

HUMAN-COMPUTER INTERACTION

LEARNING OBJECTIVES

Once you have mastered the material in this chapter you will be able to:

1. Understand human-computer interaction.
2. Know how fit affects performance and well-being.
3. Understand the technology acceptance model (TAM) and usability.
4. Know how to design for the cognitive styles of individuals and for persons with disabilities.
5. Understand the different types of, and reasons for, using alternative user interfaces.
6. Design effective dialog for HCI.
7. Understand the importance of user feedback.
8. Articulate HCI implications for designing ecommerce Web sites.
9. Formulate queries that permit users to search the Web.

Throughout the book your awareness of human-computer interaction (HCI) and its importance to your task as a systems analyst has grown. Your attentiveness to the surrounding issues of HCI and its existence in organizational settings should have been heightened. While awareness is important, by now you recognize that you need to master the concepts surrounding HCI as well as become proficient at assessing human information requirements and incorporating your findings into your designs. Furthermore, the European Union (EU) and the United States have come forth with specific guidelines for usability. These guidelines mandate making Web sites and electronic services accessible to the disabled.

This chapter fills in some of the details about HCI and working with users. It also gives you some experience in applying HCI concepts that you have been learning to help in your design of human-computer interfaces; feedback, ecommerce Web sites, and Web queries.

UNDERSTANDING HUMAN-COMPUTER INTERACTION

Designing for HCI means “Ensuring system functionality and usability, providing effective user interaction support, and enhancing a pleasant user experience.” Furthermore, “The overarching goal is to achieve both organizational and individual

user effectiveness and efficiency. To reach these goals, managers and developers need to be knowledgeable about the interplay among users, tasks, task contexts, information technology (IT), and the environments in which systems are used,” (Carey et al., 2004, p. 358).

How do we ensure that our systems are user centered, so that they appropriately include users’ needs as well as organizational needs? One way is to understand HCI concepts, another is to consider interfaces in the light of HCI issues, and another is to apply standard design concepts to computers in new ways because of an HCI approach.

Knowledge about the interplay among users, tasks, task contexts, IT, and the environments in which the systems are used comprises the basis of human-computer interaction. The main tactic of HCI in systems analysis and design is to repeatedly elicit feedback from users about their experiences with prototyped designs (which could be screens, forms, interfaces, and the like), refining the design based on the suggested changes, trying them with users again until the design is acceptable, and until it is frozen by the analyst.

HOW FIT AFFECTS PERFORMANCE AND WELL-BEING

Let’s begin our exploration of human-computer interaction with some useful definitions that are commonly shared among those working in the field.

Fit A good fit between the HCI elements of the human, the computer, and the task that needs to be performed leads to performance and well-being, as shown in Figure 14.1. Just as it is important that new shoes comfortably fit the shape of your foot, hold up during the activity you will be doing (such as running), and are made of a material (such as leather) that is durable and cost-effective, so too is it important that the fit among the user, computer, and task all correspond.

Analysts want the best fit in their design. You want to make the best possible use of people in designing a computerized task that is intended to meet an organizational objective. Better fit is meant to result in better performance and greater overall well-being for the human involved in the system.

Fortunately, humans’ capacity to learn better ways to work also influences the fit. We would never try running a marathon with a shoe right out of the box, without first getting our foot used to it by breaking it in. By the same token, users can be trained to develop a better fit by learning their tasks and computers thoroughly. Training continues to be an important way to improve fit. Chapter 17 contains more specifics about how the analyst can facilitate user training for new systems and software.

Task In the foregoing chapters you have learned many methods to help you thoroughly understand, document, and graphically depict the tasks that people currently perform in the organization. You have also learned methods to help you design new tasks that will help them reach their objectives with the new systems you are creating. As you recall, tasks can be structured and routine, or they can be ill defined and without apparent structure. Complex tasks that require human, system, and task interaction are supported by ecommerce and Web systems, ERP systems, and wireless systems inside and outside the organization.

Performance The definition of the word *performance* in the HCI context is also key. In this case, “performance” refers to a combination of the efficiency involved in

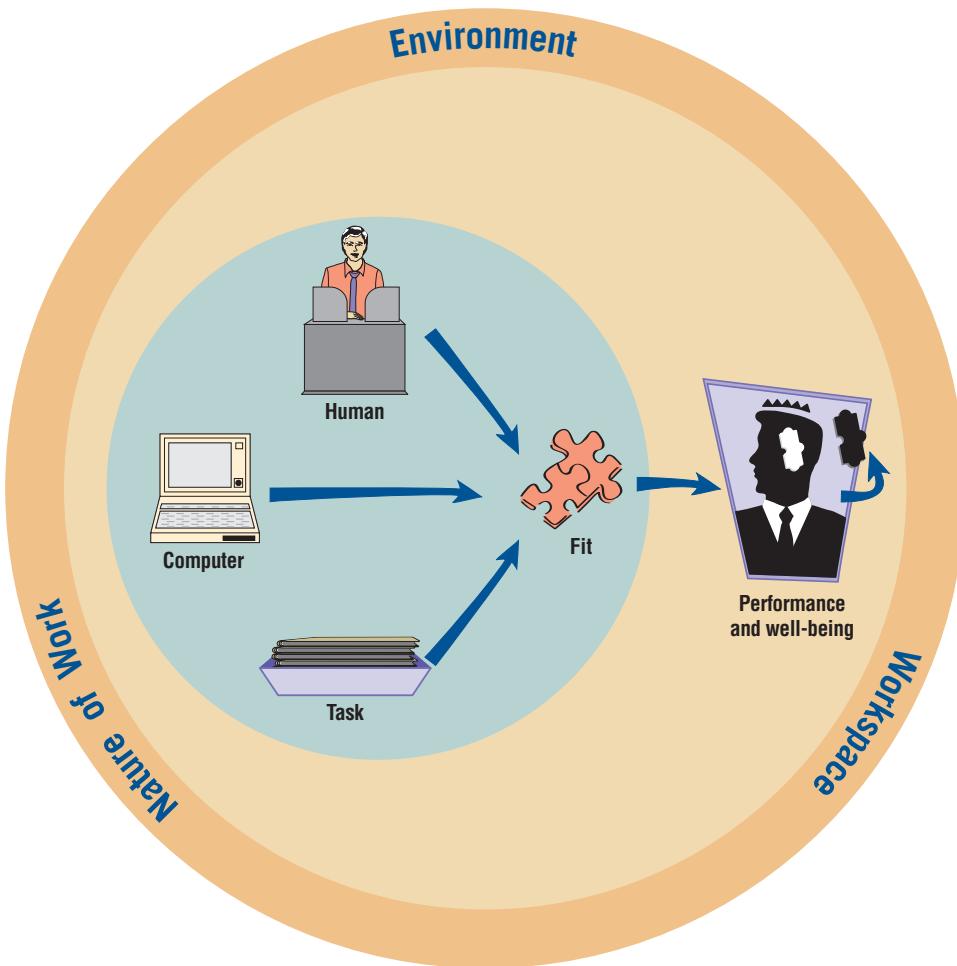


FIGURE 14.1

The “fit” among the human, computer, and task affects performance and well-being.

performing a task and the quality of the work that is produced by the task. For example, if analysts are using a high-level software or CASE tool to create data flow diagrams in which they are proficient, we would predict that the quality of the data flow diagrams produced would be high. The performance is also efficient, because the analysts are using an automated tool with which they are familiar. They can work rapidly, with good results. The task fits the objective, which is to create high-quality data flow diagrams to document a system. The efficiency of producing such diagrams with a CASE tool, which can then be used to store, retrieve, communicate, and modify the DFD diagrams, is excellent, compared to alternatives such as using a drawing tool unrelated to a data dictionary or drafting diagrams by hand, neither of which offer such features.

Well-Being At this point, we can introduce the concept of well-being, which is a concern for a human’s overall comfort, safety, and health; in sum it is their physical as well as psychological state. Does using a CASE tool for producing DFDs on a computer serve the analyst’s well-being? Yes, because the task fits well with the analyst, the software, the objective, and the computer. Notice that the analyst is working in an environment where they are physically comfortable, are psychologically stimulated to be creative, and can be productive; also, the analyst’s work is valued by peers and clients, as well as valued monetarily by the employing organization.

Psychological attitudes (the affective component) are also important. How users feel about themselves, their identities, their work life, and performance can

all be gauged through assessing their attitudes. As an analyst taking an HCI perspective, you are concerned about how humans' attitudes color the way they feel about technology and their tasks, and whether their attitudes hinder or enhance their experience.

THE TECHNOLOGY ACCEPTANCE MODEL AND ATTITUDE

The technology acceptance model (TAM), as proposed by Davis in 1989 and later refined and improved by Davis and others, basically is a way for analysts to organize their thinking about whether users will accept and use information technology. It can be used to shape training after a system has been developed, but it can also be used early in the development process to garner user reactions to prototypes so that systems can be changed early on in the development process to increase the likelihood of their adoption and use.

There are many theoretical components and a good deal of research to argue the intricacies of TAM. However, from a practical point of view, you need to be aware that a large body of research on the acceptance and use of technology exists in the information systems literature and that TAM is one of the most popular subjects. TAM draws its power from examining the perceived usefulness of the system to increase one's job performance and the belief about how easy the system will be to use when a user sits down to accomplish a task. So we have the two keys: **perceived usefulness** and **perceived ease of use**. Both can be used to understand how users intend to interact with a proposed system. Some researchers add an explicit attitude dimension to their conceptualization of the technology acceptance model that can help them think more specifically about what psychological states will shape the way users accept or reject the use of the information systems they design.

Attitudes toward computers include user satisfaction with the human-computer interface, as well as users' overall satisfaction with the system. These are generally ascertained through special user satisfaction surveys and are often used following implementation to estimate the overall success or failure of a systems project. (See Chapter 17 for some examples.) When you attempt to characterize attitudes toward computers, you may be surprised by all of the possible human responses that are conveyed. Most of the literature that we are examining for HCI will look at a variety of user attitudes, including satisfaction, anxiety, enjoyment, and playfulness in approaching technology.

The technology acceptance model also points out the importance of whether users find a system useful and are thus motivated to use it. Since this is an important HCI concern, we can measure whether the information technology is found to be useful by examining whether the system provides support for an organizational member's individual tasks. We can also measure whether there are important tasks that a user of the new system could not perform prior to its implementation. Our measurements can also determine whether the system extends a user's capabilities (for example, increasing the ability to perform higher-level analysis quickly or performing an on-the-spot translation of a financial report into another language complete with currency conversions). Part of the usefulness criterion in HCI can also be measured by ascertaining whether users find it rewarding to use the system through postimplementation interviews and observations.

USABILITY

Usability is a term that is defined differently depending on the branch of science in which one first encounters it. For our purposes in exploring usability through an

HCI lens, we will try to focus on usability as a way for designers to evaluate the systems and interfaces they create with an eye toward addressing as many HCI concerns as thoroughly as possible. Usability studies (according to www.useit.com) are all about finding out what works in the world and what doesn't. The ISO has created usability standards that you can explore on http://www.usabilitynet.org/tools/r_international.htm. The standards cover the use of the product (effectiveness, efficiency, and satisfaction in a particular context of use), the user interface and interaction, the process used to develop the product, and the capability of an organization to apply user-centered design.

Nielsen and Mack (1994) and Nielsen, Molich, Snyder, and Farrell (2001) have published usability heuristics (or rules of thumb) based on their thousands of usability tests of interfaces and, later, tests of ecommerce Web sites. They include visibility of system status, match between the system and the real world, user control and freedom, consistency and standards, error prevention, reconnection rather than recall, flexibility and efficiency of use, aesthetic and minimalist design, help that users recognize, diagnosis and recovery from errors, and help and documentation. Some of these are already familiar to you from the input and output design chapters.

Figure 14.2 is a usability survey to give directly to users who have personally interacted with a prototype. It asks users outright about some key usability and ergonomic questions. Another approach is to write up use case scenarios for the system. These are helpful in examining usability concerns.

DESIGNING FOR THE COGNITIVE STYLES OF INDIVIDUAL USERS

One important consideration is that data, particularly data used for decision making, are made available in different forms so that users with different cognitive abilities can make sense of them. Some users may prefer to examine tables and make decisions, some prefer graphs, and others want to read text.

It is even possible for the same person to want different types of presentations at different times. For example, suppose a manager wants to compare inventory held at different stores in a region. A graph can present the data very effectively. A column chart can use colors to show when a store is near its stockout level, and it can also show the relative amount of stock by allowing the user to visually compare the height of the bars directly.

Suppose now the same decision maker wants information about a particular store in a given month. The graphical depiction may have been set up to show the stores from highest to lowest inventory on a month-by-month basis. The user may prefer to return to the table that lists stores alphabetically, with the months listed chronologically. As you can see, the same person may want to see the same data in two very different ways.

Pivot Tables Pivot tables allow users to arrange data in a table in any way they choose. An example of a pivot table template created in Microsoft Excel is shown in Figure 14.3. The user would take an item from the pop-up box called "Pivot Table Field List," such as **Product**, drag it over to the table template, and drop it in one of the blank areas. In this example, the user drags and drops **Product** into the area on the left entitled "Drop Row Fields Here." The user drops **Sales** into the largest area that says "Drop Data Items Here."

