapter 1. Consistency is often extolled as a virtue, yet it is sometimes appropriate to be inconsistent. In the design of communicators, the problems of consistency arise again. The device needs to have external consistency, i.e., consistency with users' expectations from their use of other similar tools, and also internal consistency, i.e., consistency with other items of software that the device supports. Sometimes these two design goals are in conflict, and it is appropriate to design a new solution for a particular situation.

The N9000 web browser was developed for the Nokia N9000 communicator. Many design decisions had to be dealt with, especially the problem of consistency (Ketola et al., 2000). Nokia has an internal style guide that all its products must follow in order to maintain internal consistency. External consistency with PC-based products is difficult to achieve because of physical constraints, and because the operating system for the N9000 is not commonly used with a PC. Other constraints on the design were:

1. The N9000 does not have a pointing device. Pointing is therefore done by selection using the scrolling bars. Scrolling down causes selection to jump from one hyperlink to the next; scrolling up causes it to jump to the previous link.
2. In cellular devices, connection rate is limited to 9600 bps, which is slower than the fixed-line rate. Connection can also take up to 30 seconds, considerably slower than the fixed-line equivalent. Web users may be accustomed to slow downloading times, but a long connection time is a new

phenomenon. A progress indicator was included in the design so that users would not become frustrated and start pressing other buttons. This leads to a further external consistency issue: should web pages be made to look the same as on faster desktop machines, or should they be designed for faster downloading?

Specific design decisions and solutions taken under these constraints were as follows:

1. The default page for a desktop web browser is a home page, but because of the connection time and the speed of downloading, the N9000 browser defaults to a list of favorite pages (called the Hotlist) instead. Thus, the default state is offline. This violates external consistency, but proved to be acceptable to users.
2. The functionality of the N9000 browser had to be carefully examined. Because of the Nokia style guide, only three buttons were available for navigating through the function hierarchy, so navigation became a major issue. To cope with the limited availability of command buttons, the N9000 employs the idea of **views,** within which only certain functions are possible. For the web browser, three views were provided: Hotlist view, Document view, and Navigation view. Users can select a document in the Hotlist view and enter the Document view. From here they are able to save, read, disconnect from the network, and close the document. However, they cannot navigate through the document. For this they need to go to the Navigation view. This conceptual shift was difficult for users to come to terms with.
3. The style guide dictated that the fourth command button be used to move upwards in the view hierarchy. It is also a part of the style guide that this button should be called "Back." In other applications this may not be a problem, but in the context of a web browser, a button labeled "Back" is interpreted differently. Internal consistency had to be obeyed here, and so the command that moved back to the previous page in the history list was called "Previous." This caused considerable confusion for users.
4. Optimizing web pages for display on mobile communicators involves the following three issues: content, because it's important to optimize download times; page layout, because of the small size of the screen; and navigation, because it's important to minimize the number of file downloads. User trials showed that, in the mobile context, users are more interested in getting the text information quickly than in downloading the graphics. Downloading unwanted pages also proved to be a key aspect of usability. Good link naming and clear, predictable behavior were important because of the long downloading times; locating the wrong page expends much time and cost.

ACTIVITY 15.4 **If you are sitting near a desktop computer, study the interface of the piece of software that is running. If you are not near one, then think of the application you run most regularly on a** desktop machine. Imagine what this interface would look like if you were to reduce the screen size to a mere 158 mm × 56 mm (the size of the Nokia 9210 communicator). What difficulties can you see? What implications do you think this has for software design, and also for the user who is swapping between desktop systems and mobile systems on a regular basis?

Comment If the same screen design is carried over to the mobile device then either everything will have to be miniaturized, so that the tool bars, icons and menus will become unreadable, or left at the same size, so that they will take up too much space on the screen. The interface therefore must be designed differently. This has implications for consistency for users who might be using the same application in a desktop environment and on the mobile device.

What kind of user testing does Nokia use? As mentioned earlier, there were confidentiality problems in testing the first generation of communicators on the intended user population. Hence, user testing could be done only after the product was released on the market. One kind of summative testing Nokia did was to find out what questions people have when first using the communicator. Users were given the device to use for some weeks and were then asked to report on positive and negative features. The results from this study confirmed the developers' concerns about the effects of consistency with other similar applications designed to run on desktop machines. Another study involved sending questionnaires to more critical communicator users whose experience ranged from 0 to 12 months, to find out if their reactions were similar.

As can be seen from this case study, Nokia uses a number of methods to develop their communicators for the general public. Furthermore, many design decisions and problems have to be dealt with, ranging from the lack of real users for testing, to how to let users send text messages with only a few keys and a very confined space.

15.3.3 Philips' approach to designing a communicator for children

We now consider how another company went about designing a mobile communicator aimed at a specific user group, children (mostly girls) aged between 7 and 12. Developing a tool for this user group is quite different from developing a tool for use by the general public, where there is likely to be a huge range of different users. An advantage of designing a device for a smaller set of users is that they are likely to have similar needs and preferences, meaning that the device can be customized much more to their requirements. This case study draws on material reported in Oosterholt et al. (1996).

Which approach did Philips use? The Philips process of development for this particular communicator made extensive use of prototyping techniques and participatory design. Children were involved from the initial concepts stage right through to final product testing. Each time a prototype was produced, it was shown to children for comment and feedback. A central part of the design process involved developing interface metaphors. Again, when ideas for metaphors were

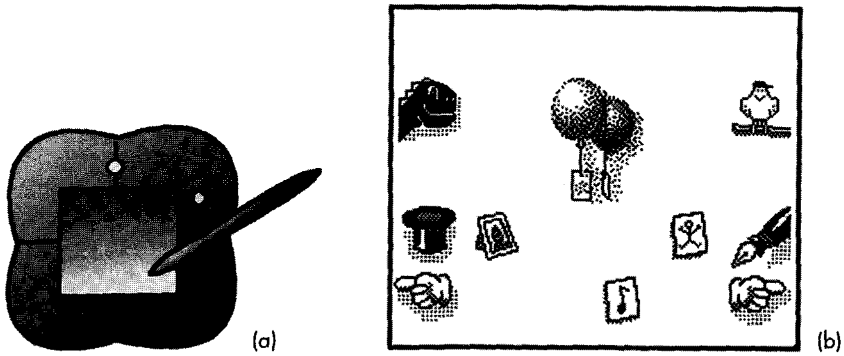


Figure 15.6 (a) The communicator with pen. (b) Product display showing 'the world'.

proposed, the designers turned to the girls in a spirit of participatory design in order to elicit their responses.

What usability and user experience goals were considered important? In the Nokia communicator example we saw the importance of usability goals focusing on effectiveness and efficiency, especially the need to move smoothly among critical tasks. In contrast, Philips focused more on the user experience goals of being enjoyable, entertaining, and fun. Other goals were that it should encourage creativity and provide personal and magical applications. The girls had expressed a specific desire for these.

What functionality did the communicator provide? The communicator was designed to have a touch-sensitive screen, pen input, infrared communications, and audio output (see Figure 15.6(a)). The interface was built on the metaphor of a world in which the users can move around freely, picking things up and starting applications (see Figure 15.6(b)). Available applications include a calendar, alarm clock, photo album, fortune teller, and communicator. The user can also perform tasks such as writing letters, composing tunes, drawing pictures, and sending them to other similar devices (see Figure 15.7).

What methods were used? Development of the product was divided into four phases: initiation, concept creation, specification, and finalization. Whereas Nokia adopted techniques from contextual design, Philips used mainly low-fidelity prototyping techniques for this particular project. Different prototypes were used throughout the development and for different purposes.

During the initiation phase, foam models were used to elicit feedback on the color, shape, size, styles, and robustness of the device, among other things. Using group discussions to encourage the youngsters to express their opinions a lot of feedback was gained from the foam models, even though the models contained no functionality. For example, children liked the idea of protecting the screen when carrying it, so they wanted different bags and cases to be provided for it; privacy was an important aspect, so they did not want it easily accessible by others; the pen should be stored safely within the device rather than underneath it for fear of it

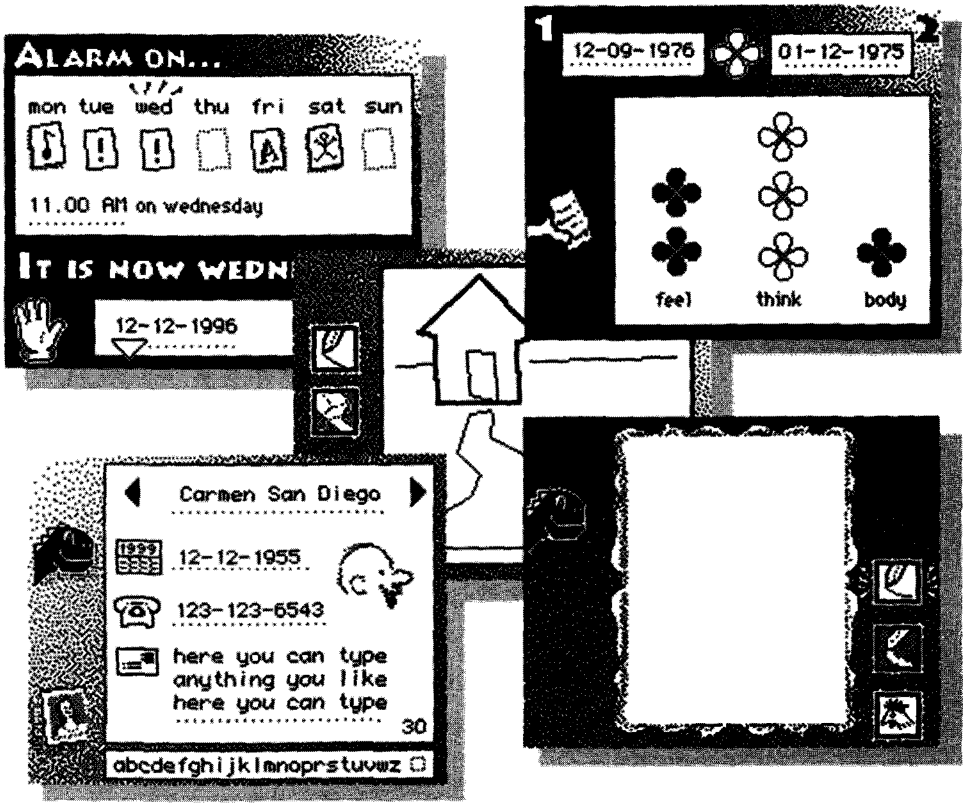


Figure 15.7 Some of the built-in applications.

being lost. One surprising result was that the children did not like the colors. The initial colors were bright (See Figure 15.8 on Color Plate 8), but they wanted dark colors more akin to their parents' hi-fi equipment at home.

The session with the models also provided input for the first user interface design, which was animated using a computer-based tool. This was used to explore navigation, pen-based dialog, types of application, and visual style.

During the concept creation phase, dynamic visualizations, which are like the storyboards described in Chapter 8 but are computer-based, were used to capture the initial ideas about interface and functionality (see Figure 15.9).

During the specification phase, foam models were again used to decide the size of the screen appropriate for writing on while standing up. As well as the size, different display formats were simulated (see Figure 15.10). These prototypes proved to be effective, again eliciting a lot of useful feedback. For example, left-handed users used the upper left part of the product to lean on while writing and the right-handed children used the lower right portion, yielding the design implication that the product should have hand resting places at these two points.

Also during specification, ideas for the interface design were evaluated by youngsters at a fair. There were two main contenders for the interface design.

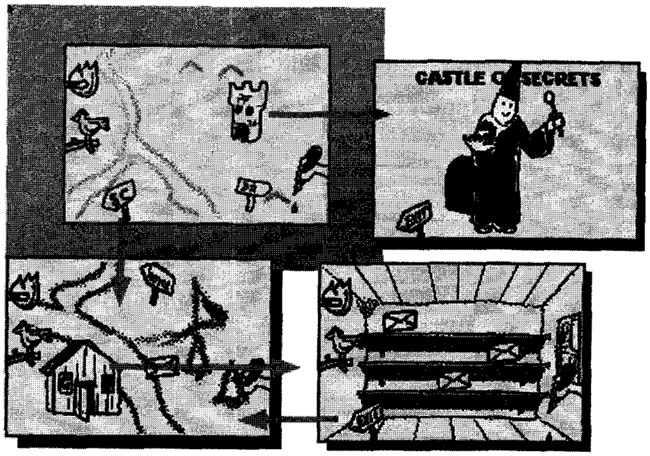


Figure 15.9 The first dynamic visualizations.

One provided direct access to each of the applications in the device, represented as a static matrix of options. This meant that the visual presentation and size of the applications was limited by the size of the screen. The other interface **worked** by indirect access, through a navigation model based on the idea of a window moving over a linked list of options.

Prototyping was also used in the finalization phase for market evaluations.

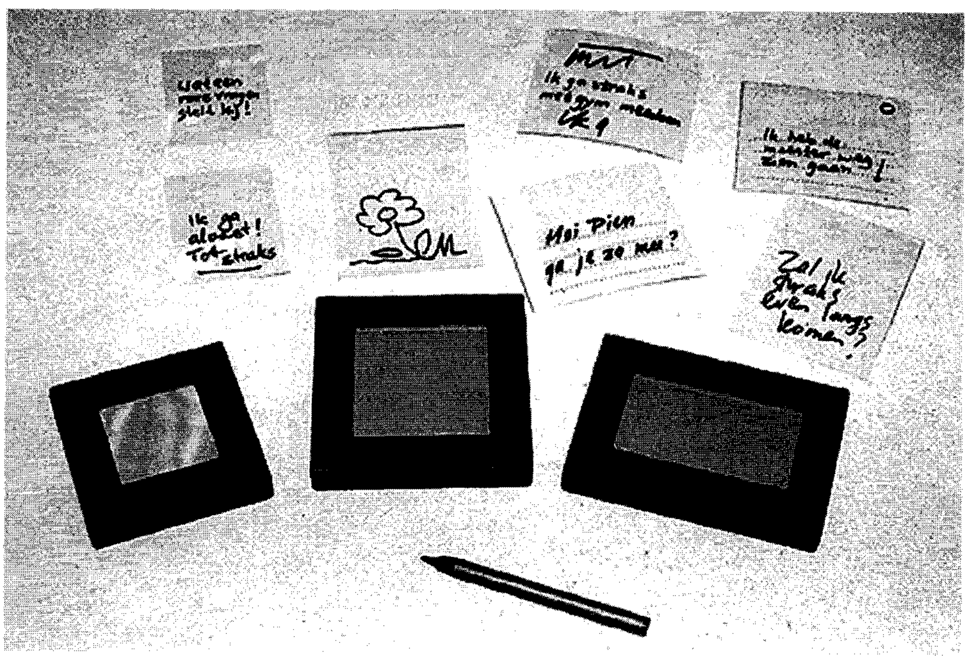


Figure 15.10 Foam models for investigating display size and screen format.