Finally the user takes the item called **Quarter** and drops it into the area called "Drop Column Fields Here." The result is a table that shows each of the products in alphabetical order and its sales for each of the four quarters we have data for, followed by the grand total for the year. This table is shown in Figure 14.4.

Usability Survey

Please fill this out after you complete your interaction with the prototype. Circle a number as you respond to each question.
 Please hand your survey to the analyst when you have completed it. Thank you for this important feedback.

Prototype being evaluated _____ Version _____ Date _____/_____/_____

Human–Computer Interaction Factors		Very Poor	Average	Very Good			
Physical/Safety Concerns		1. How well were you able to read the display or form?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
Usability Concerns How well did the system:		2. If audio was used, were you able to hear it?	1 1 1	3 3 3	4 4 4	5 5 5	
		3. Did you consider the system safe to use?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
Pleasing and Enjoyable Attributes		4. Help you cut down on making errors?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		5. Allow you to recover from an error if you made one?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		6. Help you use it easily?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		7. Help you remember how to use it?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		8. Make it easy to learn how to use it?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
Usefulness Attributes How well did the system:		9. Was the system attractive?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		10. Was the system engaging (you wanted to use it)?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		11. Do you trust it as a system?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		12. Was it satisfying to use?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		13. Was it enjoyable to use?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		14. Was the system entertaining?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		15. Was the system fun to use?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
Usefulness Attributes How well did the system:		16. Support your individual task or tasks?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		17. Help you to extend your capabilities?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		18. Make itself rewarding to use?	1 1 1	2 2 2	3 3 3	4 4 4	5 5 5
		19. Permit you to do tasks that the other system would not allow you to do?	1 1 1 1	2 2 2 2	3 3 3 3	4 4 4 4	5 5 5 5

FIGURE 14.2

A form may be used to survey users of prototypes on key usability and ergonomic factors. (Categories based on Zhang, Carey, Te’eni, and Tremaine, 2005, table of HCI concerns, p. 522.)

Of course, the user could have done the opposite, that is, drag the item **Quarter** to the leftmost column and the **Product** to the area that says “Drop Column Fields Here.” That operation, however, would have produced a table with many columns (one for each product) and only five rows (one for each quarter plus a row for the total). The resulting table would have been difficult to read.

Many different tables can be displayed by just rearranging these four variables. If the user dragged the variable **Category** over to the area that says “Drop Column Fields Here,” the columns would have been categories of products, rather than the quarters, and the resulting table would have clearly shown which of the items belonged in each category and produced subtotals for each category. If **Category** was

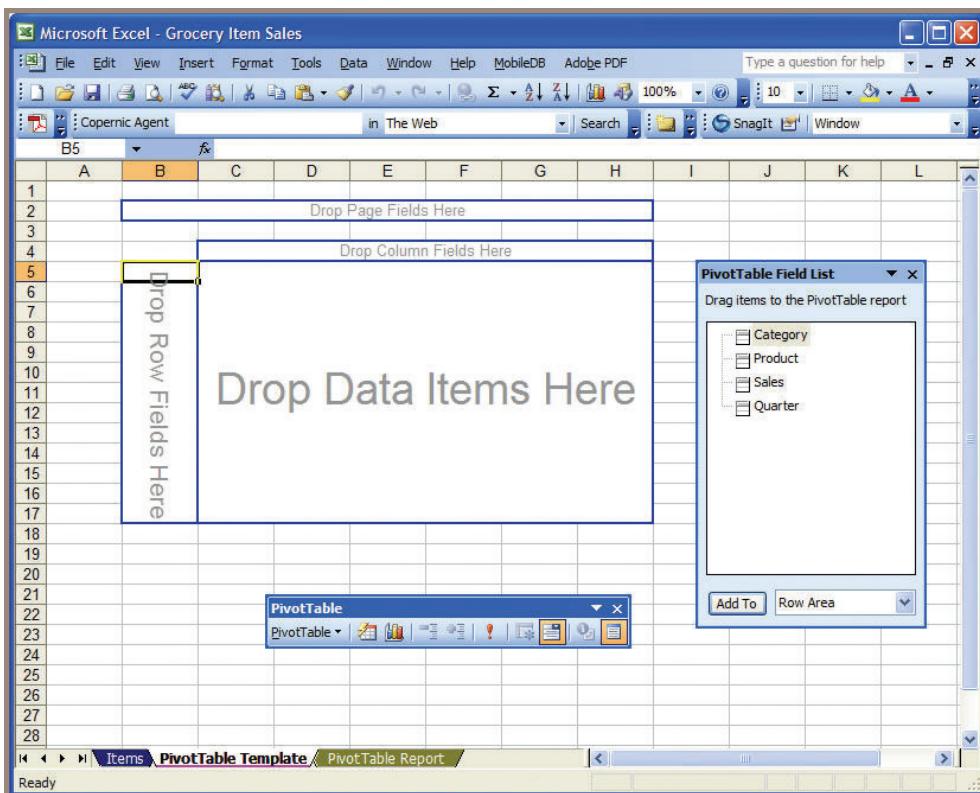


FIGURE 14.3

A pivot table template can make it easier for users to see information displayed in different ways.

dragged to the area at the very top of the template that says “Drop Page Fields Here,” then each category would have a table of its own beginning on a separate page.

The idea of a pivot table is useful because it gives users greater control over how they look at data in different ways within a table. We can examine this same concept for graphs in the next section.

Sum of Sales						
Product	Quarter	Qtr 1	Qtr 2	Qtr 3	Qtr 4	Grand Total
Alice Mutton		2667.6	4013.1	3900	6000.15	16580.85
Aniseed Syrup		544	600	140	440	1724
Boston Crab Meat		1474.41	2272	3887.92	2162	9796.33
Camembert Pierrot		3329.28	3989.9	10273.1	3060	20652.28
Carnarvon Tigers		1500	2362.5	7100	4987.5	15950
Chai		705.6	878.4	1174.5	2128.5	4887
Chang		2720.8	228	2061.5	2028.25	7038.55
Chartreuse verte		590.4	360	1100.7	2424.6	4475.7
Chef Anton's Cajun Seasoning		225.28	2970	1337.6	682	5214.88
Chef Anton's Gumbo Mix				288.22	85.4	373.62
Chocolade		744.6	162.56	68.85	306	1282.01
Côte de Blaye		25127.36	12806.1	7312.12	1317.5	46563.08
Escargots de Bourgogne			265	1393.9	417.38	2076.28
Filo Mix		187.6	742	226.8	911.75	2068.15
Flétanmysost		4454.8	174.15	2541.29	2472.5	9642.74
Geitost		294	242.5	99.5	150	786
Genen Shouyu			176.7	1298.12		1474.82
Gnocchi di nonna Alice		6931.2	9868.6	6771.6	9032.6	32604
Gorgonzola Telino		487	2993.12	1458.75	2681.87	7620.74
Grandma's Boysenberry Spread				1750	750	2500
Gravad lax		208	421.2			629.2
Guaraná Fantástica		529.2	467.55	219.37	337.5	1553.62
Gudbrandsdalsost		2649.6	1267.2	4473	5652	14041.8

FIGURE 14.4

After the user drags the items Product, Quarter, and Sales to the template, the table looks like this.

Visual Analysis of Databases You might be surprised to learn that innovative visual displays of data existed for quite some time, as early as the eighteenth century. Barriers to widely using visual displays included lack of imagination, the inability to draw graphs and charts in a cost-effective manner, and a lack of appreciation for such displays. After all, the audience for the information must understand the information in the diagram or it adds little value.

Software that enables the user to visually examine a database or spreadsheet is available. One example is Tableau Software's product (www.tableausoftware.com). Using an approach similar to the pivot tables we saw in Microsoft Excel, Tableau allows the user to drag and drop variables onto either a row or a column, and they appear on a graph. In Figure 14.5, the **Region** and **Weekday** were designated as columns and the **SUM (Sales Total)** was designated as a row. Each **Product Category** was then graphed (with "furniture" in blue, "office supplies" in orange, and "technology" in green).

The graph demonstrates that technology sales were higher than the other categories, but in particular technology sales were much higher than either furniture or office supplies in the East. The user was easily able to see this because the **Region** was singled out as one of the separators by dragging it to the area as a column.

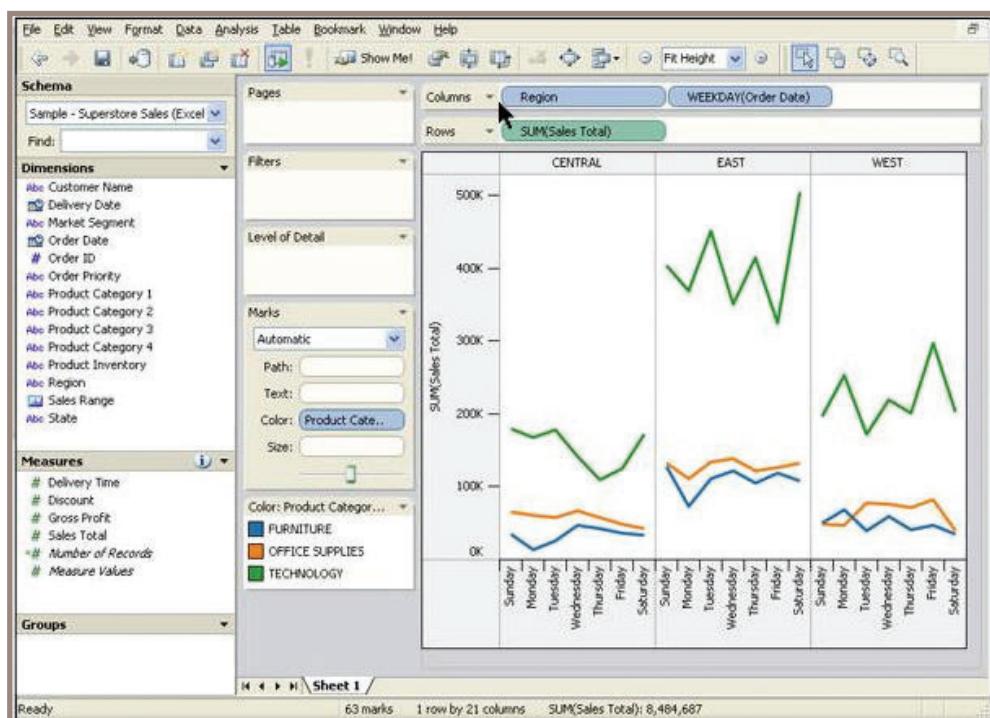
Tableau is a well designed software package because it goes much further in extending user capabilities to perform their tasks through the use of pivot table techniques. The developers also realized that users may want to group the data into what they consider a meaningful group. Users may then continue analysis by examining one of the groups further.

Figure 14.6 examines the **SUM (Gross Profit)** from each **Product Category** from our example. This graph uses color to indicate a profit (green) or a loss (red). In fact, the intensity of the color indicates the amount of profit or loss.

This graph can be used to explore the situation more deeply by selecting the three clusters of circles that are bright red, isolating them, and then looking at the data for those observations in more detail. Users can examine graphs or simply

FIGURE 14.5

This table, showing the daily sales by category and by region, was produced using Tableau. (Courtesy of www.tableausoftware.com.)



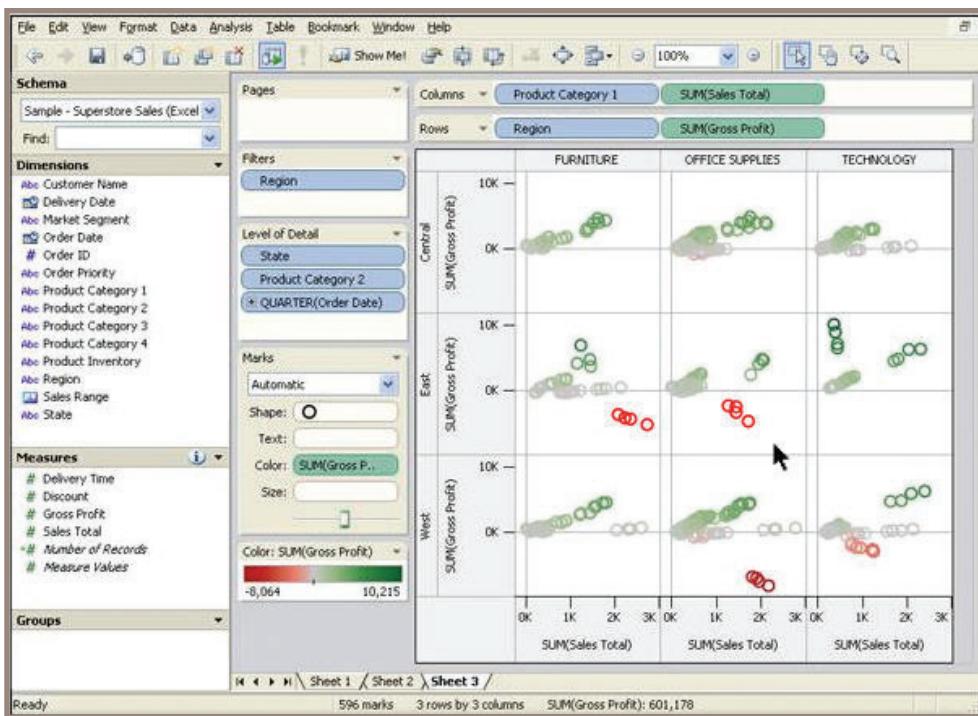


FIGURE 14.6

Products yielding losses are highlighted in bright red on this scatter plot, created using Tableau. (Courtesy of www.tableausoftware.com.)

look at the observations in a table. Once again, they have control over how the information is presented and thus control their task for best cognitive fit.

Another example from Tableau, presented in Figure 14.7 shows that this software can also create a dashboard (explained in Chapter 11). Here a table, a scatter plot, and a column chart are all shown on the same page. Visual analysis tools like this support visual thinking and extend the user's cognitive capabilities to do so. An appropriate visual display will increase the chances of making an appropriate decision.

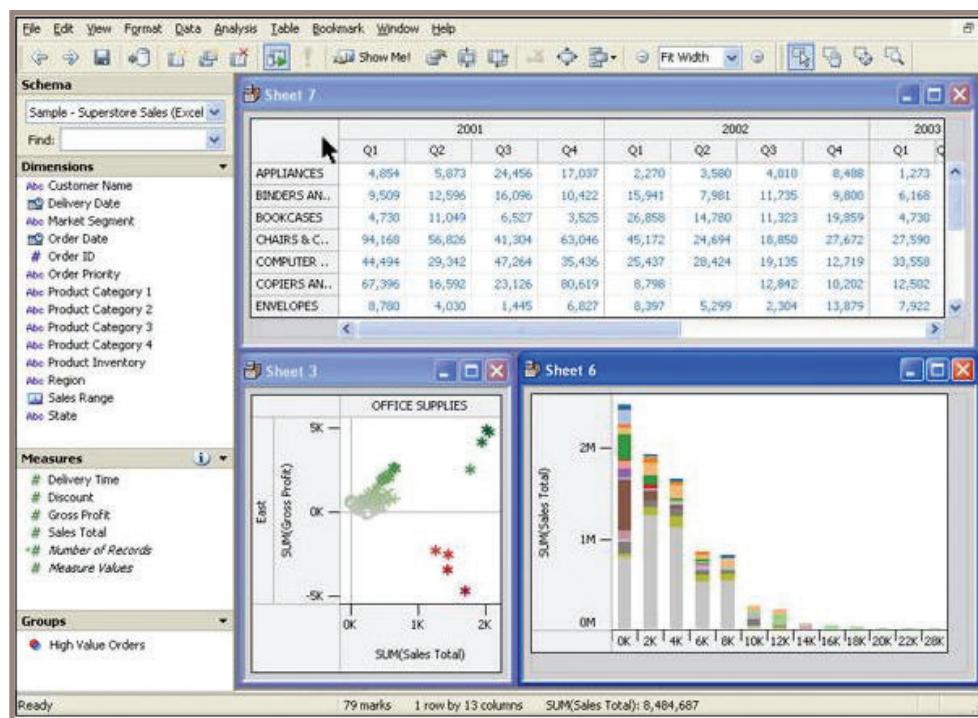


FIGURE 14.7

When different graphs or tables can be displayed on the same page, the page resembles a dashboard. (Courtesy of www.tableausoftware.com.)

PHYSICAL CONSIDERATIONS IN HCI DESIGN

In Chapters 11, 12, and 13 you learned the basis for sound design of screens, forms, Web sites, and databases. This included the special use of fonts, color, and layout design to communicate to users and to help them do the right thing with the input and output they encountered. To examine the underlying reasons for much of the design you learned, it is useful to look at human sensory capabilities and limitations that will inform our design. In keeping with the HCI philosophy, an analyst should be able to compensate, overcome, or replace human senses to a varying extent.

Vision As you learn to become a systems analyst, you are becoming accustomed to designing screens and reports for sighted people. The use of color, fonts, graphics, software, and PowerPoint presentations for displays and printed reports as input and output were detailed in Chapters 11 and 12. However, from an HCI perspective, you will also want to think in terms of limitations on human vision. Factors such as length of the distance from display to the person performing a task; the angle of the display in relation to the person viewing it; the size and uniformity of the characters; the brightness, contrast, balance, and glare of the screen; and whether a display is blinking or stable can all be designed to standards established through ISO and other national and international groups.

Hearing Humans also have limits to the amount of stress their senses can withstand. Noisy laser printers and phone conversations can lead to overload on human hearing. Office workers can wear noise-canceling headphones or get a personal music player like an iPod, but these solutions may have the effect of isolating a person from the organizational setting and may even diminish their capability to perform the task at hand. As an analyst you will need to consider noise when you design office systems.

Touch When using an HCI perspective to evaluate the usefulness of keyboards and other input devices, we can rate the human-computer fit as well as the dimensions examining the human-computer-task fit. Later in this chapter we will discuss the choices of human-computer interfaces, such as keyboards, direct manipulation, using a stylus, a mouse, and touch screens.

Keyboards have been ergonomically designed to provide the correct feedback for the person doing data entry. Users know by the firmness of the key under their finger that the keystroke has been entered. Although keyboards can be silenced, they are often designed with a click of feedback that is emitted when a key is hit. Keyboards also include slightly raised bumps on what are called home keys, often the *f* and the *j* keys, which orients users to where their fingers are positioned on the keyboard, enabling them to look at the screen or type from a printed page on their desk without continually glancing at the keyboard.

Although the popular QWERTY keyboard that we most often use with computers today was originally designed to slow down typists so that mechanical keys of the day would not become entangled, this layout has proved to be quite an efficient way to enter data. In fact, since users do so well with this familiar interface, it is difficult to conduct experiments comparing the efficiency of QWERTY keyboards with other innovative keyboards.

Designing for data entry using numeric keypads as the human entry device also provides a decision point for designers. Notice that numbers on your mobile phone are ordered differently than numbers on a numeric keypad or calculator. Your phone may be arranged with the numbers 1, 2, 3 on the top row. When you

look at a calculator layout or a numeric keypad on your keyboard, you will see 7, 8, and 9 on the top row instead. Research now points to the superiority of the calculator layout when the user is doing a lot of data entry. However, the phone digit layout is supposed to be better for locating a number. As a designer, you are constantly examining the fit between the human, the computer, and the tasks set by the organization.

Microsoft is currently developing touch screens featuring familiar icons that will help illiterate workers find domestic jobs through a touch screen interface that does not require reading, yet still allows unemployed workers power over choosing new employers. While illiteracy can be remedied through education, some limitations cannot.

CONSIDERING HUMAN LIMITATIONS, DISABILITIES, AND DESIGN

All humans have limitations in their physical capabilities. Some are immediately visible, others are not. When designing from an HCI perspective, you start realizing that limitations are often discussed in terms of disabilities. The application of HCI to supporting and enhancing the physical capabilities of humans is one of the most promising application areas. Strides in biomedical engineering mean that there is research to support the blind or those with low vision, those who are deaf or have impaired hearing, and people with limited mobility.

There are also improvements in the technical supports available to those who face difficulties in cognitive processing, including persons suffering with symptoms of autism, dyslexia, and attention deficit disorder. As a systems analyst you will be working under the legal provisions of the country in which you are working. For instance, if you are designing for workplaces in the United States, you may want to access the obligations of an employer under the Americans with Disabilities Act at www.eeoc.gov/types/ada.html. There you will find definitions of who is considered disabled, which states in part, “An individual with a disability is a person who: has a physical or mental impairment that substantially limits one or more major life activities; has a record of such impairment; or is regarded as having such an impairment.”

An employer in the United States is expected to make reasonable accommodation to employ a disabled person, which includes “Making existing facilities used by employees readily accessible to and usable by persons with disabilities; job restructuring, modifying work schedules, reassignment to a vacant position; acquiring or modifying equipment or devices, adjusting or modifying examinations, training materials, or policies, and providing qualified readers or interpreters.”

A qualified employee or application is an individual who, “with or without reasonable accommodation, can perform the essential functions of the job in question.” An employer is required to make reasonable accommodation to the known disability of a qualified applicant or employee if it would not impose an undue hardship on the operation of the business. Undue hardship is defined as “an action requiring significant difficulty or expense when considered in light of factors such as an employer’s size, financial resources, and the nature and structure of its operation. An employer is not required to lower quality or production standards to make an accommodation.”

One of the best ways to ensure the broadest possible accommodation is to begin designing from an HCI perspective. That way, your foremost concern will always be assisting a user in accomplishing a task, set by the organization, with the use of technology. However, when accommodations for disabled people are necessary, there are many sources to examine and many assistive devices to consider.

SCHOOL SPIRIT COMES IN MANY SIZES

Matt Scott manages the student-alumni clothing department for a large bookstore in Saratoga Springs.

"Our clothing sales depend not only on whether our sports teams win or lose, but the overall well-being of our students and alums. If they are proud of their university, and want to show their school spirit, they'll buy up everything on our racks," exclaims Matt. "But don't underestimate the weather as a factor," he adds. "If the weather turns cold in October, you'll see a surge in people buying warm sweaters, pullovers, and gloves."

"Our store serves the major three universities in our area," Matt goes on to state. "First, there is Hyde Park, what we call 'the football school.' They have about 17,000 students going there. They have high demand for school-branded clothing, particularly in the fall. Then, of course, there's Pierce University. Pierce thinks it's part of the Ivy League, so the students like to buy crew and Lacrosse shirts. They have about 7,500 students. Then there is St. David's, with about 3,000 students. They are devoted to their basketball team. They really have faith in them. You'll see sales pick up in the second semester, particularly during 'March Madness.'"

Mr. Scott continues, admitting, "I thought about asking the students what to stock, but an email survey is out of the question. I get spammed a lot, so I mostly don't bother with email. Unfortunately, the lead time for getting official branded sportswear into the store is really long, and we run the risk of stocking out. But we try to never run out."

You've been asked to design a set of tables and graphs that will help analyze the sales of Matt Scott's school clothing. Start by listing

about 20 different items of school-branded clothing for men and women fans, including items such as hooded sweatshirts, T-shirts, baseball caps, sweatbands, running shorts, and so on. Many of them feature fanciful embroidered designs depicting their mascots in menacing or endearing poses. Hyde Park has their Golden Retrievers; Pierce has their much beloved birds, the Puffins; and St. David's has their Dragons.

Put the items into categories. Then think about what the data would look like. Does it make sense for Matt to look at the data weekly, monthly, or by semester? Will he want to look back five years to see if there were any trends? Set up tables identifying the rows and columns and the content of the main cells. Suggest several tables so that Matt can analyze them in different ways.

Now construct graphs that analyze the same data. Using some of the examples found in this book, suggest the type of graphs and show the data so that different users with different styles can make some decisions regarding the trend of sales over the last few years. Remember to compare the schools as well. Suggest the appropriate graphs from column, line, scatter plots, or even pie charts.

Also suggest three or four specific changes you would make to allow someone who has low vision to be able to read the graphs more easily. Magnification is one way to change a graph, but may not be the best approach.

Consider the size of the schools since this may become the most important factor when determining how Matt Scott should adjust his ordering for David's, Hyde, and Pierce.

For people who are blind or who have low vision, there are braille keyboards as well as special speech software that reads Web pages and other documents aloud. There are also screen magnifiers that fit over a display to magnify the entire screen.

For people who lack certain perceptual sensitivity (incorrectly called color blindness), you can work at testing the colors you are choosing to make certain that they can be easily distinguished from each other. Particular problems occur telling the difference between red and green for instance. Always design the screen or form with alternative cues, such as icons, written text, or audio cues that reinforce the content. For instance if a hyperlink that has been clicked on turns blue to show it has been followed, you can also add another icon to the display to indicate that it has been followed or create a separate sidebar list that shows which Web sites have been visited. These are better alternatives than relying solely on color to convey your message.

For users who experience impaired hearing, you can make sure that the documents and screens you design include access to written versions of the audio material. Alternatively, you might design tasks where headphones can be successfully used.

If you are designing computer tasks for those with limited mobility, you can think of speech input rather than keyboarding. Additionally, new advances in biomedical engineering permit mobility-impaired users to move the cursor on the screen by breathing into a tube or by directing the cursor to the desired spot on the

Guidelines for the HCI Approach to Systems Design

- Examine the task to be done and consider the fit among the human, computer, and task.
- Identify what obstacles exist for users in their attempts to accomplish their assigned tasks.
- Keep in mind the perceived usefulness and perceived ease of use from TAM.
- Consider usability. Examine the usage environment by creating use case scenarios that depict what is going on between users and the technology.
- Use the information you have gained beforehand to figure out the physical and organizational environmental characteristics. Design with prototyping to accommodate diverse users and users with disabilities.

FIGURE 14.8

The HCI approach to systems design emphasizes the fit among the human, computer, and task.

screen by looking at that spot or even, in some highly specialized interfaces, by thinking about where the cursor should move.

IMPLEMENTING GOOD HCI PRACTICES

The ideal is to invite a usability specialist to serve on the systems development team with the other team members. However, many systems groups are quite small, and not many professionals are available who are involved in the practice of usability per se; so even if you make this recommended change to your project, the odds are that the position will go unstaffed or understaffed. However, don't let that discourage you. You can take some simple steps that will positively influence the outcome of your systems project. Figure 14.8 provides a list of guidelines for taking an HCI approach to systems design. As systems designers, we can become aware of important HCI dimensions, knowledgeably linking them to systems designs, and we should attempt to measure how we are meeting each of these concerns for users and organizations.