ACTIVITY 15.5 Prototypes are often used to answer specific questions. In this development, what questions were answered by producing and evaluating the foam models?

Comment Foam models were used at two specific points in the development to answer clear questions. The first set was used to consider the physical design such as size and color. They also elicited comments about storing the pen, covering the display, and having a carrying bag. The second set was used to design the display size and format. This also had the side effect of finding out useful information about where children would rest their hands on the device.

How much did the children participate in the design? One of the problems with participatory design is knowing how much to involve the users. Trying to involve children too much can be counterproductive, boring them and sometimes making them feel out of their depth. Asking children to participate too little can end up making them feel as if their views and ideas are not being sufficiently taken into account.

The Philips design team involved the children in design and evaluation from the very beginning. The first participatory design session was held during the initiation phase at a local international primary school. The session investigated the social and personal lives of 7 to 12 year-olds. Groups of 8 to 10 children were engaged in discussions and were asked to draw sketches of their ideal product. They were also asked to write stories about the use of the product, so that designers could get some contextual information about how it might be used. From this first session, it was clear that the concept was well received by the children. They particularly liked the communication, the pen-based interface, and its multifunctionality.

There were clear differences between boys, who wanted a broader range of functionality, and girls, who focused on communication. The ability to personalize was important to both groups. For example, one girl wanted the device to cough when a message arrived so that the teacher wouldn't know she was using it during class.

The whole design team was present at participatory design sessions. Spending time to get the children's opinions and to enter their world to understand how they perceive things was important for the success of the product.

One lesson that the designers drew from this exercise echoes a comment by Gillian Crampton Smith in the interview at the end of Chapter 6: users are not designers. In this instance, the children were limited in what they could design by what they knew and what they were used to. Another stakeholder group, parents, expected keyboard input, as they believed this to be more sophisticated than pen input, which was seen as old fashioned.

On the other hand, children are often more imaginative than adults, so involving the children was useful when discussing innovative ideas, or when only partial ideas were available. Working with children like this rather than adults requires a different approach, yet both adults and children need to appreciate each others' strengths and weaknesses. Box 15.3 describes the intergenerational design teams that Druin works with in projects at the University of Maryland.

BOX 15.3 Children and Adults Bring Participant Observation Close to Design

Allison Druin designs innovative technology with intergenerational design teams in which children and adults work together (Druin, 2000). In her teams children and adults observe children interacting with low-fidelity prototyping materials—crayons, pens, paper, glue, scissors, felt, furry cloth, Lego, animal parts, etc. (Figure 15.11), to explore ideas. By keeping more complex technology, such as computers, out of the picture during early observation and brainstorming sessions, adults do not dominate the scene.

Both adult and child members of Druin's teams observe and take notes while other children interact with the prototypes. This enables the team to capture impressions from both child and adult perspectives. Originally observations were recorded on a data-capture form like that in Figure 15.12 but many children prefer to draw and write simple notes like those in Figure 15.13. Adults, on the other hand, generally prefer to write, so now the team uses both techniques.

A typical observation session includes a pair of observers, an interactor and a child. The interactor's role is to ask questions that initiate discussion about the activities. Without this essential role, the children tend to feel that they are on stage being observed. But when engaged in discussion they are more likely to relax and reveal their real behavior and opinions. It is also important that the interactor does not take notes because this can make the children feel that they are being tested. Therefore, video is used to record observations.

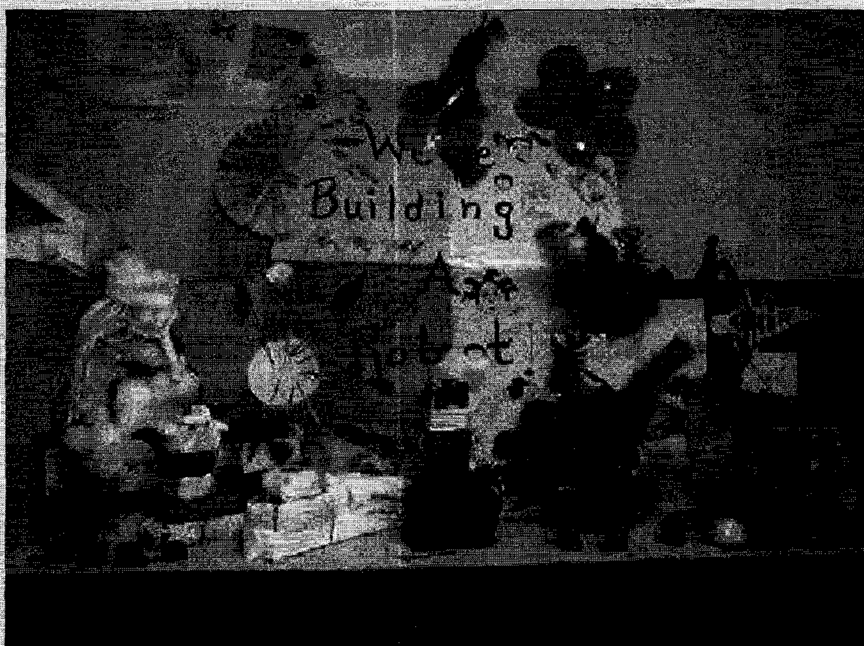


Figure 15.11 Early design ideas for "fluffy-fuzzy" robots using low-tech prototyping materials.

RAW DATA:			DATA ANALYSIS:		
Time	Quotes	Activities	Activity Patterns	Roles	Design Ideas
10.05	E: Can you draw whatever you want? [Gustav: Yes.] E: (To K) A Christmas Tree? K: Yes!	K. takes the mouse rapidly, draws a red tree, takes the yellow crayon	Drawing	Artist	
		draws something in the corner, rubs out, continues	Drawing Erasing	Artist	
		E. tries to take the mouse	Struggling for control of input device	Leader	Multiple input devices
	E: But I want the long one! E: Noo! [Difficult to erase.] E: There. E: But what's this? [Windows Start menu appears.]	E gets the mouse, tries to get the blue crayon, looks irritated when she cannot get the blue one, gets it	Difficulty selecting tools	Frustrated User	Easier way to select tools

Figure 15.12 Excerpt from a data capture form.

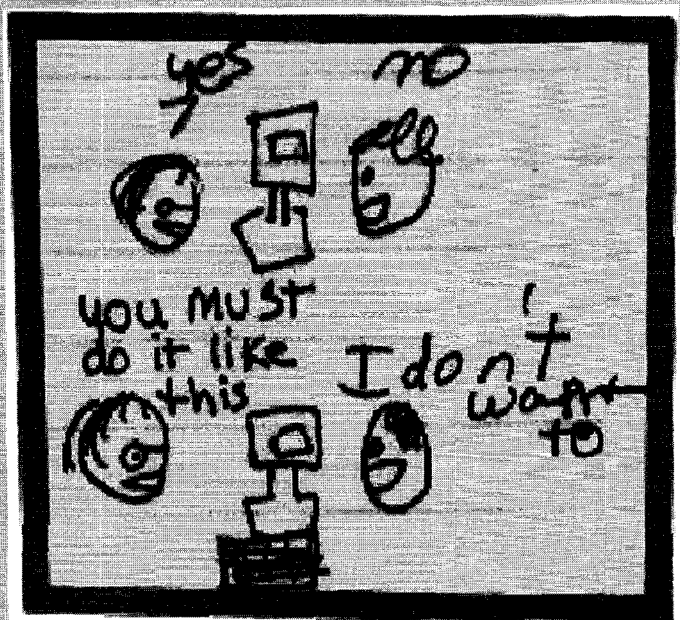


Figure 15.13 Sample notes illustrating a child's observation.

ACTIVITY 15.6 Suggest ways of helping adults and children feel comfortable together and gain mutual acceptance.

Comment Allison Druin asks everyone to dress casually in jeans, sneakers and T-shirts. The group works together at shared tables or on the floor. Snacks are important in creating a relaxed environment, and everyone uses first names. The goal is to create a group in which everyone respects each other's contributions and accepts and welcomes different contributions. Children are used to being controlled by adults and adults are used to being in control, and it takes time to break down these ingrained stereotypes.

What conceptual models did they design? By the concept creation phase, the importance of four goals for the product and its interface had emerged:

1. to support communication by stimulating social interaction among children
2. to evoke creativity and fantasy
3. to be "alive"—unexpected fun things should happen, surprising and pleasurable to the user, that give the product more character
4. to enhance intimacy—the product is a personal asset containing personal information

Five metaphors were developed by designers based on these values. Each metaphor was represented by a story. Figure 15.14 shows an illustration of one metaphor: the wizard. Specific metaphor workshops were conducted to find out how the girls reacted to the metaphors. They were asked to create a collage to visualize the metaphors, showing what they understood by them. The collages were a combination of drawings, essays, and existing pictures. The metaphor workshop showed that the girls were interested in being able to create, communicate, and organize personal things.

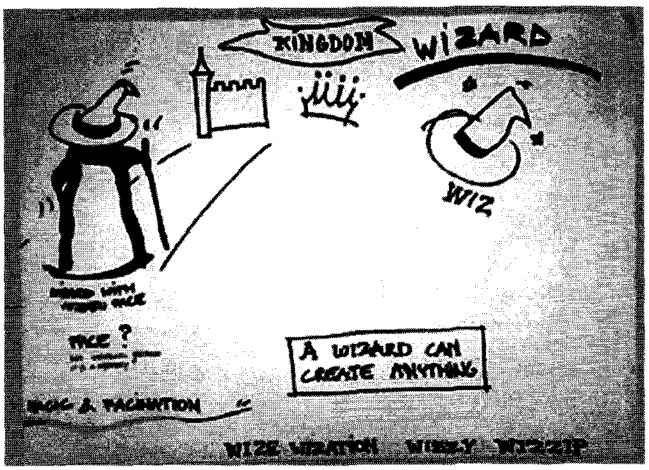


Figure 15.14 One of the metaphors: the wizard.

How did they evaluate the conceptual model? During the finalization stage, usability evaluations with children were performed to investigate the user interface itself and also to answer specific questions concerned with ideas for games, and writing performance. In most sessions, users were asked to play with the device for a certain period of time before giving feedback.

What lessons were learned from this case study? Many lessons were learned from developing an innovative product using a combination of participatory design and user testing. Some practical advice offered by Oosterholt and colleagues that can be generalized to the design of other interactive products is:

> ***Specify Your User Requirements And Define Milestones*** The rationale behind specifying user requirements is not just to develop them, but to make sure that the team agrees on the assumptions and realizes how and when they have been and can be changed.
>
> ***A Product Is Not Designed in a Vacuum*** Start thinking about additional and follow-up products at an early stage, so one does not have to change suddenly or add extra functionality in a later phase.
>
> ***Users Are Not Designers*** Not all answers can be generated by user or market tests. Users will generally relate any new product concept to existing products.
>
> ***Act Quick And Dirty If Necessary*** Often, the purpose of user testing is not to decide whether one interface concept is more usable than an alternative concept, but to discover issues that are important to the children. Small qualitative sessions of user involvement are therefore often appropriate. Furthermore, such sessions provide an opportunity for designers to "enter" the children's world.

15.4 Redesigning part of a large interactive phone-based response system

In this case study, we focus on quite a different kind of system, one being redesigned for a specific application intended to provide the general public with advice about filling out a tax return—and those of you who have to do this know only too well how complex it is. The original product was developed not as a commercial product but as an advisory system to be interacted with via the phone. We report here on the work carried out by usability consultant Bill Killam and his colleagues, who worked with the US Internal Revenue Services (IRS) to evaluate and redesigned the telephone response information system (TRIS).

Although this case study is situated in the US, such phone-based information systems are widespread across the world. Typically, they are very frustrating to use. Have you been annoyed by the long menus of options such systems provide when you are trying to buy a train ticket or when making an appointment for a technician to fix your phone line? What happens is that you work your way through several different menu systems, selecting an option from the first list of, say, seven choices, only to find that now you must choose from another list of five alternatives. Then, having spent several minutes doing this, you discover that you made the wrong choice back in the first menu, so you have to start again. Does this sound familiar? Other problems are that often there are too many options to remember, and that none of the options seems to be the right one for you. In such situations, most users long for human contact, for a real live operator, but of course there usually isn't one.

TRIS provided information via such a myriad of menus, so it was not surprising that users reported many of these problems. Consequently a thorough evaluation and redesign was planned. To do this, the usability specialists drew on many techniques to get different perspectives of the problems and to find potential solutions. Their choice of techniques was influenced by a combination of constraints: schedules, budgets, their level of expertise, and not least that they were working on redesigning part of an already existing system. Unlike new product development, the design space for making decisions was extremely limited by existing design decisions and the expectations of a large existing user population.

15.4.1 Background

Everyone over age 18 living in the US must submit a tax return each year either individually or included in a household. The age varies from country to country but the process is fairly similar in many countries. In the US this amounts to over 100 million tax returns each year. Completing the actual tax return is complex, so the IRS provides information in various forms to help people. One of the most used information services is TRIS, which provides voice-recorded information through an automated system. TRIS also allows simple automated transactions. Over 50 million calls are made to the IRS each year, but of these only 14% are handled by TRIS. This suggested to the designers that something was wrong.

15.4.2 The redesign

How do users interact with the current version of TRIS? The users of TRIS are the public, who get information by calling a toll-free telephone number. This takes them to the main IRS help desk, which is in fact the TRIS. The interface with TRIS is recorded voice information, so output is auditory. Users navigate through this system by selecting choices from the auditory menu that they enter by typing on the telephone keypad. First, the users have to interact with the Auto Attendant portion of the system—a sort of simulated operator that must figure out what the call is about and direct it to the proper part of the system. This sounds simple but there is a problem. Some paths have many **subpaths** and the way information is classified under the four main paths is often not intuitive to users. Furthermore, some of the functionality available through TRIS is provided by two other independent systems, so users can become confused about which system they are dealing with and may not even know they are dealing with a different system. Users get very few clues that these other systems exist or how they relate to each other, yet suddenly things may be quite different--even the voice they are listening to may change. Navigating through the system, with its lack of visual feedback and few auditory clues, is difficult. Imagine being in a maze with your eyes blindfolded and your hands tied so you can't feel anything, and where the only information you get is auditory. How can you possibly remember all the instructions and construct an accurate mental model in your head to help you?

Once in TRIS, users can take various paths that:

- Provide answers to questions about tax law (provided by one of the two other computer systems accessible through TRIS).
- Allow people to order all the forms and other materials they need to complete their tax return (provided by the two other systems accessible through TRIS).
- Perform simple transactions, such as changing a mailing address, ordering a copy of a tax return, or obtaining answers to specific questions about a person's taxation.
- Reach a live operator if none of the above options are applicable or the user cannot figure out how to use the system.