Although we have been discussing the system in the abstract, it is important to recognize that the interface *is* the system for most users. However well or poorly designed, it stands as the representation of the system and, by reflection, your competence as a systems analyst. A well designed interface improves the fit among the task, the technology, and the user.

Your goal must be to design interfaces that help users and businesses get the information they need in and out of the system by addressing the following objectives:

1. Matching the user interface to the task.
2. Making the user interface efficient.
3. Providing appropriate feedback to users.
4. Generating usable queries.
5. Improving the productivity of computer users.

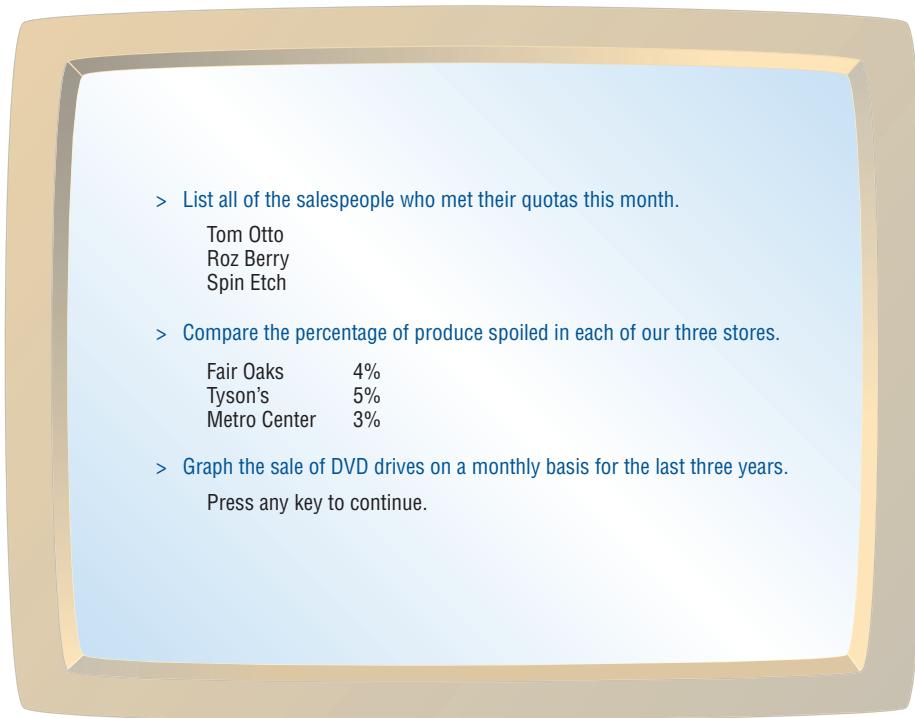
With these goals in mind, we move to more detailed discussions of how each of the objectives can be met.

TYPES OF USER INTERFACE

In this section, several different kinds of user interfaces are described, including natural-language interfaces, question-and-answer interfaces, menus, form-fill interfaces, command-language interfaces, graphical user interfaces (GUIs), and a variety of Web interfaces for use on the Internet. The user interface has two main components: presentation language, which is the computer-to-human part of the transaction, and action language, which characterizes the human-to-computer portion. Together, both concepts cover the form and content of the term *user interface*.

FIGURE 14.9

Natural-language interfaces.



NATURAL-LANGUAGE INTERFACES

Natural-language interfaces are perhaps the dream and ideal of inexperienced users, because they permit users to interact with the computer in their everyday, or natural, language. No special skills are required of the user, who interfaces with the computer using natural language.

The display depicted in Figure 14.9 lists three natural-language questions from three different applications. Notice that interaction with each seems very easy. For instance, the first sentence seems straightforward: "List all of the salespeople who met their quotas this month."

The subtleties and irregularities residing in the ambiguities of English produce an extremely exacting and complex programming problem. Attempts at natural-language interfacing for particular applications in which any other type of interface is infeasible (say, in the case of a user who is disabled) are meeting with some success; however, these interfaces are typically expensive. Implementation problems and extraordinary demand on computing resources have so far kept natural-language interfaces to a minimum. The demand exists, though, and many programmers and researchers are working diligently on such interfaces. It is a growth area, and it therefore merits continued monitoring. Some Web sites, such as Ask.com, use a natural interface for users to enter their search query. When the query is entered, Ask.com responds with a list of responses that match the question entered by the user.

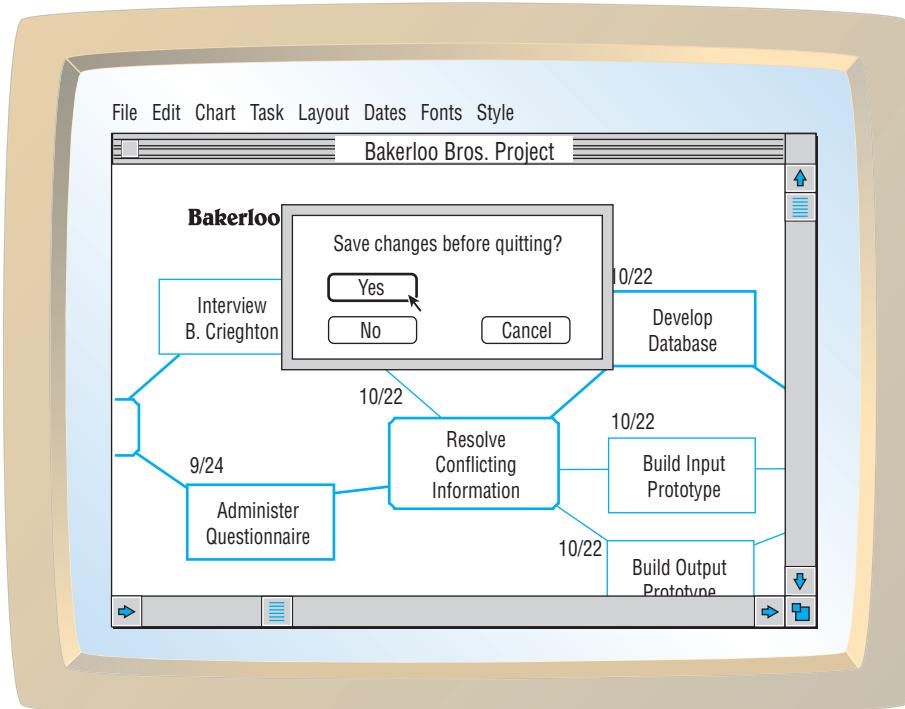
QUESTION-AND-ANSWER INTERFACES

In a question-and-answer interface, the computer displays a question to the user on the display. To interact, the user enters an answer (via a keyboard stroke or a mouse click), and the computer then acts on that input information in a preprogrammed manner, typically by moving to the next question.

A type of question-and-answer interface called a dialog box is shown in Figure 14.10. A dialog box acts as a question-and-answer interface within another

FIGURE 14.10

A dialog box: one type of question-and-answer interface.



application, in this case a PERT chart for a systems analysis project for the Bakerloo Brothers. Notice that the rounded rectangle for “Yes” is highlighted, indicating that it is the most likely answer for this situation. The main interface for this application need not necessarily be question and answer. Rather, by incorporating a dialog box, the programmer has included an easy-to-use interface within a more complicated one.

Wizards used to install software are a common example of a question-and-answer interface. The user responds to questions about the installation process, such as where to install the software or features. Another common example is the use of the Office Assistant with Microsoft products. When the user needs help, the Office Assistant asks questions and responds to the answers with additional questions designed to narrow the scope of the problem. Users unfamiliar with particular applications or not knowledgeable about a topic may find question-and-answer interfaces the most comfortable, quickly gaining confidence through their success.

MENUS

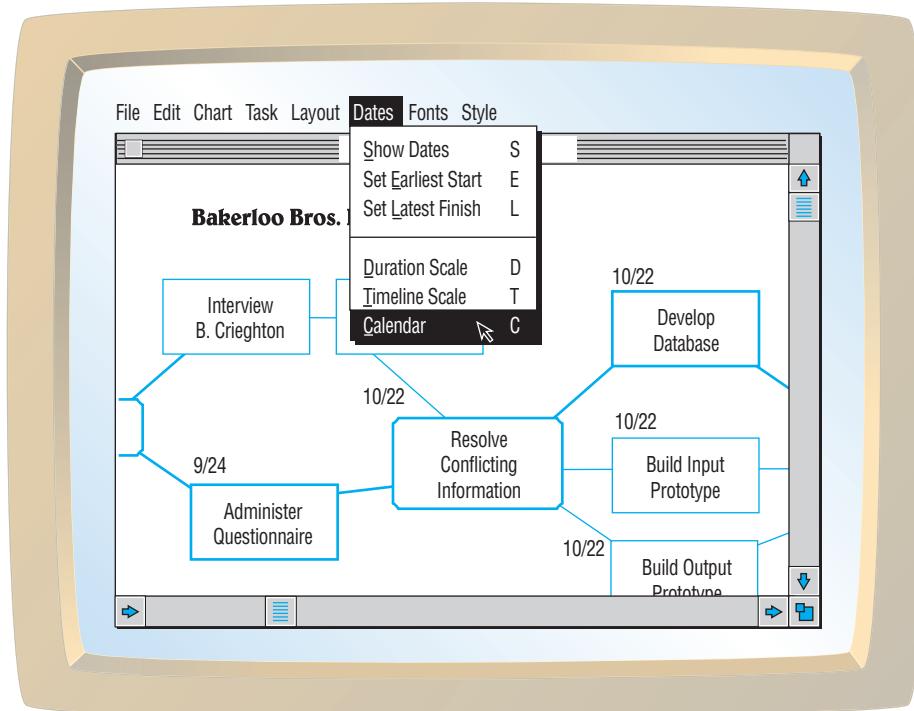
A menu interface appropriately borrows its name from the list of dishes that can be selected in a restaurant. Similarly, a menu interface provides the user with an onscreen list of available selections.

In responding to the menu, a user is limited to the options displayed. The user need not know the system but does need to know what task should be accomplished. For example, with a typical word processing menu, users can choose from the Edit, Copy, or Print options. To utilize the menu best, however, users must know which task they desire to perform.

Menus are not hardware dependent. Variations abound. Menus can be set up to use keyboard entry, light pen, or mouse. Selections can be identified with a number, letter, or keyword, or users can click on a selection with a mouse. Consistency is important in designing a menu interface.

FIGURE 14.11

A pull-down menu is there when the user needs it.



Menus can also be put aside until the user wants to employ them. Figure 14.11 shows how a pull-down menu is used while constructing a PERT diagram for a systems analysis project being completed for the Bakerloo Brothers. The user puts the pointer on **Dates** and pulls it down. Then the user puts the pointer on **Calendar**, selecting the option to display the project on a conventional monthly calendar.

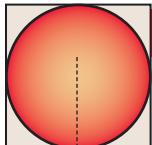
Menus can be nested within one another to lead a user through options in a program. Nested menus allow the screen to appear less cluttered, which is consistent with good design. They also allow users to avoid seeing menu options in which they have no interest. Nested menus can also move users quickly through the program.

GUI menus are used to control PC software and have the following guidelines:

1. The main menu bar is always displayed.
2. The main menu uses single words for menu items. Main menu options always display secondary pull-down menus.
3. The main menu should have secondary options grouped into similar sets of features.
4. The drop-down menus that display when a main menu item is clicked often consist of more than one word.
5. These secondary options perform actions or display additional menu items.
6. Menu items in grey are unavailable for the current activity.

An object menu, also called a pop-up menu, is displayed when the user clicks on a GUI object with the right mouse button. These menus contain items specific for the current activity, and most are duplicate functions of main menu items.

Experienced users may be irritated by nested menus. They may prefer to use a single-line command entry to speed things up. Other users might use the shortcut abbreviations or key combinations such as Alt > I > P > C, which inserts a picture that is clip art in a Microsoft Office document.



CONSULTING OPPORTUNITY 14.2

I'D RATHER DO IT MYSELF

"I can get Mickey to download any data I need from the Web or our server to my PC," DeWitt Miwaye, an upper-level manager for Yumtime Foods (a Midwest food wholesaler) tells you. "Getting data is no problem. What I don't want are a lot of reports. I'd rather play with the data myself."

Miwaye goes on to tell you that as an executive, he doesn't use his PC as often as he'd like, maybe only three times a month, but he has some very specific ideas about what he'd like to do with it.

"I'd like to be able to make some comparisons myself. I could compare the turnover rate for all 12 of our warehouses. I'd also like to

see how effectively the capacity of each of our warehouses is being used. Sometimes I'd like to be able to graph the comparisons or see a chart of them over time."

In three paragraphs, compare three different types of interfaces that Miwaye could use. Then recommend one interface for his use that takes into account his infrequent use of the PC, his enjoyment of working with raw data, and his desire to see data displayed in a variety of ways.

FORM-FILL INTERFACES (INPUT/OUTPUT FORMS)

Form-fill interfaces consist of onscreen forms or Web-based forms displaying fields containing data items or parameters that need to be communicated to the user. The form often is a facsimile of a paper form already familiar to the user. This interface technique is also known as a form-based method and input/output forms.

Figure 14.12 shows a form-fill interface. A pull-down menu for **Part No.** automatically enters a **Description** and **Unit Price** for the item. When the user tabs to the **Quantity** field and enters the number of items being purchased, the software automatically calculates the **Extended Price** by multiplying **Quantity** by **Unit Price**.

Forms for display screens are set up to show what information should be input and where. Blank fields requiring information can be highlighted with inverse or flashing characters. The cursor is moved by the user from field to field by a single stroke of an arrow key, for instance. This arrangement allows movement one field backward or one field forward by clicking the appropriate arrow key. It provides the user good control over data entry. Web-based forms afford the opportunity to

The screenshot shows a 'Purchase Order' form with the following details:

Order Date:	05/14/2006 (MM/DD/YYYY)	Required by:	06/12/2006 (MM/DD/YYYY)	Requisition No.:	MTC30023
Vendor Name and Address		Ship To			
Hamington Office Supplies 100 Nathan Lane Rochester, NY 14604		Jonathan Harris 2001 Biltmore Blvd. Samsel, NY 14225			
Part No.	Description	Quantity	Unit Price	Extended Price	
OS23561	Note pads, 4 in. x 6 in., box of 25	10	9.95	99.50	
OS93851	Clear tape, 12mmx33mm, box of 100	3	19.99	59.97	
OS83955	Hi-Liter, assorted colors, box of 12	2	8.56	17.12	
				0.00	
				0.00	
				0.00	
				0.00	

FIGURE 14.12

An example of the form-fill interface from Form Flow by JetForm.

include hyperlinks to examples of correctly filled-out forms or to further help and examples.

Form input for displays can be simplified by supplying default values for fields and then allowing users to modify default information if necessary. For example, a database management system designed to show a form for inputting checks may supply the next sequential check number as a default when a new check form is exhibited. If checks are missing, the user changes the check number to reflect the actual check being input.

Input for display screen fields can be alphanumerically restricted so that, for example, users can enter only numbers in a field requesting a Social Security number, or they can input only letters where a person's name is required. If numbers are input where only letters are allowed, the computer may alert the user via audio output that the field was filled out incorrectly.

The chief advantage of the input/output form interface is that the printed version of the filled-in form provides excellent documentation. It shows field labels as well as the context for entries. In addition, Web forms can return incomplete forms to the user with an explanation of what data must be entered to complete the transaction. Often, fields with missing data are marked with a red asterisk. Web-based documents can be sent directly to billing if a transaction is involved, or they can go directly to a consumer database if a survey is being submitted. Web-based forms push the responsibility for accuracy to the user and make the form available for completion and submission on a 24-hour, 7-day-a-week, worldwide basis.

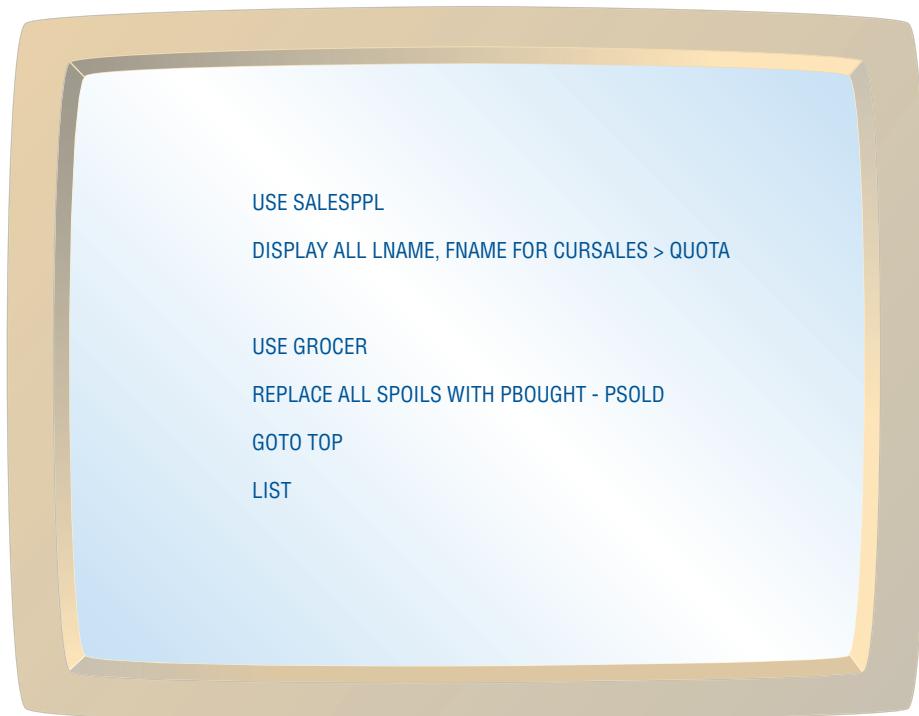
There are few disadvantages to input/output forms. The main drawback is that users experienced with the system or application may become impatient with input/output forms and may want more efficient ways to enter data.

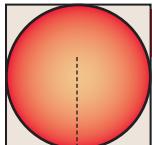
COMMAND-LANGUAGE INTERFACES

A command-language interface allows the user to control the application with a series of keystrokes, commands, phrases, or some sequence of these three methods. It is a popular interface that is more refined than those previously discussed.

FIGURE 14.13

Command-language interfaces.





DON'T SLOW ME DOWN

"I've seen 'em all," Carrie Moore tells you. "I was here when they got their first computer. I guess I've sort of made a career of this," she says cheerfully, pointing to the large stack of medical insurance claim forms she has been entering into the computer system. As a systems analyst, you are interviewing Carrie, a data entry operator for AbundaCare (a large medical insurance company), about changes being contemplated in the computer system.

"I'm really fast compared with the others," she states as she nods toward the six other operators in the room. "I know, because we have little contests all the time to see who's the fastest, with the fewest errors. See that chart on the wall? That shows how much we enter and how quickly. The gold stars show who's the best each week. Performance measures are my friends."

"I don't really mind if you change computers. Like I say, 'I've seen 'em all.'" She resumes typing on her keyboard as she continues the

interview. "Whatever you do, though, don't slow me down. One of the things I'm most proud of is that I can still beat the other operators. They're good too, though," Carrie adds.

Based on this partial interview with Carrie Moore, what type of user interface will you design for her and the other operators? Assume the new system will still require massive amounts of data entry from a variety of medical insurance forms sent in by claimants.

Compare and contrast interfaces such as natural language, question and answer, menus, input/output forms, and Web-based form-fill documents. Then choose and defend one alternative. What qualities possessed by Carrie and the other operators—and the data they will be entering—shaped your choice? Make a list of them. Is there more than one feasible choice? Why or why not? Respond in a paragraph.

Two application examples of command language are shown in Figure 14.13. The first shows a user who asks to use a file containing data on all salespeople, then asks the computer to display all last names and first names for all salespeople whose current sales (CURSALES) are greater than their quotas. In the second example, a user asks to use a file called GROCER, then directs the computer to calculate the spoilage (SPOILS) by subtracting produce sold from produce bought. After that is done, the user asks to go back to the top of the file and to print out (LIST) the file.

The command language has no inherent meaning for the user, and that fact makes it dissimilar to the other interfaces discussed so far. Command languages manipulate the computer as a tool by allowing the user to control the dialog. Command language affords the user more flexibility and control. When the user employs command language, the command is executed by the system immediately. Then the user may proceed to give it another command.

Command languages require memorization of syntax rules that may prove to be obstacles for inexperienced users. Experienced users tend to prefer command languages, possibly because of their faster completion time.

GRAPHICAL USER INTERFACES

The key to graphical user interfaces (GUIs) is the constant feedback on task accomplishment that they provide to users. Continuous feedback on the manipulated object means that changes or reversals in operations can be made quickly, without incurring error messages. The concept of feedback for users is discussed thoroughly in a later section.

The creation of GUIs poses a challenge, because an appropriate model of reality or an acceptable conceptual model of the representation must be invented. Designing GUIs for use on intranets, extranets, and, more pressingly, on the Web requires even more careful planning (see Chapter 12 on Web site design). By and large, the users of Web sites are unknown to the developer, so a design must be clear-cut. The choice of icons, language, and hyperlinks becomes an entire set of

THAT'S NOT A LIGHTBULB

From your preliminary analysis, it appears that a substantial reduction in errors will be realized if sales clerks at Bright's Electric (which sells electrical parts, bulbs, and fixtures to wholesale customers) adopt an online system. The new system would allow sales clerks to withdraw a part from inventory (and thereby update inventory), return a part to inventory, check on the inventory status, and check on whether a part is backordered. Currently, to update inventory, sales clerks fill out a three-part form by hand. The customer gets one, inventory keeps one, and at the end of the day the originals are deposited in the front office.

The next morning, the first thing the lone office worker does is enter the data from the forms into the computer. Errors occur when she enters the wrong part numbers or quantities. Additional time is consumed when inventory workers hunt for a part they think might be in stock but is not. Updated inventory sheets are available to the sales clerks around noon, but by that time they have already taken from inventory twice the number of parts that will be taken out after noon. Clearly, a well designed online system would help reduce these errors and also help with inventory control.

The owner, Mr. Bright, has entertained the idea of an online system and dropped it several times over the last five years. The chief reason is that the sales clerks, who would be the heaviest users of the system, do not think the systems analysts they've talked to can fulfill their needs.

M. T. Sockette, the sales clerk who has been with Bright's the longest, is the most vocal, telling you, "We know the parts, we know our customers. What we could do with a computer here would be great. The guys they've brought in here to get it going, though... I mean, they say things like, 'You can step right up and type one 60-watt General Electric lightbulb into the computer.'

"To us, that's not a lightbulb, it's a GE60WSB. All of us know the part numbers here. We pride ourselves on it. Typing in all that junk will take all day."

After talking to Mr. Bright, you decide to implement an online system. You have talked to M. T. and the others and reassured them that the system will use the part numbers they're familiar with and will save them time. Although they're skeptical, you've persuaded them to give it a try.

What type of user interface will you design for the sales clerks? Before you come to your solution, do a careful analysis in three paragraphs that compares and contrasts various user interfaces—natural language, question and answer, menus, input/output forms, command language, and Web-based form-fill documents—for their suitability at Bright's. Then choose one interface and explain in a paragraph why you find this one the most appropriate based on what you know about Bright's sales clerks and their current system. Draw a prototype of a display that will be part of your solution. Describe in a paragraph how you will test its usability with the sales clerks.

decisions and assumptions about what kinds of users the Web site is hoping to attract. The designer must also adhere to conventions that users now expect to encounter on Web sites.

OTHER USER INTERFACES

Other less common user interfaces are growing in popularity. These interfaces include pointing devices such as the stylus, touch-sensitive screens, and speech recognition and synthesis. Each of these interfaces has its own special attributes that uniquely suit it to particular applications.

The stylus (a small pointed stick that resembles a pen) is becoming popular because of new handwriting recognition software for personal digital assistants (PDAs) and mobile phones. Palms, combination mobile phones and PDAs, and Pocket/PC devices have been a success because they have are useful and easy to use. Additionally, they are portable and sell for a comparatively low price. There has been an explosion of fun and useful applications written for these mobile devices, including popular programs for restaurant reviews such as Zagats and for popular games such as Sudoku. Data entry is also facilitated with a docking cradle so that you can synchronize data with your PC.

A tablet PC is a notebook computer with a stylus- or touch-sensitive display. It is much more powerful than a handheld computer but weighs considerably more. Both handhelds and tablet PCs can be equipped with built-in Wi-Fi or Bluetooth® communication.

Touch-sensitive displays allow a user to use a finger to activate the display. Touch-sensitive displays are useful in public information displays, such as maps of cities and their sights posted in hotel lobbies or car rental facilities. They can also be used to explain dioramas in museums and to locate camping facilities in state parks. Touch-sensitive displays require no special expertise from users, and the screen is self-contained, requiring no special input device that might be broken or stolen.

With voice recognition, the user speaks to the computer, and the system is able to recognize an individual's vocal signals, convert them, and store the input. Voice recognition inventory systems are already in operation, and automobiles now feature voice input systems that respond to a driver's voice commands to navigate or to change the radio station.

An advantage of voice recognition systems is that they can speed data entry enormously, and free the user's hands for other tasks (for example, driving). Speech input adds still another dimension to the PC. It is now possible to add equipment and software that allows a PC user to speak commands such as "open file" or "save file" to avoid using the keyboard or mouse. Users with limited mobility or impaired sight can benefit from voice recognition systems. In the example shown in Figure 14.14, the user corrects a word by pulling down a menu of alternative words that sound the same.

When evaluating the interfaces you have chosen, keep some standards in mind:

1. The necessary training period for users should be acceptably short.
2. Early in their training, users should be able to enter commands without thinking about them or without referring to a help menu or manual. Keeping interfaces consistent throughout applications can help in this regard.
3. The interface should be seamless so that errors are few and those that do occur are not occurring because of poor design.
4. The time that users and the system need to bounce back from errors should be short.
5. Infrequent users should be able to relearn the system quickly.