ACTIVITY 15.7 Why is developing an accurate mental model of TRIS difficult for users?

Comment Much of TRIS is hidden to the users. Their interaction with it is indirect, through listening to responses from the system and pressing various keys (whose meaning is always context dependent). There is no visual interface and users have only speech output to support their mental model development. Because speech is transient, unlike visual feedback, users must work out the conceptual model without visual cues. The user interface to this system is a series of menus in a tree structure and, since human short-term memory is limited, the structure of the system must also be limited to only a few branches at each point in the tree. Another problem is that TRIS accepts input only from the telephone number keypad, so it's not possible to associate unique or meaningful options with user choices.

What are the main problems identified with the existing version of TRIS? Because one of the main problems users have when using TRIS is developing a mental model of the system it is hard for users to find the information they need. In addition, TRIS was not designed to reveal the mapping of the underlying systems and often did things that made sense from a processing point of view but not from the user's. This is probably because the programmers took a data-oriented view of the system rather than a user-oriented one. For example, TRIS used the same software routine to gather both a social security number and an employee identification number for certain interactions. This may be efficient from a code-development standpoint, since only one code module needs to be designed and tested, but from the user's perspective it presented several problems. The system always had to ask the user which type of number was expected, even though only one of these numbers made sense for many questions being asked. Consequently, many users unfamiliar with employee identification numbers were not sure what to answer, those who knew the difference wondered why the system was even asking, and all users had yet another chance to make an entry error.

What methods did the usability experts use to identify the problems with the current version of TRIS? To begin with the usability specialists did a general review of the literature and industry standards and identified the latest design guidelines and current industry best practices for interactive voice response (IVR) systems. These guidelines formed the basis for a heuristic evaluation of the existing TRIS user interface and helped identify specific areas that needed improvement. They also used the GOMS keystroke-level modeling technique to predict how well the interface supported users' tasks. Menu selection from a hierarchy of options is quite well suited to a GOMS evaluation, although certain modifications were necessary to estimate values for average performance times.

What did they do with the findings of the evaluation? Once the analysis of the existing interface and user tasks was complete, the team then followed a set of design guidelines and standards, to develop three alternative interfaces for the Auto Attendant part of TRIS. An expert peer panel then reviewed the three alternatives and jointly selected the one that they considered to have the highest usability. The usability specialists also performed a further GOMS analysis for comparison with the existing system. The analysis predicted that it would only take 216.2 seconds to make a call with the new system, compared with 278.7 seconds with the original system. While this kind of prediction can highlight possible savings, it says little about which aspects of the redesign are more effective and why. The usability specialists, therefore, needed to carry out other kinds of user testing.

ACTIVITY 15.8 Why is it that the results from a GOMS analysis do not necessarily predict the best design?

Comment The keystroke-level analysis predicts performance time for experts doing a task from beginning to end. Not all of the users of TRIS will be experts, so performance time is not the only predictor of good usability.

The usability specialists did *three iterations* of user testing in which they simulated how the new system would work. When they were confident the new Auto Attendant interface had sufficient usability, they redesigned a subset of the underlying functionality. A new simulation of the entire Auto Attendant portion of TRIS was then developed. It was designed to support two typical tasks that had been identified earlier as problematic, to:

- find out the status of a tax refund
- order a transcript of a tax return for a particular year

These tasks also provide examples of nearly all of the user–system interactions with TRIS (e.g., caller identification, numeric data entry, database lookup, data playback, verbal instructions, etc.). A separate simulation of the existing system was also developed so that the new and existing designs could be compared. The user interaction was automatically logged to make data collection easier and unobtrusive.

What conflicts can arise when suggesting changes for improvement? When carrying out an evaluation of an existing product, often "jewels in the mud" stick out—glaring usability problems with a system that, if changed, could result in significant improvements. However, conflicts can arise when suggesting such changes, especially if they may decrease the efficient running of the system. The usability specialists quickly became aware that the TRIS system was making too many cognitive demands on users. In particular, the system expected users to select from too many menu choices too quickly. They also realized that immediate usability improvements could be gained by just a few minor changes: breaking menu choices into groups of 3–5 items; making the choices easier to understand; and separating general navigation commands (e.g., repeat the menu or return to the top menu) from other choices with pauses. However, to make these changes would require adding additional menus and building in pauses in the software. This conflicts with the way engineers write their code: they are extremely reluctant to purposely add additional levels to a menu structure and resist purposely slowing down a system with pauses.

ACTIVITY 15.9 The gap between programmers' goals and usability goals is often seen in large systems like TRIS that have existed for some time. How might such problems be avoided when designing new systems?

Comment It can be hard to get changes made when a system has been in operation for some time, but it is important for interaction designers to be persistent and convince the programmers of the benefits of doing so. Involving users early in design and frequent cycles of 'design-test-redesign' helps to avoid such problems in the design of new systems.

How were the usability tests devised and carried out? In order to do usability tests, the usability specialists had to identify goals for testing, plan tasks that would satisfy those goals, recruit participants, schedule the tests, collect and analyze data, and report their findings. Their main goals were to:

- evaluate the navigation system of the redesigned TRIS Auto Attendant
- compare the usability of the redesign with the original TRIS for sample tasks

Twenty-eight participants were recruited from a database of individuals who had expressed interest in participating in a usability test. There was an attempt to recruit an equal number of males and females and people from a mixture of education and income levels. The participants were screened by a telephone interview and were paid for their participation. The tests were conducted in a usability lab that provided access to the two simulated TRIS systems (the original design and the redesign). The lab had all the usual features (e.g., video cameras) and a telephone. Timestamps were included in the videotape and the participants' comments were recorded.

The order of the tasks and the order in which the systems were used was counter-balanced. This was done so that participants' experience on one system or task would not distort the results. So, half the participants first experienced the original TRIS design and the other half first experienced the redesigned TRIS system. That way, if a user learned something from one or other system the effects would be balanced. Similarly, the usability specialists wanted to avoid ordering effects from all the participants doing the same task first. Half the participants were therefore randomly allocated to do task A first and the other half to do task B. Taking both these ordering effects into account produced a **4** $\times$ **4** experimental design with eight participants for each condition.

ACTIVITY 15.10 Compare the description of this testing procedure with that for HutchWorld in Chapter 10. What differences do you notice and how can they be explained?

Comment The testing for HutchWorld is more typical. There were fewer participants and only one version of the system was tested at any time. In the TRIS test a larger number of participants were involved and the tests were more like an experiment. TRIS is complex, particularly the mapping between TRIS and the underlying functionality, although the system's purpose is clearly defined. By the time the usability specialists started the tests, they believed that they had fixed the major usability problems because they had responded first to the expert reviewers' feedback and then to the GOMS analysis. They were therefore confident that the new design would be better than the original one, but they had to demonstrate this to the IRS. This style of testing was also possible because there were thousands of potential users and the cost savings over 50 million calls justified the cost of this elaborate testing procedure.

How did they ensure that the participants tested were a representative set of users? In order to get demographic information to make sure the participants were representative, a questionnaire was given to all of them. It revealed a broad range of ethnicity, educational accomplishment, and income among the **18** women and **14** men who took part in the tests. Most had submitted tax returns during the last five years and most were experienced with interactive voice response systems. Eight participants indicated strong negative feelings about IVR systems, saying they were frustrating, time-consuming, and user-unfriendly.

What data was collected during the user testing? A total of 185 subnavigation steps made up the two tasks for the current TRIS. Participants successfully completed **91** steps on their first attempt **(49%** of the total). This was compared with a similar number of steps for the redesigned system: **187** subnavigation steps made up the same tasks for the redesigned TRIS. Participants were able to complete **117** of the steps on the first attempt (62% of the total), indicating an improvement of over 10%.

The average time to perform tasks was also analyzed. The summary data for the two tasks is shown in Table 15.1. As you can see, performance time on the redesigned system was much better for both tasks.

How was the user's satisfaction with the system assessed? At the end of each task, participants were asked to evaluate how well they thought the system enabled

Table 15.1 Average total task completion time by systems in seconds (s)

Task	Original system (s)	Redesigned system (s)
A	264.3	186.9
B	348.7	218.1

them to accomplish their tasks by completing a user satisfaction questionnaire. The responses again indicated that participants thought the redesign was easier to use and they preferred it. Regardless of the order in which participants used the two systems, the scores on the ***redesigned*** system were consistently much better than for the ***original*** system. The questionnaire provided statements that the participants had to rate on a 7-point scale. The difference between the two systems was highly significant, averaging over **3** rating-scale points higher on each statement.

ACTIVITY 15.10 User satisfaction questionnaires like the ones just described enable usability specialists to get answers to questions they regard as important. How can you make sure you collect opinions on all the topics that are most important to users?

Comment Asking users' opinions informally after pilot testing the questionnaire helps to make sure that you cover everything, but it is not foolproof. Furthermore, you may not want to increase the length of the questionnaire. Two other approaches that could be used separately are to ask users to think aloud and to use open-ended interviews. However, the think aloud method can distort the performance measures, so that is not such a good idea. Open-ended interviews are better, and this was done by the usability specialists in this case.

Participants were also invited to make any additional comments they wanted about the two systems. These were then categorized in terms of how easy the new system was considered to navigate, whether it was less confusing, faster, etc. Specific complaints included that some wording was still unclear and that not being able to return to previous menus easily was annoying. No matter how much usability testing and redesign you do, there is always room for improvement.

Would it have been better to redesign the entire system? It would have been far too expensive and time-consuming to redesign and test the whole system. A skill that usability specialists need when dealing with this much complexity is how to limit the scope of what they do and still produce useful results.

What other design features could be considered besides improving efficiency? Given that the system is aimed at a diverse set of users, many whose native language is not English, a system that uses different languages would be useful (the Olympic Messaging System used in the Los Angeles games did this very successfully). **A** range of voices could also be tested to compare the acceptability of different kinds of voices.

This case study has illustrated how to use different techniques in the evaluation and redesign of a system. Expert critiques and GOMS analyses are both useful tools for analyzing current systems and for predicting improvements with a proposed new design. But until the systems are actually tested with users, there is no way of knowing whether the predictions are accurate. What if users can theoretically carry out their tasks faster but in practice the interface is so poor that they cannot use it? In many cases, testing with real users is needed to ensure that the new design really does offer an improvement in usability. In this case study, results from usability testing were able to indicate that not only was the new design faster but users also liked it much better.

Summary

The three case studies illustrate how different combinations of design and evaluation techniques can be used effectively together to arrive at a design for a new product or redesign of an existing system. Quite different demands are placed on the design team when redesigning an existing product compared with designing a new product. Many practical problems and constraints will be encountered in both situations and experience of designing different systems will help you learn how to deal with them.

Key points

- Design involves trade-offs that can limit choices but can also result in exciting design challenges.
- Prototypes can be used for a variety of purposes throughout development, including for marketing presentations and evaluations.
- The design space for making changes when upgrading a product is limited by previous decisions.
- The design space is much greater when building new products.
- Rapid prototyping and evaluation cycles help designers to choose among alternatives in a very short time.
- Simulations are useful for evaluating large systems intended for millions of users when it is not feasible to work on the system directly.
- Piecing together evidence from data from different sources can provide a rich picture of usability problems, why they occur, and possible ways of fixing them.

Further Reading

BREWSTER, S., AND DUNLOP, M. (2000) (eds.) *Personal Technologies.* **Special issue on Human Computer Interaction and Mobile Devices, 4, 2&3. This collection of articles discusses many issues in the design of mobile devices and would be a good starting point for anyone interested in pursuing this area.**

BERGMAN, ERIC. (2000) (ed.) *Information Appliances and Beyond.* **San Francisco, CA: Morgan Kaufinann. This book contains an excellent collection of practical articles describing how different information appliances have been developed, from interactive toys and games to a vehicle navigation system.**

KILLAM, H. W. AND AUTRY, M. (2000) IVR interface design standards: A practical analysis. In Proceedings of HFES/IEA **44th Annual Meeting. This paper describes aspects of the TRIS study in more detail.**

Reflections from the Authors

To end the book, we each present some of our views about interaction design.

Helen: When I worked as a programmer/analyst in the City of London, during the early 1980s, I was always surprised and impressed by the workarounds that my company's clients devised in order to make the software they used work for them. At the same time, of course, I was also disappointed that the software didn't support them better. The real end users were often not consulted during the development, and had the systems thrust upon them. The situation nowadays is so much better, and I think it's great that the importance of involving users is now so widely recognized.

There have been great technological advances, creating some quite incredible devices, but we also shouldn't forget the more mundane applications of technology, which at times I think we tend to ignore. As Gillian Crampton Smith said in her interview, the software we use has become an environment in which we spend a lot of our time, either at work or in our leisure. These are interactive systems too and deserve our attention to make them more usable.

But for me, one of the most exciting implications of the kinds of advances we are seeing in interaction design is not technological, nor because of the focus on users, but because of the increased need for multidisciplinary teams. Having to work in a multidisciplinary team creates challenges but also great opportunities to learn from other disciplines and to create a much better product. In my research, I have been involved with a variety of different designers, for example software, architectural, knitwear, and electronic. There is so much to learn from each other. I look forward to it!

Jenny: Since the three of us started working together in the early 1990s, the changes in technology have been phenomenal. The web, the Internet, and cell phones have transformed the way we live. Although the usability of these systems has improved, we need to strive to make them even more compact, computationally powerful, universally usable, and attractive.

I'm aware of my good fortune in having access to state-of-the-art technology, but what about people who aren't so privileged? We need low cost products that are faster, do more, and can be used by people of different cultures, ages, abilities, and experiences. Designing fancy web graphics may be fun but if users cannot access them because of slow Internet connections and old machines, what use are they? Designing for universal usability is a challenge and I hope this book will help you to create systems that are more usable by more people, more of the time.

My research is concerned with developing online communities that combine appropriate support for social interaction (i.e., sociability) with well designed software (i.e., usability). These virtual communities enable people to reach out to each other in new ways, but we need a deeper understanding of why some communities fail while others thrive. I hope that more multidisciplinary teams will be inspired to meet this exciting challenge.

Yvonne: Writing this book has made me realize how much and how rapidly the field of interaction design has expanded in the last ten years. When we wrote our first textbook on human-computer interaction in the early '90s, the web hadn't even arrived and mobile and wireless devices were still very much a dream. "WIMP" was very much ***the*** paradigm which interface designers (sic) developed applications for. Now everything has changed. Technology has advanced so rapidly that interaction designers (sic) now need to think about a whole host of different issues, besides the way an interface should look and behave. Moreover, there is greater eclecticism, in terms of users, settings, activities, and spaces to design for. For example, interaction designers are now involved in designing interactive products for use both indoors and outdoors (e.g., handheld devices, wearables), for work, home, school, and leisure, for both very large surfaces (e.g., interactive whiteboards) and very small screens (e.g., mobile phone displays)—to name but a few.