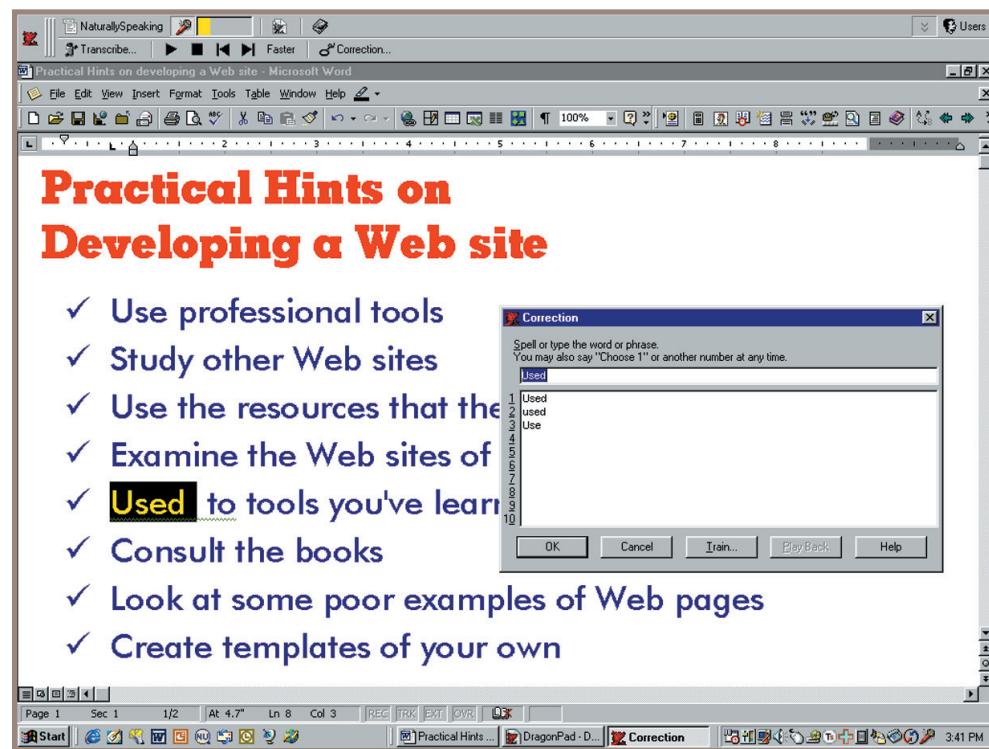


FIGURE 14.14

Using software such as Dragon NaturallySpeaking by Nuance, a user can speak commands to their computer. In this example, the user corrects a word by pulling up a menu of alternative words that sound the same.

Many different interfaces are available, and it is important to realize that an effective interface goes a long way toward addressing key HCI concerns. Users should want to use the system, and they should find it attractive and pleasing to use. In the next section, we discuss the importance of providing feedback for users to support and sustain their involvement with the system, so that they will be able to accomplish their tasks.

GUIDELINES FOR DIALOG DESIGN

Dialog is the communication between the computer and a person. Well designed dialog makes it easier for people to use a computer and lessens their frustration with the computer system. Recall the elements of the TAM (technology acceptance model) indicating that perceived usefulness and perceived ease of use will lead first to an intention to use the system and eventually to using it. There are several key points for designing good dialog. Some of them were mentioned in Chapter 12. They include the following:

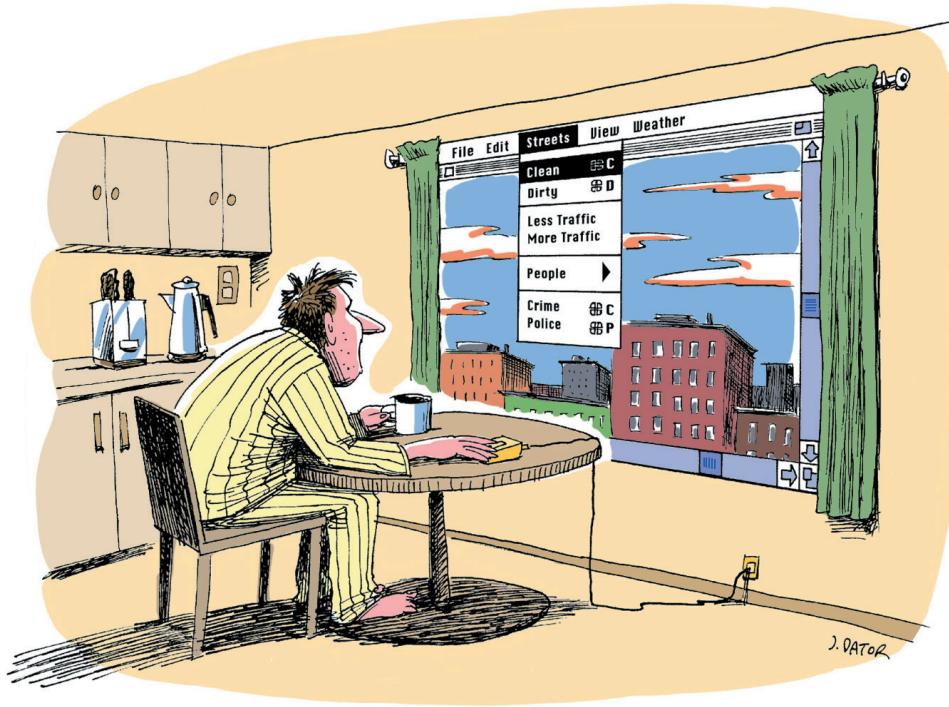
1. Meaningful communication, so that the computer understands what people are entering and people understand what the computer is presenting or requesting.
2. Minimal user action.
3. Standard operation and consistency.

MEANINGFUL COMMUNICATION

The system should present information clearly to the user. This means having an appropriate title for each display, minimizing the use of abbreviations, and providing clear user feedback. Inquiry programs should display code meanings as well as data in an edited format, such as displaying slashes between the month, day, and year in a date field or commas and decimal points in an amount field. User instructions should be supplied regarding details, such as available function key assignments. In a graphical interface, the cursor may change shape depending on the work being performed.

Users with less skill in using the computer or doing their tasks with a computer require more communication. Web sites must display more text and instructions to guide the user through the site. Intranet sites may have less dialog, because there is a measure of control over how well trained users are. Internet graphics should have pop-up text or roll-over descriptions when images are used as hyperlinks, because there may be uncertainty in interpreting their meaning, especially if the site is used internationally. Notice that EU guidelines for the display of Web graphics requires that all images be labeled, so that visually impaired users will be able to hear written descriptions announced through special software. Status line information for GUI screens is another way of providing instructions for users.

Easy-to-use help screens should be provided. Many PC help screens have additional topics that may be directly selected using highlighted text displayed on the first help screen. These hyperlinks are usually in a different color, which makes them stand out in contrast to the rest of the help text. Remember to use icons or text in addition to color coding in order to reach the largest number of users. Many of the newer GUIs often incorporate tool tip help, displaying a small help message identifying the function of a command button when the cursor is placed over it. The other side of communication is that the computer should understand what the user has entered. Hence, all data entered on the screen should be edited for validity.



Source: © 2003 Joe Dator from cartoonbank.com. All Rights Reserved.

MINIMAL USER ACTION

Keying is often the slowest part of a computer system, and good dialog will minimize the number of keystrokes required. You can accomplish this goal in a number of different ways:

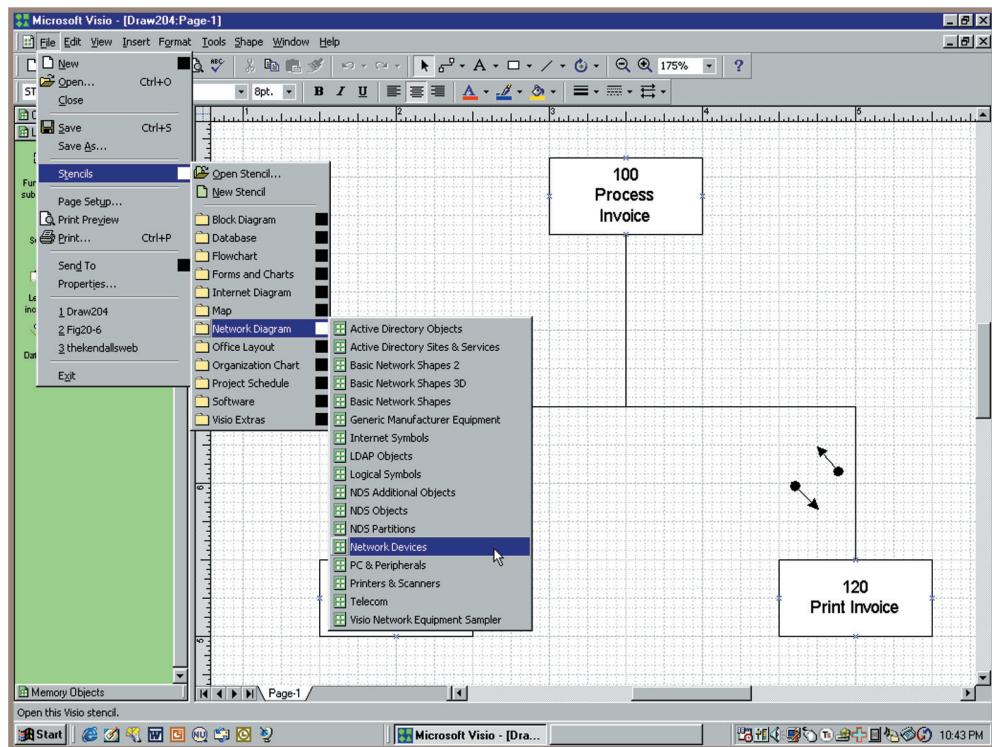
1. **Keying codes, such as airport codes when making a flight reservation, instead of whole words on entry screens.** Codes are also keyed when using a command-language interface, such as a two-letter state postal abbreviation. On a GUI screen, the codes may be entered by selecting descriptions of the codes from a pull-down list of available options. This helps to ensure accuracy, since the code is stored as a value of the drop-down list, as well as helping to provide meaningful communication since descriptions that are familiar to the user are selected. An example would be selecting a Canadian province and having the two-character postal code stored.
2. **Entering only data that are not already stored on files.** For example, when changing or deleting item records, only the item number should be entered. The computer responds by displaying descriptive information that is currently stored on the item file. Another example is when a user logs on to a Web site, the userID is used to find related records, such as a customer record, outstanding bills, orders, and so on.
3. **Supplying the editing characters (for example, slashes as date field separators).** Users should not have to enter formatting characters such as leading zeros, commas, or a decimal point when entering a dollar amount; nor should they have to enter slashes or hyphens when entering a date. In general, Web sites are an exception to this rule, since Web forms do not include slashes or decimal points. Some Web forms use a series of entry fields with editing characters between them, such as parentheses around an area code.
4. **Using default values for fields on entry screens.** Defaults are used when a user enters the same value in a screen field for the majority of the records being processed. The value is displayed, and the user may press the Enter key to

accept the default or overtype the default value with a new one. GUIs may contain check boxes and radio buttons that are selected when a Web form or dialog box opens. Context-sensitive menus appear when an object is clicked with the right mouse button. These menus contain options specific for the object under the mouse.

5. **Designing an inquiry (or change or delete) program so that the user needs to enter only the first few characters of a name or item description.** The program displays a list of all matching names, and, when the user chooses one, the matching record is displayed.
6. **Providing keystrokes for selecting pull-down menu options.** Often, these options are selected using a mouse, followed by keying. Users must move their hands from the keyboard to the mouse and back. As users become familiar with the system, shortcut keystrokes provide a faster method for manipulating the pull-down menus, because both hands remain on the keyboard. This helps users become efficient at their tasks. On a PC, keystrokes usually involve pressing a function key or the Alt key followed by a letter. Figure 14.15 is an example of nested pull-down menus with shortcut keys from Microsoft Visio Professional. Notice that the user, who is creating a structure chart, can get into a series of ever more specific menus.
7. **Use radio buttons and drop-down lists to control displays of new Web pages or to change Web forms.** For example, when a radio button is clicked, a drop-down list may change to reflect the radio button choice. A radio button may be clicked and a form may change according to the choice. A drop-down list may change or a radio button may be clicked to move to a new Web page. Drop-down lists are often provided on a Web page for quick navigation, selecting a new Web page from the drop-down list takes the viewer to that page.
8. **Provide cursor control for Web forms and other displays so that the cursor moves to the next field when the right number of characters has been entered.** An example would be when a user enters an area code for a telephone number, and, following the entry of three characters, the cursor then moves to the

FIGURE 14.15

Example of nested pull-down menus with shortcut keys from Microsoft Visio Professional.



local phone number field. Entering software registration key codes is another example. The codes are often in groups of four or five letters and, when the first field is filled, the cursor moves to the next field and so on. The analyst should examine every field to see if automatic cursor control should occur.

Any combination of these eight approaches can help the analyst decrease the number of keystrokes required by the user, thereby speeding up data entry and minimizing errors.