What this amounts to is a growing need for new methods and techniques to help in the design and evaluation of this new range of user experiences. As we point out in the book, techniques developed for screen-based systems often do not scale up very well and are inappropriate for other kinds of systems (e.g., very large collaborative virtual environments or "inhabited TV" where there may be thousands of users interacting at the same time). In addition, new theories will also need to be developed to inform the design of user experiences that are enjoyable and meaningful and expand our cognitive and social capabilities. I believe it is a very challenging time for both academic researchers and designers working in the commercial world.

References

ANNETT, J. AND DUNCAN, K. D. (1967) Task analysis and training design, *Occupational Psychology,* 41, 211–21.

APPLE COMPUTER INC. (1993) *Making IT Macintosh: The Macintosh Human Interface Guidelines Companion* (CD-ROM).

APPLE COMPUTER INC., (1987) *Human Interface Guidelines.* Harlow, UK: Addison-Wesley.

ANDREWS, D., PREECE, J., AND TUROFF, M. (2001) A conceptual framework for demographic groups resistant to online community interaction. In *Proceedings of IEEE Hawaiian International Conference on System Science (HZCSS).*

ATKINSON, P., AND HAMMERSLEY, M. (1994) Ethnography and participant observation. In N. K. Denzin and Y. S. Lincoln (eds.) *Handbook of Qualitative Research.* London: Sage.

AUSTIN, J. L. (1962) *How to Do Things with Words.* Cambridge, MA: Harvard University Press.

BAILEY, B. (2000) How to improve design decisions by reducing reliance on superstition. Let's start with Miller's Magic 7±2. *Human Factors International, Znc.* www.humanfactors.com

BAILEY, R. W. (2001) Insights from Human Factors International, Inc. (HFI) Providing consulting and training in software ergonomics. January. www.humanfactors.com/home/

BAINBRIDGE, D. (1999) *Software Copyright Law* (4th ed.). London: Butterworths.

BASILI, V., CALDIERA, G., AND ROMBACH, D. H. (1994) *The Goal Question Metric Paradigm: Encyclopedia of Software Engineering.* New York: John Wiley & Sons.

BATES, J. (1994) The role of emotion in believable characters. *Communications of the ACM,* 37(7), 122–125.

BAUM, F. L., AND DENSLOW, W. (1900) *The Wizard of Oz.* New York: Random House, Inc.

BAYM, N. (1997) lnterpreting soap operas and creating community: inside an electronic fan culture. In S. Kiesler (ed.) *Culture of the Internet.* Hillsdale, NJ: Lawrence Erlbaum Associates, 103–119.

BELLOTTI, V AND ROGERS, Y. (1997) From web press to web pressure: multimedia representations and multimedia publishing. In *Proceedings of CSCW'97,* 279–286.

BEN ACHOUR, C. (1999) *Extracting Requirements by Analyzing Text Scenarios,* Thèse de Doctorat de Université Paris-6.

BENFORD, S., BEDERSON, B. B., AKESSON, K. P., BAYON, V., DRUIN, A., HANSSON, P., HOURCADE, J. P., INGRAM, R., NEALE, H., O'MALLEY, C., SIMSARIAN, K. T., STANTON, D., SUNBLAND, Y., AND TAXEN, G. (2000) Designing storytelling technologies to encourage collaboration between young children. In *Proceedings of CHI'2000,* 556–563.

BENNETT, J. (1984) Managing to meet usability requirements. In J. Bennett, D. Case, J. Sandelin, and M. Smith (eds.) *Visual Display Terminals: Usability Issues and Health Concern.* Englewood Cliffs, NJ: Prentice-Hall.

BEWLEY, W. L., ROBERTS, T. L., SCHROIT, D., AND VERPLANK, W. (1990) Human factors testing in the design of Xerox's 8010 'Star' office workstation. In J. Preece and L. Keller (eds.). *Human-Computer Interaction: A Reader.* Hemel Hempstead, *UK:* Prentice Hall, 368–382.

BERGMAN, E. AND HAITANI, R. (2000) Designing the PalmPilot: a conversation with Rob Haitani. In *Information Appliances.* San Francisco: Morgan Kaufmann.

BEYER, H. AND HOLTZBLATT, K. (1998) *Contextual Design: Defining Customer-Centered Systems.* San Francisco: Morgan Kauffman.

BEYNON-DAVIES, P. (1997) Ethnography and information systems development: ethnography of, for and within IS development. *Information and Software Technology,* 39,531–540.

BIAS, R. G. (1994) The pluralistic usability walkthrough—coordinated empathies. In J. Nielsen and R. L. Mack (eds.) *Usability Inspection Methods.* New York: John Wiley & Sons.

BLUMBERG, B. (1996) *Old Tricks, New Dogs: Ethology and Interactive Creatures.* PhD Dissertation. MIT Media Lab.

BLY, S. (1997) Field work: is it product work? *ACM Interactions Magazine,* January and February, 25–30.

BBDKER, S. (2000) Scenarios in user-centered design—setting the stage for reflection and action. ***Interacting with Computers,*** 13(1), 61–76.

BBDKER, S., GREENBAUM, J. AND KYNG, M. (1991) Setting the stage for design as action. In J. Greenbaum and M. Kyng (eds.) ***Design at Work: Cooperative Design of Computer Systems.*** Hillsdale, NJ: Lawrence Erlbaum Associates, 139–154.

BOEHM B., EGYED A., KWAN, J., PORT, D. SHAH A., AND MADACHY, R. (1998) Using the WinWin spiral model: a case study. ***IEEE Computer,*** 31(7), 33–44.

BOEHM, B. W. (1988) A spiral model of software development and enhancement, ***IEEE Computer,*** 21(5), 61–72.

BOGDEWIC, S. P. (1992) Participant observation. In B. F. Crabtree and W. L. Miller (eds.) ***Doing Qualitative Research.*** Newbury Park, CA: Sage, 45–69.

BORCHERS, J. (2001) ***A Pattern Approach to Interaction Design.*** **Chichester, *UK:* John Wiley & Sons.**

BRAITERMAN, J., VERHAGE, S. AND CHOO, R. (2000) Designing with Users in Internet Time. ***ACM Interactions Magazine,*** VII.5, 23–27.

BREAZEAL, C. (1999) Kismet: A robot for social interactions with humans. www.ai.mit.edu/projects/kismet/

BRINKLIN, D. (2001) ***VisiCalc: Information from its creators.*** www.bricklin.com/visicalc.htm

BROWN, B. A., SELLEN, A. J., AND O'HARA, K. P. (2000) ***A diary study of information capture in working life.*** In ***Proceedings of CHI 2000,*** The Hague, Holland, 438–445.

BUCHENAU, M. AND SURI, J. F. (2000) Experience prototyping. In ***Proceedings of DIS 2000 Design Interactive Systems: Processes, Practices, Methods, Techniques,*** 17–19.

BUTTON, G. AND SHARROCK, W. (1994) Occasioned practices in the work of software engineers. In Jirotka, M. and Goguen, J. A. (eds.) ***Requirements Engineering: Social and Technical Issues.*** San Diego: Academic Press, 217–240.

CARD, S. K., MACKINLEY, J. D., AND SHNEIDERMAN, B. (1999) (eds.) ***Readings in Information Visualization: Using Vision to Think.*** San Francisco: Morgan Kaufmann.

CARD, S. K., MORAN, T. P. AND NEWELL, A. (1983) ***The Psychology of Human-Computer Interaction.*** Hillsdale, NJ: Lawrence Erlbaum Associates.

CARROLL, J. M. (2000) Introduction to the special issue on "Scenario-Based Systems Development," ***Interacting with Computers,*** 13(1), 41–42.

CARROLL, J. M. (1990) ***The Nurnberg Funnel.*** Cambridge, MA: MIT Press.

CASSELL, J. (2000) Embodied conversational interface agents. ***communications of the ACM,*** 43(3), 70–79.

CHENG, L., STONE, L., FARNHAM, S., CLARK, A. M., AND ZANER-GODSEY, M. (2000) HutchWorld: Lessons Learned. A Collaborative Project: Fred Hutchinson Cancer Research Center & Microsoft Research. In ***Proceedings of the Virtual Worlds Conference 2000,*** Paris, France.

CHI Panel (2000) Scaling for the Masses: Usability Practices for the Web's Most Popular Sites.

CHIN, J. P., DIEHL, V. A., AND NORMAN, K. L. (1988) Development of an instrument measuring user satisfaction of the human-computer interface. ***In Proceedings of CHI'88.***

COCKBURN, A. (1995) Structuring use cases with goals. members.aol.com/acockburn/papers/usecases.htm.

COGDILL, K. (1999) ***MEDLINEplus Interface Evaluation: Final Report.*** College Park, MD: College of Information Studies, University of Maryland.

COMER, E. R. (1997) Alternative lifecycle models. In Merlin Dorfman and Richard H. Thayer (eds.) ***Software Engineering.*** Piscataway, NJ: IEEE Computer Society Press.

CONKLIN, J. AND BEGEMAN, M. L. (1989) gIBIS: A tool for all reasons. ***Journal of the American Society for Information Science,*** 40(3), 200-213.

CONSTANTINE, L. L. AND LOCKWOOD, L. A. D. (1999) ***Software for use.*** Harlow, UK: Addison-Wesley.

COYLE, A. (1995) Discourse analysis. In G. M. Breakwell, S. Hammond, and C. Fife-Schaw (eds.) ***Research Methods in Psychology.*** London: Sage.

CRAIK, K. J. W. (1943) ***The Nature of Explanation.*** Cambridge University Press.

CRAMPTON SMITH, G. (1995) The hand that rocks the cradle. ***ID Magazine,*** May/June, 60–65.

CUSUMANO, M. A. AND SELBY, R. W. (1995) ***Microsoft Secrets.*** London: Harper-Collins Business.

CUSUMANO, M. A. AND SELBY, R. W. (1997) How Microsoft builds software. ***Communications of the ACM,*** 40(6), 53-61.

DANIS, C. AND BOIES, S. (2000) Using a technique from graphic designers to develop innovative systems design. In ***Proceedings of DIS 2000,*** 20–26.

DENZIN, N. K., AND LINCOLN, Y. S. (1994) ***Handbook of Qualitative Research.*** London: Sage.

DIX, A., FINLAY, J., ABOWD, G., AND BEALE, R. (1993) ***Human-Computer Interaction*** (2nd ed.). London: Prentice-Hall Europe.

DOURISH, P. AND BELLOTTI, V. (1992) Awareness and coordination in shared workspaces. In ***Proceedings of CSCW'92***, 107–114.

DOURISH, P. AND BLY, S. (1992) Portholes: supporting awareness in a distributed work group. In ***Proceedings of CHI'92***, 541–547.

DRAY, S. M. AND MRAZEK, D. (1996) A day in the life of a family: an international ethnographic study. In D. Wixon and J. Ramey (eds.) ***Field Methods Casebook for Software Design.*** New York: John Wiley and Sons, 145–156.

DRUIN, A. (2000) The role of children in the design of new technology. ***University of Maryland, Human-Computer Interaction Laboratory Technical Report*** 99–23. www.cs.umd.edu/hcil.

DRUIN, A. (1999) ***The Design of Children's Software.*** San Francisco, CA: Morgan-Kaufmann.

DUMAS, J. S., AND REDISH, J. C. (1999) ***A Practical Guide to Usability Testing*** (Revised Edition). Exeter, UK: Intellect.

EASON, K. (1987) ***Information Technology and Organizational Change.*** London: Taylor and Francis.

EBLING, M. R., AND JOHN, B. E. (2000) On the Contributions of different empirical data in usability testing. In ***Proceedings of ACM DIS 2001,*** 289–296.

EDWARDS, A. D. N. (1992) Graphical user interfaces and blind people. In ***Proceedings of ICCHP '92***, Vienna: Austrian Computer Society, 114–119.

EHN, P. (1989) ***Word-oriented Design of Computer Artifacts*** (2nd edn.) Hillsdale, NJ: Lawrence Erlbaum Associates.

EHN, P. AND KYNG, M. (1991) Cardboard computers: mocking-it-up or hands-on the future. In J. Greenbaum and M. Kyng (eds.). ***Design at Work.*** Hillsdale, NJ: Lawrence Erlbaum Associates.

EICK, S. G. (2001) Visualizing online activity. ***Communications of the ACM,*** 44(8), 45–50.

ERICKSON, T. D. (1990) Working with interface metaphors. In B. Laurel (ed.). ***The Art of Human-Computer Interface Design.*** Boston: Addison-Wesley 65–73.

ERICKSON, T., SMITH, D. N., KELLOGG, W. A., LAFF, M., RICHARDS, J. T., AND BRADNER, E. (1999) Socially translucent systems: social proxies, persistent conversation and the design of "Babble". In ***Proceedings of CHI'99,*** 72–79.

ERIKSON, T. D., AND SIMON, H. A. (1985) ***Protocol Analysis: Verbal Reports as Data.*** Cambridge, MA: The MIT Press.

FETTERMAN, D. M. (1998) ***Ethnography: Step by Step*** (2nd ed.) Thousand Oaks, CA: Sage.

FISH, R.S. (1989) Cruiser: a multimedia system for social browsing. ***SIGGRAPH Video Review*** (video cassette) Issue 45, Item 6.

FISKE, J. (1994) Audiencing: cultural practice and cultural studies. In N. K. Denzin and Y. S. Lincoln (eds.) ***Handbook of Qualitative Research.*** Thousand Oaks, CA: Sage, 189–198.

FITTS, P. M. (1954) The information capacity of the human motor system in controlling amplitude of movement. ***Journal of Experimental Psychology,*** 47,381–391.

FITZPATRICK, G., MANSFIELD, T., KAPLAN, S., ARNOLD., D., PHELPS, T., AND SEGALL, B. (1999) Augmenting the workaday world with Elvin. In ***Proceedings of the Sixth European Conference on Computer-Supported Cooperative Work.*** Dordrecht, The Netherlands: Kluwer, 431-450.

FONTANA, A., AND FREY, J. H. (1994) ***Interviewing: The art of science.*** In N. Denzin and Y. Lincoln (eds.) ***Handbook of Qualitative Research.*** London: Sage, 361–376.

FROHLICH, D. AND MURPHY, R. (1999) Getting physical: what is fun computing in tangible form? In ***Computers and Fun*** 2, ***Workshop,*** 20 Dec. York, UK.

GAVER, B., DUNNE, T., AND PACENTI, E. (1999) Cultural probes. ***ACM Interactions Magazine,*** January and February, 21–29.

GENTNER, D. AND NIELSEN, J. (1996) The anti-Mac interface. *Communictions of the ACM,* 39 (8) 70–82.

GILL, J. AND SHIPLEY, T. (1999) ***Telephones–What Features do Disabled People Need?*** RNIB.

GOETZ, J. P., AND LECOMPTE, M. D. (1984) ***Ethnography and Qualitative Design in Educational Research.*** Orlando, FL: Academic Press.

GOUGH, P. A., FODEMSKI, F. T., HIGGINS, S. A., AND RAY, S. J. (1995) Scenarios — an industrial case study and hypermedia enhancements. In ***Proceedings of 2nd IEEE Symposium on Requirements Engineering,*** IEEE Computer Society, 10–17.

GOULD, J. D., AND LEWIS, C. H. (1985) Designing for usability: key principles and what designers think. ***Communications of the ACM,*** 28(3), 300–311.

GOULD, J. D., BOIES, S. J., LEVY, S., RICHARDS, J. T., AND SCHOONARD, J. (1987) The 1984 Olympic Message System: a test of behavioral principles of system design. ***Communications of the ACM,*** 30(9), 758–769.

GOULD, J. D., BOIES, S. J., LEVY, S., RICHARDS, J. T., AND SCHOONARD, J. (1990) The 1984 Olympic Message System: a test of behavioral principles of system design. In J. Preece and L. Keller (eds.) ***Human-Computer Interaction (Readings).*** Hemel Hempstead, UK: Prentice Hall International Ltd., 260–283.

GRAY, W. D., JOHN, B. E., AND ATWOOD, M. E. (1993) Project Ernestine: validating a GOMS analysis for predicting and explaining real-world performance. ***Human-Computer Interaction,*** 8(3), 237–309.

GREEN, T. R. G. (1990) The cognitive dimension of viscosity: A sticky problem for HCI. In D. DIAPER, D. GILMORE, G. COCKTON AND B. SHAKEL (eds.) ***Human-Computer Interaction—INTERACT'90.*** Elsevier Publishers, 79–86.

GREIF, I. (1988) ***Computer Supported Cooperative Work: a book of readings.*** San Francisco: Morgan Kaufmann.

GRUDIN, J. (1989) The case against user interface consistency. ***Communications of the ACM,*** 32(10), 1164–1173.

GRUDIN, J. (1990) The computer reaches out: the historical continuity of interface design. In ***Proceedings of CHI'90***, 261–268.

GUINDON, R. (1990) Designing the design process: exploiting opportunistic thoughts. ***Human-Computer Interaction,*** 5(2&3), 305–344.

HALVERSON, C. (1995) Inside the cognitive workplace: new technology and air traffic control. PhD Thesis, Dept. of Cognitive Science, University of California, San Diego.

HAMMERSLEY, M. AND ATKINSON, P. (1983) ***Ethnography: principles in practice.*** London: Tavistock.

HARPER, R. (2000) The organization of ethnography, In ***Proceedings of CSCW*** 2000, 239–264.

HARRISON, S., BLY, S. ANDERSON, S. AND MINNEMAN (1997) The media space. In Finn, K. E. Sellen, A. and Wilbur, S. B. (eds.) ***Video-Mediated Communication.*** Mahwah, NJ: Lawrence Earlbaum Associates, 273–300

HARTFIELD, B. AND WINOGRAD, T. (1996) Profile: IDEO. In T. Winograd (ed.) ***Bringing Design to Software.*** ACM Press.

HARTSON, H. R. AND HIX, D. (1989) Toward empirically derived methodologies and tools for human-computer interface development. ***International Journal of Man-Machine Studies,*** 31, 477–494.

HAUMER, P., JARKE, M., POHL, K., AND WEIDENHAUPT, K. (2000) Improving reviews of conceptual models by extended traceability to captured system usage. ***Interacting with Computers,*** 13(1), 77–95.

HEATH, C. AND LUFF, P. (1992) Collaboration and control: crisis management and multimedia technology in London Underground line control rooms. In ***Proceedings of CSCW'92***, 1, 1–2, 69–94.

HEATH, C., JIROTKA, M., LUFF, P., AND HINDMARSH, J. (1993) Unpacking collaboration: the interactional organization of trading in a city dealing room. In ***Proceedings of the Third European Conference on Computer-Supported Cooperative Work.*** Dordrecht: Kluwer.

HEINBOKEL, T., SONNENTAG, S., FRESE, M., STOLTE, W., and BRODBECK, F. C. (1996) Don't underestimate the problems of user centredness in software development projects—there are many! ***Behaviour & Information Technology,*** 15(4), 226–236.

HOCHHEISER, H., AND SHNEIDERMAN, B. (2001) Using interactive visualization of WWW log data to characterize access patterns and inform site design. ***Journal of the American Society for Information Science,*** 52, 4, 331–343.

HOLTZBLATT, K. AND JONES, S. (1993) Contextual Inquiry: a participatory technique for systems design. In D. Schuler, and A. Namioka, (eds.) ***Participatory Design: Principles and Practice,*** Hillsdale, NJ: Lawrence Erlbaum Associates, 177–210.

HOLTZBLATT, K., AND BEYER, H. (1996) Contextual Design: principles and practice. In D. Wixon and J. Ramey, (eds.) ***Field Methods Casebook for Software Design.*** New York: John Wiley and Sons, 301–333.

HUGHES, J. A., KING, RANDALL, D. AND SHARROCK (1993) Ethnography for system design: a guide, COMIC working paper, COMIC-LANCS-2-N. More information about COMIC is available from Cooperative Systems Engineering Group, Computing Department, Lancaster University, UK.

HUGHES, J. A., KING, V., RODDEN, T., AND ANDERSEN, H. (1994) Moving out of the control room: ethnography in system design. In ***Proceedings of CSCW'94***, Chapel Hill, NC.

HUGHES, J. A., O'BRIEN, J., RODDEN, T. AND ROUNCEFIELD, M. (1997) Designing with Ethnography: a Presentation Framework for Design. In *Proceedings of DIS '97*, 147–159.

HUGHES, J. A., SOMMERVILLE, I., BENTLEY, R. AND RANDALL, D. (1993a) Designing with ethnography: making work visible. *Interacting with Computers*, 5(2), 239–253.

HUTCHINS, E. (1995) *Cognition in the Wild.* Cambridge, MA: MIT Press.

ISENSEE, S., KALINOSKI, K. AND VOCHATZER, K. (2000) Designing Internet appliances at Netpliance. In E. Bergman (ed.) *Information Appliances and Beyond.* San Francisco: Morgan Kaufmann.

ISHII, H. AND ULLMER, B. (1997) Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of CHI'97*, 234–241.

ISHII, H., KOBAYASHI, M., AND Grudin, J. (1993) Integration of interpersonal space and shared workspace: Clearboard design and experiments. *ACM Transactions on Information Systems,* 11 (4), 349–375.

JACOBSON, I., CHRISTERSON, M., JONSSON, P. AND OVERGAARD, G. (1992) *Object-Oriented Software Engineering—A Use Case Driven Approach.* Harlow, UK: Addison-Wesley.

JOHNSON, M. and LAKOFF, G. (1980) *Metaphors We Live By.* Chicago: The University of Chicago Press.

JOHNSON-LAIRD, P. N. (1983) *Mental Models.* Cambridge: Cambridge University Press.

KAHN, R., AND CANNELL, C. (1957) *The Dynamics of Interviewing.* New York: John Wiley & Sons.

KARAT, C. M. (1993) The cost-benefit and business case analysis of usability engineering. InterChi '93, Amsterdam, Tutorial Notes 23.

KARAT, C.-M. (1994) A comparison of user interface evaluation methods. In J. Nielsen and R. L. Mack (eds.) *Usability Inspection Methods.* New York: John Wiley & Sons.

KARAT, J. (1995) Scenario Use in the Design of a Speech Recognition System. In J. M. Carroll (ed.) *Scenario-based Design,* 109–134. New York: John Wiley & Sons.

KARAT, J. AND BENNET, J. L. (1991) Using scenarios in design meetings. In Karat, J. (ed.) *Taking Design Seriously.* London: Academic Press.

KAY, A. (1969) *The Reactive Engine.* PhD Dissertation, Electrical Engineering and Computer Science, University of Utah.

KEIL, M. AND CARMEL, E. (1995) Customer-developer links in software development. *Communications of the ACM,* 38(5), *33-44.*

KEMPTON, W. (1986) Two theories of home heat control. *Cognitive Science,* 10, 75–90.

KETOLA, P., HJELMEROOS, H., AND RAIHA, K.-J. (2000) Coping with consistency under multiple design constraints: The case of the Nokia 9000 WWW browser. *Personal Technologies* 4(2&3), 86–95.

KIM, S. (1990) Interdisciplinary cooperation. In *The Art of Human-Computer Interface Design. B.* Laurel (ed.) Reading, MA: Addison-Wesley.

KOENEMANN-BELLIVEAU, J., CARROLL, J. M., ROSSON, M. B., AND SINGLEY, M. K. (1994) Comparative usability evaluation: critical incidents and critical threads. In *Proceedings of CHI'94.*

KOTONYA, G. AND SOMMERVILLE, I. (1998) *Requirements engineering: processes and techniques.* Chichester, UK: John Wiley & Sons.

KRAUT, R., FISH, R., ROOT, R. AND CHALFONTE, B. (1990) Informal communications in organizations: form, function and technology. In S. Oskamp and S. Spacapan (eds.) *People's Reactions to Technology in Factories, Offices and Aerospace.* The Claremont Symposium on Applied Social Psychology. Thousand Oaks, CA.: Sage Publications, 145–199.

KUHN, S. (1996) Design for people at work. In T. Winograd, (ed.) *Bringing Design to Software.* Boston: Addison-Wesley.

KUJALA, S. AND MÄNTYLÄ, M. (2000) Is user involvement harmful or useful in the early stages of product development? In *CHI* 2000 *Extended Abstracts,* ACM Press, 285–286.

LAKOFF, G. AND JOHNSON, M. (1980) *Metaphors we Live By.* Chicago: The University of C cago Press.

LAMBOURNE, R., FEIZ, K., AND RIGOT, B. (1997) Social trends and product opportunities: Philips' Vision of the Future Project. In *Proceedings of CHI'97*, 494–501.

LANSDALE, M. (1988) The psychology of personal information management. *Applied Ergonomics,* 55, 55-66.

LANSDALE, M. AND EDMONDS, E. (1992) Using memory for events in the design of personal filing systems. *International Journal of Human-Computer Studies,* 26, 97–126.

LARSON, K., AND CZERWINSKI, M. (1998) Web page design: implications of memory, structure and scent for information retrieval. In *Proceedings of CHI '98*, 25–32.

LAUREL, B. (1993) *Computers as Theatre.* New York: Addison-Wesley.

LAZAR, J., AND PREECE, J. (1999) Designing and implementing web-based surveys. ***Journal of Computer Information Systems, xxxix*** (4), 63-67.

LEE, J., KIM, J., AND MOON, JAE YUN (2000) What makes Internet users visit cyber stores again? Key design factors for customer loyalty. In ***Proceedings of CHI*** 2000, 305–312.

LESTER, J. C., AND STONE, B. A. (1997) Increasing believability in animated pedagogical agent. In ***Proceedings of Autonomous Agents'97,*** 16–21.

LESTER, J. C., CONVERSE, S. A., STONE, B. A., AND BHOGAL, R. S. (1997) The personal effect: affective impact of animated pedagogical agents. In ***Proceedings of CHI'97***, 359–366.

LIDDLE, D. (1996) Design of the conceptual model. In T. Winograd, (ed.) ***Bringing Design to Software.*** Reading, MA: Addison-Wesley, 17–31.

LUND, A. M. (1994) Ameritech's usability laboratory: from prototype to final design. ***Behaviour & Information Technology,*** 13(1 & 2), 67-80.

LYNCH, P. J., AND HORTON, S. (1999) ***Web Style Guide (Preliminary Version).*** New Haven, CT. and London: Yale University Press.

M880 (2000) OSS CD part of M880 ***Software Engineering.*** Milton Keynes, UK: The Open University.

MACKAY, W. E., RATZER, A. V., AND JANECEK, P. (2000) Video artifacts for design: bridging the gap between abstraction and detail. In ***Proceedings of DIS 2000***, 72–82.

MACKENZIE, I. S. (1992) Fitts' law as a research and design tool in human-computer interaction. ***Human-Computer Interaction,*** 7, 91–139.

MAES, P. (1995) Intelligent software. ***Scientific American,*** 273(3), 84-86.

MAGLIO, P. P., MATLOCK, T., RAPHAELY, D., CHERNICKY, B., AND KIRSH D. (1999) Interactive skill in Scrabble. In ***Proceedings of Twenty-first Annual Conference of the Cognitive Science Society.*** Mahwah, NJ: Lawrence Erlbaum Associates.

MAHER, M. L. AND PU, P. (1997) ***Issues and Applications of Case-Based Reasoning in Design.*** Hillsdale, NJ: Lawrence Erlbaum Associates,

MAIDEN, N. A. M. AND RUGG, G. (1996) ACRE: selecting methods for requirements acquisition. ***Software Engineering Journal,*** 11(3), 183–192.

MALONE, T. W. (1983) How do people organize their desks? Implications for the design of office information systems. ***ACM Transactions on Office Information Systems,*** 1(1) 99–112.

MANDLER, R., SALOMON, G. AND WONG, Y. Y. (1992) A 'pile' metaphor for supporting casual organization of information. In ***Proceedings of CHI'92***, 627–634.

MANN, S. (1996) Smart clothing: wearable multimedia computing and personal imaging to restore the technological balance between people and their environment. In ***Proceedings of ACM Multimedia,*** 96,163–174.

MARCUS, A. (1993) Human communication issues in advanced UIs. ***Communications of the ACM,*** 101–109.

MARK, G., FUCHS, L. AND SOHLENKAMP, M. (1997) Supporting groupware conventions through contextual awareness. In ***Proceedings of the Fifth European Conference on Computer-Supported Cooperative Work.*** Dordrecht, The Netherlands: Kluwer, 253–268.

MARMASSE, N. AND SCHMANDT, C. (2000) Location-aware information delivery with ComMotion. In ***Proceedings*** of ***Handheld and Ubiquitous Computing, Second International Symposium, HUC*** 2000, Springer-Verlag, 157–171.

MARSHALL, C., AND ROSSMAN, G. B. (1999) ***Designing Qualitative Research*** (3rd ed.). Thousand Oaks, CA: Sage Publications.

MARTIN, H. AND GAVER, B. (2000) Beyond the snapshot: from speculation to prototypes in audiophotography. In ***Proceedings of DIS 2000***, 55–65.

MATEAS, M., SALVADOR, T., SCHOLTZ, J. AND SORENSEN, D. (1996) Engineering ethnography in the home. ***Companion for CHI*** '96, ACM, 283–284.

MAYHEW, D. J. (1999) ***The Usability Engineering Lifecycle.*** San Francisco: Morgan Kaufmann.