STANDARD OPERATION AND CONSISTENCY

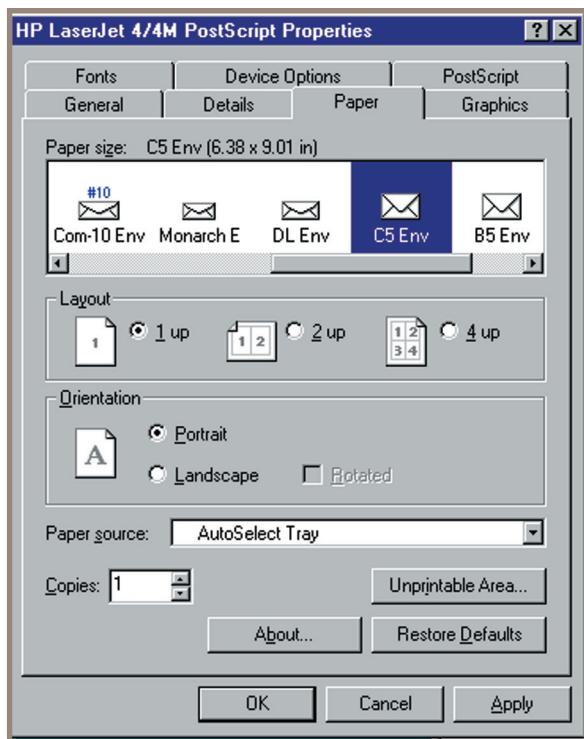
The system should be consistent throughout its set of different displays and in the mechanisms for controlling the operation of the displays throughout different applications. Consistency makes it easier for users to learn how to use new portions of the system once they are familiar with one component. You can achieve consistency by:

1. Locating titles, date, time, and operator and feedback messages in the same places on all displays.
2. Exiting each program by the same key or menu option. On Web pages there is often a logout image. When using programs such as Microsoft Access, use a common icon to exit the display screen. When using function keys to exit, it would be a poor design that used function key 4 (F4) to exit the ADD CUSTOMER program and function key 6 (F6) to exit the CHANGE CUSTOMER program.
3. Canceling a transaction in a consistent way, usually by pressing the Esc key on a PC or by using a function key (usually F12) on a mainframe.
4. Obtaining help in a standardized way. The industry standard for help is function key 1 (F1), and most software developers for the PC are adopting this convention.
5. Standardizing the colors used for all displays or Web pages. Error messages are typically displayed in red. Use a standard combination of a color and an icon to denote an error. Remember to keep the background screen color the same for all applications.
6. Standardizing the use of icons for similar operations when using a graphical user interface. For example, a small piece of paper with a bent upper corner often represents a document. Use standard images on a Web page.
7. Using consistent terminology in a display screen or Web site.
8. Providing a consistent way to navigate through the dialog. For example, find a consistent way to add records or to work with a Web site, such as using the same buttons for **Back** and **Next**.
9. Using consistent font alignment, size, and color on a Web page.

An example of good GUI design is the tab control dialog box shown in Figure 14.16. Currently, the user is choosing HP LaserJet print options, and he or she is in the **Paper** tab but also has the choice of six other tabs, including **Fonts** and **Graphics**. This display shows the options that a user can select by clicking on the left or right arrows on the horizontal sliding bar that runs along the bottom of the **Paper** size window: Com-10 Env, Monarch E, DL Env, C5 Env, and so on. The dark highlight indicates that the user has chosen to print a C5 envelope. Notice that the designer of this interface has used option buttons for both **Layout** and **Orientation**. The user has clicked on a choice of Portrait for orientation. A drop-down menu is also used to select the **Paper** source. In this instance, the user has chosen AutoSelect Tray. The designer has also used push buttons at the very bottom of the display that allow users to enter **OK**, **Cancel**, or **Apply** in regard to the options they have just chosen.

FIGURE 14.16

This tab control dialog box has seven tabs. The chosen tab “Paper” appears as if it is in front of the other tabs.



FEEDBACK FOR USERS

All systems require feedback to monitor and change behavior, as discussed in Chapter 2. Feedback usually compares current behavior with predetermined goals and gives back information describing the gap between actual and intended performance.

Because humans themselves are complex systems, they require feedback from others to meet psychological and cognitive processing needs discussed earlier in this chapter. Feedback also increases human confidence. How much feedback is required is an individual characteristic.

When users interface with machines, they still need feedback about how their work is progressing. As designers of user interfaces, systems analysts need to be aware of the human need for feedback and build it into the system. In addition to text messages, icons can often be used. For example, displaying an hourglass while the system is processing encourages the user to wait a while rather than repeatedly hitting keys to get a response.

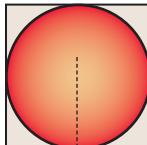
Feedback to the user from the system is necessary in seven distinct situations, as shown in Figure 14.17. Feedback that is ill timed or too plentiful is not helpful, because humans possess a limited capacity to process information. Each of the seven situations in which feedback is appropriate is explained in the upcoming

FIGURE 14.17

Feedback is used in many ways.

Feedback Is Needed to Tell the User That:

- The computer has accepted the input.
- The input is in the correct form.
- The input is not in the correct form.
- There will be a delay in the processing.
- The request has been completed.
- The computer is unable to complete the request.
- More detailed feedback is available (and how to get it).



WAITING TO BE FED

"Yeah, we were sold a package all right. This one right here. Don't get me wrong, it gets the work done. We just don't know when."

You are talking with Owen Itt, who is telling you about the sales unit's recent purchase of new software for its networked PCs that allows the input of sales data for each of its 16 salespeople, provides output showing comparison data for them, and projects future sales based on past sales records.

"We've had some odd experiences with this program, though," Owen continues. "It seems slow or something. For instance, we're never sure when it's done. I type in a command to get a file and nothing happens. About half a minute later, if I'm lucky, the display I want might come up, but I'm never sure. If I ask it to save sales data, I just get a whirring sound. If it works, I'm returned to where I was before.

If it doesn't save data, I'm still returned to where I was before. It's confusing, and I never know what to do. There's nothing on the display screen that tells me what to do next. See the little manual that came with it? It's dog-eared because we have to keep thumbing through trying to figure out what to do next. Or we go online to try to get some help, but their technical assistance is just about nonexistent. It takes way too much time, too."

Based on what you've heard in the interview, take this opportunity to supplement the software by designing some onscreen feedback for Owen and his sales team. The feedback should address all of Owen's concerns, and follow the guidelines for giving feedback to users, and the guidelines for designing good displays. Draw a prototype of the displays you think are necessary to address the problems Owen lists.

subsections. Web sites should display a status message or some other way of notifying the user that the site is responding and that input is either correct or in need of further information.

TYPES OF FEEDBACK

Acknowledging Acceptance of Input The first situation in which users need feedback is to learn that the computer has accepted the input. For example, when a user enters a name on a line, the computer provides feedback to the user by advancing the cursor one character at a time when the letters are entered correctly. A Web example would be a Web page displaying a message that "Your payment has been processed. Your confirmation number is 1234567. Thank you for using our services."

Recognizing That Input Is in the Correct Form Users need feedback to tell them that the input is in the correct form. For example, a user inputs a command, and the feedback states "READY" as the program progresses to a new point. A poor example of feedback that tells the user that input is in the correct form is the message "INPUT OK," because that message takes extra space, is cryptic, and does nothing to encourage the input of more data. When placing an order on the Web or making a payment, a confirmation page often displays, requesting that the user review the information and click a button or image to confirm the order or payment.

Notifying That Input Is Not in the Correct Form Feedback is necessary to warn users that input is not in the correct form. When data are incorrect, one way to inform the user is to generate a window that briefly describes the problem with the input and tells how the user can correct it, as shown in Figure 14.18.

Notice that the message concerning an error in inputting the subscription length is polite and concise but not cryptic, so that even inexperienced users will be able to understand it. The subscription length entered is wrong, but the feedback given does not dwell on the user's mistake. Rather, it offers options (13, 26, or 52 weeks) so that the error can be corrected easily. On a GUI screen, feedback is often in the form of a message box with an OK button on it.

FIGURE 14.18

Feedback informs the user that input was not in the correct form and lists options.

The screenshot shows a web-based form for a newspaper subscription. The form fields are as follows:

- First Initial: M
- Middle Initial: C
- Last Name: HURST
- Number: 3349
- Street: SOUTH STREET
- Apartment: (empty)
- City: LINCOLN
- State: NE
- Zip Code: 68506
- Subscription Length in Weeks: 14
- Method of Payment: CHK

A red-bordered box contains an error message: "The subscription length you entered is not currently being offered. Please choose either 13, 26, or 52 weeks".

Web messages have a variety of formats. One method is to return a new page with the message on the side of the field containing the error. The new Web page may have a link for additional help. This method works for all Web sites, and the error detection and formatting of the new page are controlled by the server. Another method uses JavaScript to detect the error and display a message box on the current screen with details about the specific error. An advantage of this method is that the Web page does not have to be sent to the server, and the page is more responsive. Disadvantages are that, if JavaScript is turned off, the error will not be detected, and only one error is displayed at a time. There must also be a way of detecting the error on the server. A second disadvantage is that JavaScript may not detect errors that involve reading database tables, such as verifying a credit card number. This may be offset by using Ajax, which can send the number to the server and return an error to the Web page. Remember, however, that as many as 25 percent of users intentionally turn off their JavaScript capability; so analysts need to follow a variety of tactics when communicating errors.

Web pages may also use JavaScript to detect multiple errors and display text messages on the page. Caution must be used so that the error messages are bold enough for the user to notice. A small red line of text may go unnoticed. A message box or audible beeps may be used to alert the users that one or more errors have occurred.

The analyst must decide whether to detect and report errors when a Submit button or link is clicked, called batch validation, or detect errors one at a time, such as when a user enters a month of 14 and leaves the field. The second method is a riskier approach since poor coding may put the browser into a loop, and the user will have to shut down the browser.

So far, we have discussed visual feedback in text or iconic form, but many systems have audio feedback capabilities as well. When a user inputs data in the incorrect form, the system might beep instead of providing a window. But audio feedback alone is not descriptive, so it is not as helpful to users as onscreen instructions. Use audio feedback sparingly, perhaps to denote urgent situations. The same

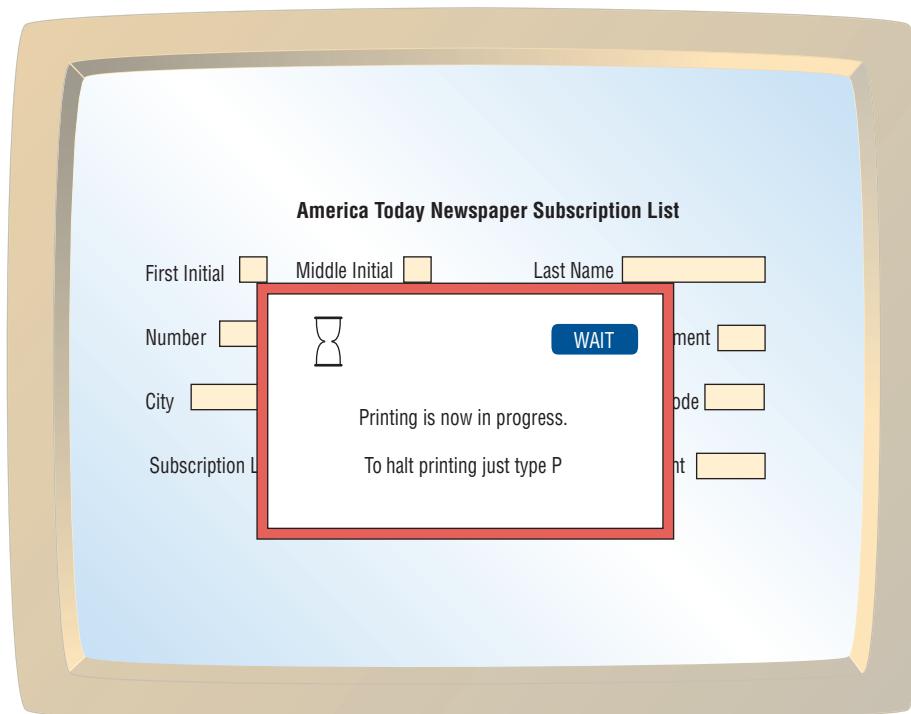


FIGURE 14.19

Feedback tells the user that there will be a delay during printing.

advice also applies to the design of Web sites, which may be viewed in an open office, where sounds carry and a coworker's desktop speakers are within earshot of several other people.

Explaining a Delay in Processing One of the most important kinds of feedback informs the user that there will be a delay in processing his or her request. Delays longer than 10 seconds or so require feedback so that the user knows the system is still working.

Figure 14.19 shows a display providing feedback in a window for a user who has just requested a printout of the newspaper's subscription list. The display shows a sentence reassuring the user that the request is being processed, as well as a sign in the upper right corner instructing the user to "WAIT" until the current command has been executed. The display also provides a way to stop the operation if necessary.

Sometimes during delays, while new software is being installed, a short tutorial on the new application is run, which is meant to serve as a distraction rather than feedback about the installation. Often, a list of files that are being copied and a status bar are used to reassure the user that the system is functioning properly. Web browsers usually display the Web pages that are being loaded and the time remaining.

It is critical to include feedback when using Ajax to update Web forms. Because a new Web page does not load, the user may not be aware that data are being retrieved from the server that will change the current Web page. When a drop-down list is changing, a message, such as "Please wait while the list is being populated" informs the user that the Web page is changing.

Timing feedback of this sort is critical. Too slow a system response could cause the user to input commands that impede or disrupt processing.