MCLAUGHLIN, M., GOLDBERG, S. B., ELLISON, N. AND LUCAS, J. (1999) Measuring Internet audiences: patrons of an online art museum. In S. Jones (ed.) ***Doing Internet Research: Critical Issues and Methods for Examining the Net.*** Thousand Oaks, CA: Sage, 163–178.

MICROSOFT CORPORATION (1992) ***The Windows Interface, An Application Design Guide.*** Microsoft Press.

MILLER, G. (1956) The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. ***Psychological Review,*** 63, 81–97.

MILLER, L.H. AND JOHNSON, J. (1996) The Xerox Star: an influential user interface design. In M. Rudisill, C. Lewis, P. G. Polson, and T. D. McKay, (eds.) ***Human-Computer Interface Design.*** San Francisco: Morgan-Kaufmann.

MILLINGTON, D. AND STAPLETON, J. (1995) Special report: developing a RAD standard. ***IEEE Software***, 12(5), 54–6.

MONTEMAYOR, J., DRUIN, A. AND HELANDER, J. (2000) PETS: A personal electronic teller of stories.' In C.A. Druin & J. Helander (eds.) ***Robots for Kids.*** San Francisco: Morgan Kaufmann.

MORAN, T. P., AND R. J. ANDERSON (1990) The workaday world as a paradigm for CSCW design. In ***Proceedings of the CSCW*** '90,381–393.

MORIKAWA, O. AND MAESAKO, T. (1998) HyperMirror: towards pleasant-to-use video mediated communication system. In ***Proceedings of CSCW'98***, 149–158.

MOSIER, J. N. AND TAMMARO, S. G., (1997) When are group scheduling tools useful? In ***Proceedings of CSCW '97***, 6, 53–70.

MULLER, M. J. (1991) PICTIVE—An exploration in participatory design. In ***Proceedings of CHI*** '91, 225–231.

MULLER, M. J., TUDOR, L. G., WILDMAN, D. M., WHITE, E. A., ROOT, R. W., DAYTON, T., CARR, R., DIEKMAN, B., AND DYKSTRA-ERICKSON, E. (1995) Bifocal tools for scenarios and representations in participatory activities with users. In J. M. Carroll (ed.) ***Scenario-Based Design.*** New York: John Wiley & Sons, 135–163.

MULLET, K. AND SANO, D. (1995) ***Designing Visual Interfaces.*** Mountain View, CA: Prentice-Hall.

MYERS, B. A. (1995) State of the Art in User Interface Software Tools. In R. Baecker, J. Grundin, W. Buxton, and S. Greenberg (eds.) ***Readings in Human-Computer Interaction: Toward the Year*** 2000 (2nd ed.) San Francisco: Morgan Kaufmann, 344–356.

MYERS, B., HUDSON, S. E., AND PAUSCH, R. (2000) Past, present and future of user interface software tools. ***ACM Transactions on Computer-Human Interaction,*** 7(1), 3–28.

NARDI, B. A., AND O'DAY, V. L. (1999) ***Information Ecologies: Using Technology with a Heart.*** Cambridge, MA: The MIT Press.

NELSON, T. (1980) Interactive Systems and the design of Virtuality. ***Creative Computing,*** Nov.–Dec., 1980.

NELSON, T, (1990) The right way to think about software design. In B. Laurel, (ed.) ***The Art of Human-Computer Design.*** Reading, MA: Addison-Wesley.

NEWMAN, W. AND LAMMING, N. (1995) ***Interactive System Design.*** Harlow, UK: Addison-Wesley.

NIE, N. H., AND EBRING, L. (2000) ***Internet and Society. Preliminary Report.*** Stanford, CA: The Stanford Institute for the Quantitative Study of Society.

NIELSEN, J. (1992) Finding usability problems through heuristic evaluation. In ***Proceedings of CHI'92***, 373–800.

NIELSEN, J. (1993) ***Usability Engineering.*** San Francisco: Morgan Kaufmann.

NIELSEN, J. (1994a) Heuristic evaluation. In J. Nielsen and R. L. Mack (eds.) ***Usability Inspection Methods.*** New York: John Wiley & Sons.

NIELSEN, J. (1994b) Enhancing the explanatory power of usability heuristics. In ***Proceedings of ACM CHI'94***, 152–158.

NIELSEN, J. (1999) www.useit.com

NIELSEN, J. (2000) ***Designing Web Usability.*** Indianapolis: New Riders Publishing.

NIELSEN, J. (2001) Ten Usability Heuristics. www.useit.com/papers/heuristic

NIELSEN, J., AND MACK, R. L. (1994) ***Usability Inspection Methods.*** New York: John Wiley & Sons.

NODDER, C., WILLIAMS, G., AND DUBROW, D. (1999) Evaluating the usability of an evolving collaborative product––changesin user type, tasks and evaluation methods over time. In ***Proceedings of GROUP'99***, 150–159.

NOLDUS (2000) ***The Observer Video-Pro.*** www.noldus.com/products/observer/obs_spvta30.html.

NONNECKE, B., AND PREECE, J. (2000) Lurker demographics: counting the silent. In ***Proceedings of CHI 2000***, 73–80.

NORMAN, D. (1983) Some observations on mental models. In Gentner, D. and A. L. Stevens (eds.) ***Mental Models.*** Hillsdale, NJ: Lawrence Earlbaum Associates.

NORMAN, D. (1988) ***The Design*** of ***Everyday Things.*** New York: Basic Books.

NORMAN, D. (1990) Four (more) issues for Cognitive Science. ***Cognitive Science Technical Report No.*** 9001, Dept. of Cognitive Science, UCSD, USA.

NORMAN, D. (1993) ***Things That Make Us Smart.*** Reading, MA: Addison-Wesley.

NORMAN, D. (1999) Affordances, conventions and design. ***ACM Interactions Magazine,*** May/June 1999, 38-42.

NYGAARD, K. (1990) The origins of the Scandinavian school, why and how? ***Participatory Design Conference 1990 Transcript,*** Computer Professionals for Social Responsibility.

OLSON, J. S., AND MORAN, T. P. (1996) Mapping the method muddle: guidance in using methods for user interface design. In M. Rudisill, C. Lewis, P. B. Polson and T. D. McKay (eds.) *Human-Computer Interface Design: Success Stories, Emerging Methods, Real-World Context,* San Francisco: Morgan Kaufmann, pp. 269–300.

OOSTERHOLT, R., KUSANO, M., AND DE VRIES, G. (1996) Interaction design and human factors support in the development of a personal communicator for children. In *Proceedings of CHI '96*, 450–465.

OPPENHEIM, A. N. (1992) *Questionnaire Design, Interviewing and Attitude Measurement.* London: Pinter Publishers.

OREN, T., SALOMON, G., KREITMAN, K. AND DON, A. (1990) Guides: characterizing the interface. In B. Laurel (ed.) *The Art of Human-Computer Interface Design.* Reading, MA: Addison-Wesley, 367–381.

PAGE, S. R. (1996) User-centered Design in a commercial software company. In D. Wixon and J. Ramey, (eds.) *Field Methods Casebook for Software Design.* New York: John Wiley & Sons, 197–213.

PAYNE, S. (1991) A descriptive study of mental models. *Behaviour and Information Technology,* 10, 3–21.

Penpoint *hci.stanford.edu/cs147/notes/penpoint.html*

PICARD, R. W. (1998) *Affective Computing.* Cambridge, MA: MIT Press.

PLOWMAN, L., ROGERS, Y. AND RAMAGE, M. (1995) What are workplace studies for? In *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work,* Dordrecht: The Netherlands, Kluwer, 309–324.

POTTER, J. AND WETHERELL, M. (1987) *Discourse and Social Psychology.* London: Sage.

PREECE, J. (2000) *Online Communities: Designing Usability, Supporting Sociability.* Chichester, UK: John Wiley & Sons.

PREECE, J., ROGERS, Y., SHARP, H., BENYON, D., HOLLAND, S., AND CAREY, T. (1994) *Human-Computer Interaction.* Wokingham, *UK:* Addison-Wesley.

PRESSMAN, R. (1992) *Software Engineering: A Practitioner%Approach.* New York: McGraw-Hill.

QUINTANAR, L. R., CROWELL, C. R., AND PRYOR, J. B. (1982) Human-computer interaction: a preliminary social psychological analysis. *Behavior Research: Methods and Instrumentation,*13, (2), 210–220.

REEVES, B., AND NASS, C. (1996) *The Media Equation: How People Treat Computers, Television, and New Media like Real People and Places.* Cambridge: Cambridge University Press.

RETTIG, M. (1994) Prototyping for tiny fingers. *Communications of the ACM,* 37(4), 21–27.

RHODES, B., MINAR, N. AND WEAVER, J. (1999) Wearable computing meets ubiquitous computing: reaping the best of both worlds. In *Proceedings of the Third International Symposium on Wearable Computers (ISWC '99),* San Francisco, 141–149.

RIBA (1988) *Architect's Job Book: Volume 1, Job Administration* (5th edition), London: RIBA Publications.

ROBERTSON, S. AND ROBERTSON, J. (1999) *Mastering the Requirements Process.* Boston: Addison-Wesley.

ROBINSON, J. P., AND GODBEY, G. (1997) *Time for Life: The Surprising Ways that Americans Use Their Time.* University Park, PA: The Pennsylvania State University Press.

ROBSON, C. (1993) *Real World Research.* Oxford, UK: Blackwell.

ROBSON, C. (1994) *Experimental Design and Statistics in Psychology.* Aylesbury, England: Penguin Psychology

ROGERS, Y. (1993) Coordinating computer-mediated work. *Computer Supported Cooperative Work,* 1, 2995–3315.

ROGERS, Y. AND SCAIFE, M. (1998) How can interactive multimedia facilitate learning? In J. Lee (ed.) *Intelligence and Multimodality in Multimedia Interfaces: Research and Applications.* Menlo Park, CA: AAAI Press.

ROSE, A., SHNEIDERMAN, B., AND PLAISANT, C. (1995) An applied ethnographic method for redesigning user interfaces. In *Proceedings of DIS* 95,115–122.

ROTH, I. (1986) An introduction to object perception. In I. Roth and J.B. Frisby (eds.) *Perception and Representation: A Cognitive Approach.* Milton Keynes: Open University.

RUBIN, J., (1994) *Handbook of Usability testing: How to Plan, Design and Conduct Effective tests.* New York: John Wiley & Sons.

RUBINSTEIN, R. AND HERSH, H. (1984) *The Human Factor: Designing Computer Systems for People.* Woburn, MA: Digital Press.

RUDD, J., STERN, K. R. AND ISENSEE, S. (1996) Low vs. High-fidelity Prototyping Debate. *ACM Interactions Magazine,* January, 76–85.

RUDMAN, C. AND ENGELBECK, G. (1996) Lessons in choosing methods for designing complex graphical user interfaces. In M. Rudisill, C. Lewis, P. B. Polson and T. D. McKay (eds.). *Human-Computer Interface Design: Success Stories, Emerging Methods, Real-World Context.* San Francisco: Morgan Kaufmann, 198–228.

SACKS, H., SCHEGLOFF, E., AND JEFFERSON, G. (1978) A simplest systematics for the organization of turn-taking for conversation. *Language,* 50, 696–735.

SCAIFE, M. AND ROGERS, Y. (1996) External cognition: how do graphical representations work? *International Journal of Human-Computer Studies,* 45,185–213.

SCAIFE, M., AND ROGERS, Y. (2001) Informing the design of virtual environments. *International Journal of Human-Computer Systems,* 55(2), 115–143.

SCAIFE, M., ROGERS, Y., ALDRICH, F., AND DAVIES, M. (1997) Designing for or designing with? Informant design for interactive learning environments. In *Proceedings of CHI '97,* 343–350.

SCHANK, *R.* C. (1982) *Dynamic Memory: a Theory of Learning in Computers and People.* Cambridge, UK: Cambridge University Press.

SCHÖN, D. (1983) *The Reflective Practitioner: How Professionals Think in Action.* New York: Basic Books.

SCHRAGE, M. (1996) Cultures of prototyping. In T. Winograd (ed.) *Bringing Design to Software.* Boston: Addison-Wesley.

SEARLE, J. (1969) *Speech Acts.* Cambridge: Cambridge University Press.

SEGALL, B., AND ARNOLD, D. (1997) Elvin has left the building: A publish/subscribe notification service with quenching. In *Proceedings of AUUG Summer Technical Conference,* Brisbane, Australia.

SHACKEL, B. (1990) Human factors and usability. In J. Preece and L. Keller (eds.) *Human-Computer Interaction: Selected Readings.* Hemel Hempstead, *UK:* Prentice-Hall, 27-41.

SHAPIRO, D. (1995) Noddy's guide to . . . ethnography and HCI. *HCI Newsletter* 27, 8–10.

SHARF, B. F. (1999) Beyond netiquette: the ethics of doing naturalistic discourse research on the Internet. In S. Jones (ed.) *Doing Internet Research: Critical issues and methods for examining the net.* Thousand Oaks, CA: Sage Publications, 243–256.

SHARP, H. C., ROBINSON, H. M., AND WOODMAN, M. (1999) The role of culture in successful software process improvement. In *EUROMICRO '99, Proceedings of 25th EUROMICRO Conference.* Piscataway, NJ: IEEE Press, II, 170–176.

SCHEGLOFF, E. A., AND SACKS, H. (1973) Opening up closings. *Semiotica,* 7,289–327.

SHNEIDERMAN, *B.* (1983) Direct manipulation: a step beyond programming languages. *IEEE Computer,* 16(8), 57–69.

SHNEIDERMAN, B. (1998) *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (3rd ed.). Reading, MA: Addison-Wesley.

SHNEIDERMAN, B. (1998a) Relate-Create-Donate: A teaching philosophy for the cyber-generation. *Computers in Education,* 31(1), 25–39.

SILFVERBERG, M., MACKENZIE, I. S., AND KORHONEN, P. (2000) *Predicting text entry speed on mobile phones.* In *Proceedings of* CHI'2000, 9–16.

SMITH, *D.,* IRBY, C., KIMBALL, R., VERPLANK, B. AND HARSLEM, E. (1982) Designing the Star user interface. *Byte,* 7(4), 242–82.

SMITH, S. L. AND MOSIER, J. N. (1986) *Guidelines for Designing User Interface Software.* Report ESD-TR-86–278, Electronic Systems Division, Bedford, MA: The Mitre Corporation.

SOMMERVILLE, I. (2001) *Software Engineering* (6th ed.) Boston and Harlow, UK: Addison-Wesley.

SPENCER, R. (2000) The streamlined cognitive walkthrough method: working around social constraints encountered in a software development company. In *Proceedings of CHI* 2000,253–359.

SPIEGEL, D., BLOOM, J. R., KRAEMER, H. C., AND GOTTHEIL, E. (1989) Effect of psychosocial treatment on survival of patients with metastatic breast cancer. *The Lancet,* October 4,888–891.

SPREENBERG, P., SALOMON, G., AND JOE, P. (1995) Interaction design at IDEO product development. In *Proceedings of ACM CHI'95 Conference Companion,* 164–165.

SPROULL, L., SUBRAMANI, M. M., KIESLER, S., WALKER, J. H., AND WATERS, K. (1996) When the interface is a face. *Human-Computer Interaction,* 11, 97–124.

STROMMEN, E. (1998) When the interface is a talking dinosaur: learning across media with ActiMates Barney. In *Proceedings of CHI'98,* 288–295.

SUCHMAN, L. A. (1983) Office procedures as practical action: models of work and system design. *ACM Transactions on Office Information Systems,* 1(4), 320–328.

SUCHMAN, L. A. (1987) ***Plans and Situated Actions.*** Cambridge: Cambridge University Press.

SULLIVAN, K. (1996) Windows 95 user interface: A case study in usability engineering. In ***Proceedings of CHI '96***, 473–480.

TAYLOR, A. (2000) IT projects: sink or swim. ***The Computer Bulletin,*** January, 24–26.

TEASLEY, B., LEVENTHAL, L., BLUMENTHAL, B., INSTONE, K., AND STONE, D. (1994) Cultural diversity in user interface design. ***SIGCHI Bulletin,*** 26(1), 36–40.

THIMBLEBY, H. (1990) ***User Interface Design.*** Harlow, UK: Addison Wesley.

TRACTINSKY, N. (1997) Aesthetics and apparent usability: empirically assessing cultural and methodological issues. In ***Proceedings of CHI'97***, 115–122.

TUDOR, L. G. (1993) A participatory design technique for high-level task analysis, critique and redesign: The CARD method. In ***Proceedings of the Human Factors and Ergonomics Society 1993 Meeting,*** Seattle, October 1993,295–299.

VÄÄNÄNEN-VAINIO-MATTILA, K. AND RUUSKA, S. (2000) Designing mobile phones and communicators for consumers' needs at Nokia. In E. Bergman (ed.) ***Information Appliances and Beyond.*** San Francisco: Morgan Kaufmann, 169–204.

VEEN, J. (2001) ***The Art and Science of Web Design.*** Indianapolis: New Riders Publishing.

VERPLANK, B. (1989) Tutorial Notes. In ***Proceedings of CHI'89 Conference.***

VERPLANK, B. (1994) Interview with Bill Verplank. In PREECE, J., ROGERS, Y., SHARP, H., BENYON, D., HOLLAND, S., AND CAREY, T., ***Human-Computer Interaction.*** Wokingham, UK: Addison-Wesley, 467–468.

VILLER, S. AND SOMMERVILLE, I. (1999) Coherence: an approach to representing ethnographic analyses in systems design. ***Human-Computer Interaction,*** 14.

WALKER, J., SPROULL, L., AND SUBRAMANI, R. (1994) Using a human face in an interface. In ***Proceedings of CHI'94,*** 85–91.

WEBB, B. R. (1996) The role of users in interactive systems design: when computers are theatre, do we want the audience to write the script? ***Behaviour and Information Technology,*** 15(2), 76-83.

WEISER, M. (1991) The computer for the 21st Century. ***Scientific American,*** 265 (3), 94–104.

WELLNER, P. (1993) Interacting with paper on the digital desk. ***Communications of the ACM,*** 36(7), 86–96.

WHARTON, C., RIEMAN, J., LEWIS, C., AND POLSON, P. (1994) The cognitive walkthrough method: a practitioner's guide. In J. Nielsen and R. L. Mack (eds.), ***Usability Inspection Methods.*** New York: John Wiley & Sons.

WHITESIDE, J., BENNETT, J. AND HOLTZBLATT, K. (1988) Usability engineering: our experience and evolution. In ***Handbook of Human-Computer Interaction.*** Helander, ***M.*** (ed.) Amsterdam: Elsevier Science Publishers, 791–817.

WHITTAKER, S., AND SCHWARTZ, H. (1995) Back to the future: pen and paper technology supports complex group coordination. In ***Proceedings of CHI'95,*** 495–502.

WILLIAMS, F., RICE, R. E., AND ROGERS, E. M. (1988) ***Research Methods and the New Media.*** New York: ***The*** Free Press, Macmillan Inc.

WITMER, D. F., COLMAN, R. W., AND KATZMAN, S. L. (1999) From paper-and-pencil to screen-and-keyboard. In S. Jones (ed.) ***Doing Internet Research: Critical Issues and Methods for Examining the Net.*** Thousands Oaks, ***CA:*** Sage, 145–161.

WINOGRAD, T. (1988) A languagelaction perspective on the design of cooperative work. ***Human-Computer Interaction,*** 3, 3–30.

WINOGRAD, T. (1994) Categories, disciplines, and social coordination. ***Computer Supported Cooperative Work,*** 2,191–197.

WINOGRAD, T. (1996) (ed.) ***Bringing Design to Software.*** Reading, MA: Addison-Wesley.

WINOGRAD, T. (1997) From computing machinery to interaction design. In P. Denning and R. Metcalfe (eds.) ***Beyond Calculation: the Next Fifty Years of Computing.*** Amsterdam: Springer-Verlag, 149–162.

WINOGRAD, T. AND FLORES, W. (1986) ***Understanding Computers and Cognition.*** Norwood, NJ: Addison-Wesley.

WIXON, D., AND WILSON, C. (1997) The usability engineering framework for product design and evaluation (Chapter 27). In M. G. Helander, T. K. Landauer, and P. V. Prabju (eds.) ***Handbook of Human-Computer Interaction.*** Amsterdam, Holland: Elsevier, 653–688.

WOOD, J. AND SILVER, D. (1995) ***Joint Applications Development*** (2nd ed.) New York: John Wiley & Sons.

Credits

Chapter 1

Figure 1.1: after Gillian Crampton Smith, The hand that rocks the cradle, ***ID Magazine,*** May/June 1995; Figure 1.2 (on Color Plate 1) (i): gif from www.electrolux.com/screenfridge/start.html, reproduced by permission of AB Electrolux; Figure 1.2(ii): gif from http://houns54.clearlake.ibm.com / solutions/media/medpub.nsf/ebrcs/Ask_a_Question? OpenDocument reproduced by permission of IBM; Figure 1.2(iii): gif from http://www.research. philips.com/pressmedia/pictures/passw3.html, copyright © Philips Research, reproduced by permission of Philips Research; Figure 1.4: figure under section heading 32.1 Interdisciplinary Cooperation, Chapter by S.Kim in ***The Art of Human Interface Design,*** edited by B. Laurel (1990), Addison Wesley; Figure 1.5: gif from www.ideo. com/studies/scout.htm, reproduced by permission of IDEO; Figure 1.6(a) and (b): screenshots from www.qualcomm.com/eudora reproduced by permission of QUALCOMM Eudora Products; Figure 1.8: screenshot of Photoshop™ menu reproduced by permission of Adobe Systems Incorporated; Table 1: reproduced by permission of www.useit.com/papers/heuristic/heuristic-list.html, copyright © Jakob Nielsen. All Rights Reserved. Fig 1. Interview: reproduced by permission of IDEO.

Chapter 2

Figure 2.1 (on Color Plate 2): gif from www.ai.mit. edu/projects/medical-vision/surgery/ surgical_navigation.html reproduced by permission of Michael E. Leventon; Figure 2.6(a): gif from http://vibes.cs.uiuc.edu/Project/VR/Virtue/VirtueOverview.htm, reproduced by permission of Dr Daniel A. Reed (University of Illinois at Urbana-Champaign) from work on the Collaborative Virtual Environments for Direct Software Manipulation research project, supported in part by the Defense Advanced Research Projects Agency under contract numbers DABT63-94-C0049, F30602-96-C-0161, DABT63-96-C0027, N66001-97-C-8532, in part by the National Science Foundation under grants CDA 94-01124 and ASC 97-20202, and in part by the Department of Energy under contracts B-341494, W-7405-ENG-48, and 1-B-333164; Figure 2.5: The Finder Desktop from ***Apple Human Interface Guidelines,*** Apple Computer Inc. (1987), Addison Wesley; Figure 2.6(b) (on Color Plate 3): gif from http://www.evl.uic.edu/pape/projects/crayoland/big/, copyright © 1997 Dave Pape, image courtesy of the Electronic Visualization Laboratory, University of Illinois at Chicago; Figure 2.7: gif of annotated screen dump for Visicalc™ used with permission of Lotus Development Corporation—Visicalc is a trademark of Lotus Development Corporation; Figure 2.8: Johnson, J. ***et al.***, The Xerox "Star": a retrospective, in ***IEEE Computer,*** copyright © 1989 IEEE, reproduced by permission of IEEE; Figure 2.9: Figure 1.10 (page 16) from ***The Psychology of Everyday Things,*** by Donald A. Norman, copyright © 1988 by Donald A. Norman, reprinted by permission of Basic Books, a member of Perseus Books, L.L.C.; Figure 2.10: Figure 32 (page 33) from ***Designing Visual Interfaces*** by ***K.*** Mullett and D. Sano © 1995 reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ 07458; Figure 2.11(i): gif from http://tangible.media.mit.edu/papers/ Tangible_Bits_CHI97.html, Ishii, H. and Ullmer, B. (1997) Tangible Bits: towards seamless interfaces, in ***CHI'97 Proceedings,*** reprinted by permission of Association for Computing Machinery, Inc.; Figure 2.11(ii) gif from www.almaden.ibm.com/cs/blueeyes/ reproduced by permission of IBM; Figure 2.11(iii): gif from www.parc.xerox.com/red/members/richgold/ livingdoc/slide6. html, reproduced by permission of Rich Gold of PARC Communications; Figure 2.12: gif from www.mbay.net/~brendah/articles/PDA. Mar.95/ reproduced by permission of General Magic, Inc.; Figure 2.13(b): gif from **http:l/thesims.ea.com/us/** reproduced by permission of Electronic Arts Inc. © 2001 Electronic Arts Inc., all rights reserved; Figure 2.14 (on Color Plate 2): gif from http://graphics. stanford.EDU/projects/iwork/ reproduced by permission of Professor Terry Winograd; ***Cartoon:*** Copyright © CartoonStock, www.CartoonStock.com.

Chapter 3

Figures 3.2(a) and (b): two screenshots of lodging information reproduced by permission of T. S. Tullis from his Ph.D. Dissertation *Predicting the Usability of Alphanumeric Displays*, Rice University, Houston, Texas, USA; Figures 3.3 and 3.10: screenshots of Google search engine reproduced by permission of Google Inc.; Figure 3.4: summarized text from page 192 from *Designing Visual Interfaces* by *K.* Mullett and D. Sano © 1995 reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ 07458; Figure 3.6: Lonsdale and Edmunds (1992) *International Journal of Human Computer Studies,* 26, 97–126, Figure 3, reproduced by permission of Academic Press Ltd; Figure 3.8: Mander, R., Salomon, G. and Wong, Y. (1992) Figure 6 (page 631) in *CHI'92 Proceedings,* reprinted by permission of Association for Computing Machinery, Inc.; Figure 3.9 (on Color Plate 4): gif of a transparent phone reproduced by permission of Lazerbuilt Limited; Figure 3.11: redrawn and adapted from Barber, P. (1988) *Applied Cognitive Psychology,* Figure *3.1* (page 63) published by Routledge and reproduced by permission of ITPS Ltd; Figure 3.12: Card, S., Moran, T. and Newell, A. (1983) *The Psychology of HCI*, Figure 2.1, page 26, reproduced by permission of Lawrence Erlbaum Associates, Inc.; Figure 3.13: reproduced courtesy of Lucent Technologies Inc. © [1997] Lucent Technologies Inc., all rights reserved; *Cartoon:* Reproduced by permission of Randy Glasbergen.

Chapter 4

Figure 4.1 (on Color Plate 5): Three gifs of BowieWorld from www.worlds.com/bowie reproduced by permission of worlds.com; Figure 4.2: reprinted from *Decision Support Systems,* 5(2), Nunamaker, *J. et al.*, Experiences at IBM with group support systems, 183–196, Figure 2 © 1989, with permission from Elsevier Science; Figure 4.3: gif of Willow Tree ACTIVboard reproduced by permission of Promethean Ltd.; Figure 4.4(a): photograph of an early model of a videophone (prototype) by courtesy of BT Archives; Figure 4.4(b): photograph of the VP-210 VisualPhone reproduced by permission of Kyocera Corporation, © 1999 Kyocera Corporation; Figure 4.5: illustration of the Video Window System in use from Kraut, R. E., Root, R. W. and Chalfonte, B. L. (1990) Informal communication in organisations (pages 145–199) in Oskamp, S. and Spacapan, S. (eds.) *People's Reactions to Technologies in Factories, Offices and Aerospace–The Claremont Symposium on Applied Psychology* copyright © 1990 Sage Publications, reprinted by permission of Sage Publications Inc.; Figure 4.7: Morikawa, O., Yamashita, J. and Fukui, Y. (2000) The sense of physically crossing paths, Figure 1 (page 183) in *CHI2000 Proceedings,* reprinted by permission of Association for Computing Machinery, Inc.; Figure 4.8: *Computer Supported Cooperative Work Journal,* 1,303, Rogers, *Y.* Figure 3, reproduced with kind permission from Kluwer Academic Publishers; Figure 4.10: reproduced by permission of the Xerox Research Centre Europe; Figure 4.11: *ECSCW* (1999) 438, Augmenting the workaday world, Fitzpatrick, G. *et al.*, Figures 4 and 5, reproduced with kind permission from Kluwer Academic publishers and the authors; Figure 4.12: Erickson, T. *et al.* (1999) Socially translucent systems, Figure 2 (page 74) in *CHI'99 Proceedings,* reprinted by permission of Association for Computing Machinery, Inc.; Figure 4.13: Winograd, T. and Flores, W. (1986) *Understanding Computers and Cognition,* Figure 5.1 (page 65), Addison Wesley: Figure 4.14: Winograd, T. (1988) Where the action is, Table A (page 257) in BYTE, reproduced by permission of CMP Media LLC and Byte.com; Figure 4.15: after Halverson, C., Inside the cognitive workplace: new technology and air traffic control. PhD Thesis, U. of California, San Diego (1995); Figure 4.16: Preece, J. and Keller, L. (1994) *Human-Computer Interaction,* Figure 3.5 (page 70) © Selection and editorial material, the Open University, reprinted by permission of Pearson Education Ltd.; *Cartoon:* Copyright © CartoonStock, www.CartoonStock.com.

Chapter 5

Figure 5.1: gif from www.ai.mit.edu/projects/humanoid-robotics-group/kismet/kismet-html reproduced by permission of Peter Menzel Photography; Figure 5.2: Figure 40 (page 40) from *Designing Visual Interfaces* by K. Mullett and D. Sano © 1995 reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ 07458; Figure 5.3 (on Color Plate 6) (i): photograph of an iMac from www.apple.com/hardware reproduced by permission of Mark Laita; Figure 5.3(ii): screenshot of

a Nokia mobile phone from www.nokia.com/phones reproduced by permission of Nokia Corporation; Figure 5.3(iii): gif fsom www.ideo.com/studies/bbc.htm reproduced by permission of IDEO; Figures 5.4(a) and (b): Marcus, A. (1993) Human communication in advanced uls, Figures 2 and 4 (pages 106 and 107) in ***Communications of the ACM,*** 36(4), 101–109, reprinted by permission of Association for Computing Machinery, Inc.; Figure 5.5: Figure 7.2 (page 147) in ***Bringing Design to Software,*** edited by Winograd, T. (1996), Addison Wesley; Figure 5.8: from Oren, T., Salomon, G. *et al.*, Guides: Characterizing the Interface, Figure 6 (page 370) in ***The Art of Human Interface Design*** edited by Laurel, B. (1990), Addison Wesley; Figure 5.9 (on Color Plate 6) (i): gif of Aibo from www.newscast.co.uk reproduced by permission of Sony Corporation; (ii): screenshot of www.ananova.com showing *Ananova*, the virtual news presenter, © Ananova Ltd. 2001, reproduced by permission of Ananova Ltd., all rights reserved; (iii): screenshot from www.e-cyas.com of E-cyas avatar reproduced by permission of I-D Media Ltd.; Figure 5.10: gifs from alive.www.media.edu/projects/alive reproduced by permission of Professor Bruce Blumberg; Figure 5.11: gif from www.csc.ncsu.edu/eos/users/1/lester/www/imedia/DAP.html reproduced by permission of Professor James Lester; Figure 5.12 (on Color Plate 7): gif from www.cs.cmu.edu/afs/cs.cmu.edu/project/oz/web/woggles_clr.html reproduced by permission of Joseph Bates, Zoesis Studios; Figure 5.13 (on Color Plate 8): gif from http:llgn.www.media.mit.edu/groups/gn/projects/humanoid/ reproduced by permission of Professor Justine Cassell; Figure 5.14: Figure 2 (page 365) in ***The Art of Human Interface Design*** edited by Laurel, B. (1991), Addison Wesley.

Chapter 6

Figures 6.2–6.4: reproduced by permission of IDEO, photographs by Jorge Davies; Figures 6.5 and 6.6: Cusumano, M. and Selby, R. (1997) How Microsoft builds software, Figures 2 and 3 (pages 56 and 57) in ***Communications of the ACM,*** 40(6) reprinted by permission of Association for Computing Machinery, Inc.; Figure 6.9: Boehm, B. W. A spiral model of software development and enhancement, ***IEEE Computer,*** 21 (5), Figure 2 (page 64) reproduced by permission of IEEE C1988 IEEE; Figures 6.11 and 6.12: Isensee, S. ***et al.*** Designing internet appliances at Netpliance from ***Information Appliances and Beyond*** (2000) edited by Bergman, E., Figures 3.2 (page 58) and 3.6 (page 71) reproduced by permission of Academic Press Inc.; Figure 6.13: Hartson, H. R. and Hix, D. (1989) How Microsoft builds software, ***International Journal of Man-Machine Studies,*** 31, 477–494, the Star lifecycle model, reproduced by permission of Academic Press Ltd.; Figure 6.14: The usability engineering lifecycle figure in ***The Usability Engineering Lifecycle*** by Mayhew, D. J. (1999) reproduced by permission of Academic Press Inc.; ***Cartoon:*** Copyright © CartoonStock, www.CartoonStock.com.

Chapter 7

Figures 7.1 and 7.5: Robertson, S. and Robertson, J. (1999) ***Mastering the Requirements Process,*** Figures 10.3 (page 184) and 1.3 (page 9) © Pearson Education Ltd 1999, reprinted by permission of Pearson Education Ltd.; Figure 7.2: Bergman, E. and Haitani, R. (2000) Designing the PalmPilot: a conversation with Rob Haitani, from ***Information Appliances and Beyond,*** (edited by Bergman, E.) Figure 4.3 (page 86) reproduced by permission of Academic Press Inc.; Figure 7.3(a) photograph of the KordGrip reproduced by permission of WetPC Pty. Ltd., Australia 2605; Figure 7.3(b) (on Color Plate 8): photograph of the KordGrip being used under water by permission of the Australian Institute of Marine Science; Figure 7.4: Gaver, B., Dunne, T. and Pacenti, E. (1999) Cultural probes, Figure 1 (page 22) in ***Interactions*** (January/February) reprinted by permission of Association for Computing Machinery, Inc.; Figure 7.7: screenshot reproduced by permission from Symbian–http://www.symbian.com;figure in Suzanne Robertson's interview reproduced by permission of The Atlantic Systems Guild Ltd.; ***Cartoon:*** © The 5th Wave, www.the5thwave.com.

Chapter 8

Figure 8.1: reprinted with kind permission of Sigil Khwaja; Figure 8.2: Figure 8.2 (page 169) in ***Bringing Design to Software,*** edited by Winograd, T. (1996), Addison Wesley; Figure 8.5: Buchenau, M. and Suri, J. F. (2000) Experience prototyping, Figure 1, in Boyarski, D. and Kellogg, W. (eds.) ***DIS 2000–Design Interactive Systems, Processes, Practices, Methods, Techniques, Conference Proceedings,*** reprinted by permission of Association for Computing Machinery,

Inc.; Figure 8.6(a) and (b): photographs reproduced by permission of ICE Ergonomics Ltd., Loughborough, UK; Figure 8.7: text quoted from Mayhew, D. (1999) ***The Usability Engineering Lifecycle,*** pages 212–214, reproduced by permission of Academic Press Inc.; Figure 8.8: reprinted from ***Interacting with Computers,*** 13 (1) Bodker, S. Scenarios in user-centred design–setting the stage for reflection and action, Figure 2 (page 70), © 2000 with permission from Elsevier Science; Figure 8.12: an excerpt from BS-EN-IS0 9241 concerning how to group items in a menu reproduced by permission of the British Standards Institute; Figure 8.14: screenshot of "arrange a meeting" icon from http://www.palm.net/Registration/RegistrationAdd.jsp reproduced by permission of Palm, Inc.; Figure 8.15: reproduced by permission of New Riders Publishing, copyright © 2001 Jeffrey Veen, from the book ***The Art and Science of Web Design*** by Jeffrey Veen; Figure 8.16: screenshot of the front web page of the Aftonbladet Newspaper from http://www.aftonbladet.se reproduced by permission of Aftonbladet Nya Medier; ***Cartoon:*** Copyright © CartoonStock, www.CartoonStock.com.

Chapter 9

Figures 9.1–9.3: Tables 1–3 (pages 7,8), Tables 4–7 (pages 9,10), Table 9 (page 15) from Viller, S. and Somerville, I. (1999) Coherence: an approach to representing ethnographic analyses in systems design, ***Human-Computer Interaction,*** 14 (special issue on representations in interactive systems and development) reproduced by permission of Lawrence Erlbaum Associates, Inc.; Figures 9.4–9.8: Figure 11.5 (page 206), Figure 17.4 (page 315), Figure 17.5 (page 316), Figure 17.2 (page 312), Figure 17.3 (page 313) from Wixon, D. and Ramey, J. (eds.) ***Field Methods Casebook for Software Design,*** © 1996 John Wiley & Sons, Inc., reprinted by permission of John Wiley & Sons, Inc.; Figure 9.9: Beyer, H. and Holtzblatt, K. (1998) ***Contextual Design,*** Figure 9.1 (page 155) reproduced by permission of Academic Press, Inc.; Figure 9.10: Ehn, P. and Kyng, M. (1991) Cardboard computers: mocking-it-up or hands-on the future, sort machine mock-up (page 175) in ***Design at Work: Cooperative Design of Computer Systems*** (Greenbaum, J. and Kyng, M., eds.) reproduced by permission of Lawrence Erlbaum Associates, Inc.; Figure 9.11: Muller, M. J. (1991) PICTIVE–an exploration in participatory design, Figures 1 and 2 (page 26) in ***CHI'91 Proceedings,*** reprinted by permission of Association for Computing Machinery, Inc.; Figure 9.12: Muller, M. J. ***et al.*** (1995) Bifocal tools for scenarios and representations in participatory activities with users, Figure 6.3 (page 149) in ***Scenario-based Design*** (Carroll, J., ed.) © John Carroll, reproduced by permission of John Carroll, Virginia Tech.; ***Cartoon:*** Reproduced by permission of Randy Glasbergen.

Chapter 10

Figures 10.1 and 10.2: Gould, J. D. ***et al.*** (1990) The 1984 Olympic Message System–a test of behavioral principles of system design, in Preece, J. and Keller, L. (eds.) ***Human-Computer Interaction (Readings)*** Figures 12.4 (page 265) and 12.1 (page 263) © Selection and editorial material, the Open University, reprinted by permission of Pearson Education Ltd.; Figures 10.3–10.8: Figure 1 (page 6), Appendix A of Usability study, Figure 3 (page 10), Appendix B (pages 14, 15) of Usability study, Table 3 (page 6) of Usability study, Summary (page 8) of Usability study from Cheng, L. ***et al.*** (2000) Hutchworld: lessons learned. A collaborative project: Fred Hutchsinson Cancer Research Center and Microsoft Research, ***Virtual Worlds Conference 2000, Paris, France*** © Springer-Verlag GmbH & Co., reproduced by permission of Springer-Verlag GmbH & Co. and the author.

Chapter 11

Cartoon: Reproduced by permission of Randy Glasbergen.

Chapter 12

Figures 12.1 and 12.2: screenshots from http:llwww.northernlight.com reproduced by permission of Northern Light Technology, Inc.; Figure 12.3: Figure 5 (pages 7 and 8) from Hochheiser, H. and Shneiderman, B. (2001) Using interactive visualizations of WWW log data to characterize access patterns and inform site design, ***Journal of the American Society for Information Science*** (in press) reproduced by permission from University of Maryland, Human-Computer Interaction Lab; Cartoon: HERMAN ® is reprinted with permission from Laughingstock Licensing Inc., Ottawa, Canada, all rights reserved.

Chapter 13

Figure 13.1: screenshot from http:llananova.com © Ananova Ltd. 2001, reproduced by permission of Ananova Ltd., all rights reserved; Figure 13.3: B. Shneiderman (1998) *Designing the User Interface: Strategies for Effecive Human-Computer Interaction,* Third Edition, Table 4.1, Part 3 (page 136), Addison Wesley; Figure 13.4: from Andrews *et al.*, A Conceptual Framework framework for demographic groups resistant to online community interaction. In *Proceedings of IEEE Hawaiian International Conference on System Science (HICSS),* 2001; Figure 13.5: Nielsen, J., Finding Usability Problems through Heuristic Evaluation. In *Proceedings of CHI'92*, 373–800; Figure 13.6: Adapted from Appendix G. page 204 (2001) Ph.D. Thesis by Dorine C. Andrews, 'Computer-Supported Social Networks: Audience-Centric Online Community Implementation.' Communications Design. University of Baltimore, Maryland; Figure 13.7: Figure 2.2 (page 33) from Nielsen, J. and Mack, R. L. (1994) *Usability Inspection Methods,* © 1994, John Wiley & Sons, Inc., reprinted by permission of John Wiley & Sons, Inc.; Figures 13.7–13.9: Figures 1–3 (pages 11, 12 and 14) from Cogdill, K. (1999) *MEDLINEplus Interface Evaluation: Final Report,* reproduced by permission of Professor Keith Cogdill, College of Information Studies, University of Maryland; Figure 13.10: screenshot from http:lIREI.com reproduced by permission of Recreational Equipment, Inc.; *Cartoon:* © The 5th Wave, www.the5thwave.com.

Chapter 14

Figure 14.1: Figure 1 (page 11) from Cogdill, K. (1999) *MEDLINEplus Interface Evaluation: Final Report,* reproduced by permission of Professor Keith Cogdill, College of Information Studies, University of Maryland; Figure 14.2: Figure 2, pages 67–80, from Lund, A.M. Ameritech's usability laboratory: from prototype to final design, *Behaviour and Information Technology,* 13, 1–2 (1994) (http://www.tandf.co.uk/journals) reproduced by permission from Taylor & Francis Ltd.; Figure 14.3: Nodder, C., Williams, G. and Dubrow, D. (1999) Evaluating the usability of an evolving collaborative product-changes in user type, tasks and evaluation methods over time, Figure 6 (page 156) in *GROUP'99, Phoenix, Arizona, USA,* reprinted by permission of Association for Computing Machinery, Inc.; Figure 14.4: Larson, K. and Czerwinski, M. (1998) Web page design: implications of memory, structure and scent for information retrieval, Figure 1 (page 28), in *CHI'98 Proceedings,* reprinted by permission of Association for Computing Machinery, Inc.; *Cartoon:* From *The Wall Street Journal*—Permission, Cartoon Features Syndicate.

Chapter 15

Figure 15.1: screenshot of the Nokia 9210 Communicator from http:l/www.nokia.com/press/photo/phones/jpeg/9210_09.jpg reproduced by permission of Nokia Corporation; Figures 15.2–15.5; Figure 7.11 (page 195), an example usage scenario (page 181), Figures 7.6 and 7.7 (pages 183 and 186) from Vaananen-Vainio-Mattila, K. and Ruuska, S. (2000) Designing mobile phones and communicators for consumers' needs at Nokia, *Information Appliances and Beyond* (Bergman, *E., ed.)* reproduced by permission of Academic Press, Inc.; Figures 15.6–15.10, including Figure 15.8 (on Color Plate 8) and 15.14: Oosterholt, R., Kusano, M. and de Vries, G. (1996) Interaction design and human factors support in the development of a personal communicator for children, Figures 1, 2, 3, 5, 9, 10 and 7 in *CHI'96 Proceedings,* reprinted by permission of Association for Computing Machinery, Inc., communicator concept development and execution by Philips Design, Eindhoven, The Netherlands; Figures 15.11–15.13: Figure 19 (page 28), Table 2 (pages 24 and 25) and Figure 16 (page 25) from Montemayor, J. *et al.* (2000) PETS: A personal electronic teller of stories, *Robots for Kids* (Druin, C.A. and Helander, J., eds.) reproduced by permission of Academic Press, Inc. and the authors, Institute for Advanced Computer Studies, University of Maryland.

The publisher has made every attempt to obtain permission to reproduce material in this book from the appropriate source. If there are any errors or omissions please contact the publisher, who will make suitable acknowledgement when the book is reprinted,

Index

Page references followed by italic t indicate material in tables. Page references followed by italic n indicate material in footnotes.