Acknowledging That a Request Is Completed Users need to know when their request has been completed and new requests may be input. Often a specific feedback

message is displayed when an action has been completed by a user, such as “Employee record has been added,” “Customer record has been changed,” or “Item number 12345 has been deleted.”

Notifying That a Request Was Not Completed Feedback is also needed to let the user know that the computer is unable to complete a request. If the display reads “Unable to process request. Check request again,” the user can then go back and check to see if the request has been input correctly rather than continue to enter commands that cannot be executed.

Offering the User More Detailed Feedback Users need to be reassured that more detailed feedback is available, and they should be shown how they can get it. Commands such as Assist, Instruct, Explain, and More may be employed. Or the user may type a question mark or point to an appropriate icon to get more feedback. Using the command Help as a way to obtain further information has been questioned, because users may feel helpless or caught in a trap from which they must escape. This convention is in use, and its familiarity to users may overcome this concern.

When designing Web interfaces, hyperlinks can be embedded to allow the user to jump to the relevant help screens or to view more information. Hyperlinks are typically highlighted with underlining, italics, or a different color. Hyperlinks can be graphics, text, or icons.

INCLUDING FEEDBACK IN DESIGN

It is well worth the systems analyst’s time to provide user feedback. If used correctly, feedback can be a powerful reinforcer of users’ learning processes, serve to improve user performance with the system, increase motivation to produce, and improve the fit among the user, the task, and the technology.

A Variety of Help Options Feedback on personal computers has developed over the years. “Help” originally started as a response to the user who pressed a function key, such as F1; the GUI alternative is the pull-down help menu. This approach was cumbersome, because end users had to navigate through a table of contents or search via an index. Next came context-sensitive help. Users could simply click on the right mouse button, and topics or explanations about the current screen or area of the screen would be revealed. Some COTS software manufacturers call these cue cards. A third type of help on personal computers occurs when the user places the arrow over an icon and leaves it there for a couple of seconds. At this point, some programs pop up a balloon similar to those found in comic strips. This balloon explains a little bit about the icon function.

The fourth type of help is a wizard, which asks the user a series of questions and then takes action accordingly. Wizards help users through complicated or unfamiliar processes such as setting up network connections or booking an airline seat online. Most users are familiar with wizards through creating a PowerPoint presentation or choosing a style for a word processing memo.

Besides building help into an application, software manufacturers offer online help (either automated or personalized with live chat) or help lines (most customer service telephone lines are not toll free, however). Some COTS software manufacturers offer a fax-back system. A user can request a catalog of various help documents to be sent by fax, and then can order from the catalog by entering the item number with a touch-tone phone.

Finally, users can seek and find support from other users through software forums. This type of support is, of course, unofficial, and the information thus obtained may be true, partially true, or misleading. The principles regarding the use of software forums are the same for those mentioned in Chapter 16, where folklore and recommendation systems are discussed. Beware! Read this section before you accept what is said on bulletin boards.

Besides informal help on software, vendor Web sites are extremely useful for updating drivers, viewers, and the software itself. Most computer magazines have some sort of “driver watch” or “bug report” that monitors the bulletin boards and Web sites for useful programs that can be downloaded. Programs will forage vendor Web sites for the latest updates, inform the user of them, assist with the downloads, and actually upgrade user applications.

SPECIAL DESIGN CONSIDERATIONS FOR ECOMMERCE

Many of the user interface design principles you have learned concerning feedback also extend to designing ecommerce Web sites. A few extra considerations shown in this section can give your Web interface designs improved functionality. They include learning to incorporate methods for eliciting feedback on the Web site from ecommerce customers and four ways to provide one-click navigation on ecommerce sites to ensure that customers can easily navigate the site and that they can readily return to it.

SOLICITING FEEDBACK FROM ECOMMERCE WEB SITE CUSTOMERS

Not only do you need to give users feedback about what is happening with an order, but you need to elicit feedback as well. Most ecommerce Web sites have a **Feedback** button. There are two standard ways to design what users will experience when they click on the **Feedback** button.

The first way is to launch the user’s email program with the email address of the company’s contact automatically entered into the **To:** field. This method prevents typing errors and facilitates ease in contacting the organization. The user does not need to leave the site to communicate with it. These messages, however, raise expectations that they will be answered just as regular mail or phone calls are. Research indicates that 60 percent of organizations with this type of email contact feature on their sites do not have anyone assigned to reply to the email messages received. Thus, the business is losing valuable feedback, allowing customers to harbor the impression that they are communicating, and engendering ill will when no response is received. If you design this type of feedback opportunity, you also need to design procedures for the organization to reply to email from the Web site. Some designers handle this problem by creating systems to automatically return an email reply, which generates a unique case or incident number, provides further instructions on how to proceed (hyperlinks to FAQ pages perhaps), or offers phone numbers to help lines that are unavailable to the general public.

The second type of design for garnering feedback from customers using an ecommerce Web site is to take users to a blank message template when they click on **Feedback**. Even a familiar tool such as Microsoft FrontPage permits you to create and insert a feedback form into your site easily. This form might begin with a header that states “Company X Feedback” and then “You can use the form below to send suggestions, comments, and questions about the X site to our Customer Service team.”

WHEN YOU RUN A MARATHON, IT HELPS TO KNOW WHERE YOU'RE GOING

Marathon Vitamin Shops was successful in getting its Web site up and running. The Web developers put the company's entire catalog online and included a choice of skins so that each type of customer would enjoy using the Web site. (See Consulting Opportunities 1.1 and 12.4 for more details.)

The analysts are meeting with owner Bill Berry and some employees to evaluate customer feedback as well as give their own reactions to the new Web site. They are meeting in a large conference room, where they have a computer with Internet access and a projector. As they sit down at the table, the entry screen for the Web site is projected at the front of the room. "The Web site has attracted lots of attention, but we want to give the customers even more so that they keep coming back," says Bill, gesturing to the screen.

He continues, "It's not like we're closing our retail stores or anything. In fact, it's just the opposite. When customers notice we're on the Web, they're eager to locate the store in their community. They want to be able to walk into a store and talk to a trained expert rather than buying everything over the Internet. We need to tell people how to get there."

"We think we can improve the site by adding special enhancements and features," says Al Falfa, a member of the systems team who originally developed and implemented the ecommerce Web site.

"Yes," says Ginger Rute, one of the other members of the systems development team, as she nods in agreement. "Blockbuster and Borders use a mapping facility from MapQuest, and Home Depot uses maps from Microsoft Vicinity, which also produces MapBlast!"

Vita Minn, another member of the original systems development team, speaks up enthusiastically, saying, "We know of a couple good message board services and chat rooms we can build into our Web site. We think they can improve the stickiness of the site, making people stay on the site longer and also making them want to return."

"That's a great idea," says Jin Singh, one of the technologically savvy Marathon employees. "We can let customers talk with one another, tell each other about a product they liked, and so on. We could even let them start their own blogs."

Vita continues by moving to the computer keyboard and saying, "Let me show you the sites at www.planetgov.com and www.worldviewer.com." As she types in the first URL, the group sees the site projected. "They use chat systems from ichat and Multicity.com, respectively," she continues.

"Customers also need to search for more information about a product or manufacturer," Al adds. "Let's make it easier for them. Let's look at www.Cincinnati.com for an example. They use Atomz to search for information."

After listening intently, Bill speaks up. "Medical information could also be useful" he says. "I've noticed that www.medpool.com has medical news from NewsEdge. I've seen people on the treadmills at my health and fitness center watching the financial channels while they exercise."

"While we're at it, why don't we add news and financial information to the Web site?" Ginger asks. "I notice that www.nmm.com has market news from a company called Moreover.com."

Think about the conversation between the systems development team and the people from Marathon Vitamin shops. Some of the enhancement suggestions involved taking advantage of free services; others required payments ranging from \$1,000 to \$5,000 annually. Although some were good ideas, others may not be practical or feasible. Perhaps some of the ideas just do not make sense for the company.

For each of the following, review what you know about the mission and business activities of Marathon Vitamin Shops. Then make a recommendation regarding each option the analysts and clients have made and defend it:

- Mashups using Google Maps.
- Chat rooms and message boards.
- Blogs.
- Search engines.
- Medical information.
- News feeds and financial markets information.

Fields can include First Name, Last Name, Email Address, Regarding (a subject field that supplies a drop-down menu of the company's product or service selections, asking the user to "Please make a selection"), an "Enter Your Message Here:" section (a free-form space where users can type in their message), and the standard **Submit** and **Clear** buttons at the bottom of the form. Using this type of form permits the analyst to have the user data already formatted correctly for storage in a database. Consequently, it makes the data entered into a feedback form easier to analyze in the aggregate.

Thus, the analyst does more than just design a response to individual email. The analyst helps the organization capture, store, process, and analyze valuable

customer information in a manner that makes it more likely that the company will be capable of spotting important trends in customer response, rather than simply reacting to individual queries.

EASY NAVIGATION FOR ECOMMERCE WEB SITES

Many authors speak of what is known as “intuitive navigation” for ecommerce Web sites. Users need to know how to navigate the site without having to learn a new interface and without having to explore every inch of the Web site before they can find what they want. The standard for this type of navigational approach is called one-click navigation.

There are four ways to design easy, one-click navigation for an ecommerce site: (1) creating a rollover menu, (2) building a collection of hierarchical links so that the home page becomes an outline of the key topic headings associated with the Web site, (3) placing a site map on the home page and emphasizing the link to it (this would also be placed on every other page on the site), and (4) placing a navigational bar on every inside page (usually at the top or on the left side of the page) that repeats the categories used on the entry screen.

A rollover menu can be created with a Java applet or with JavaScript and HTML layers, if you do not want to make users run a Java applet. The rollover menu appears when the customer using the Web site pauses the pointer over a link.

Creating an outline of the content of the site through the presentation of a table of contents on the home page is another way to speed navigation of the site. This design, however, imposes severe constraints on the designer’s creativity, and sometimes simply presenting a list of topics does not adequately convey the strategic mission of the organization to the user.

Designing and then prominently displaying the link to a site map is a third way to improve navigational efficiency. Remember to include the link to the site map on the home page and on every other page as well.

Finally, you can design navigation bars that are consistently displayed on the home page as well as at the top and on the left of all other pages that comprise the site. Once you have established (during the information requirements phase) the most useful and most used categories (usually categories such as “Our Company” “Our Products,” “Buy Now,” “Contact Us,” “Site Map,” and “Search”), remember to include them on all pages.

Including a search function is another option. Microsoft FrontPage extensions and other software have search capabilities built in; other possibilities include adding a search engine such as Google to your site. Simple search functions work well for small, manageable sites, but as a site grows large, advanced search functions that include Boolean logic (discussed later in this chapter) are needed.

Creating flexibility in the way users navigate the Web is also important. An expert Web site designer would try to incorporate many different ways to look up information on a particular subject. Figure 14.20 shows a Web page from DinoTech. For example, a user interested in an international IT career can find out information from the DinoTech Web site in three different ways. If they are interested in working in Argentina, they can click on the Argentine flag, click on the name of the country, or click on the map representing Argentina.

Designing a Web site with navigation for users with different cognitive processing or interests is desirable. It is even possible that the same user may use all three of these methods at different times. All of these add to the usability of a Web site.

The main priority in navigation is, however, that, whatever you do, you must make it extremely easy for users to return to a previous page and make it somewhat

FIGURE 14.20

An example of a Web page that allows users to navigate to the desired page in different ways.



easy to return to the place where they entered the client's site. Your main concern is keeping customers on the Web site. The longer customers are on the site, the greater the chance is that they will purchase something. So make sure that, if users navigate to a link in your client's Web site, they can easily find their way back. Doing these things will ensure the stickiness of the Web site. Do not create any barriers to the customer who wants to return to the client's Web site.

MASHUPS

An application programming interface (API) is a set of small programs and protocols used like building blocks for building software applications. When two or more APIs are used together they form a mashup. Many mashups are open source, so developers can use an API from a site like Google maps and combine it with an API that contains other data, resulting in a new Web site that creates an entirely new application.

Bogozo Real Estate (www.bogozo.com), for example, combines Craigslist real estate data with Google Maps. This new application allows a user to see properties displayed on a map, view the location of neighborhood schools, and in the specific case of New York City, see an overlay of a New York City subway map to help buyers find a place convenient to the subway system.

A large corporation that has many retail outlets in a region may want to make it easier for customers to find their retail stores. They may want to hire a company like Blipstar which provides a service that allows companies to upload information about retail stores. Blipstar geocodes them and places them on a Google map. The company then puts a link to this information on its own Web site, so customers can simply enter their ZIP or postal code and let the mashup display the location of the nearest retail store.

Mashups are becoming a new way to present information. Expect to see many useful mashup applications soon. Look for them at www.programmableweb.com.

DESIGNING QUERIES

When users ask questions of or communicate with the database, they are said to query it. Six different types of queries are among the most common. Your careful attention to query design can help reduce users' time spent in querying the database, help them find the data they want, and result in a smoother user experience overall.

QUERY TYPES

The questions we pose concerning data from our database are referred to as queries. There are six basic query types. Each query involves three items: an entity, an attribute, and a value. In each case, two of these are given, and the intent of the query is to find the remaining item. Figure 14.21 will be used to illustrate all of the query examples.

Query Type 1 The entity and one of the entity's attributes are given. The purpose of the query is to find the value. The query can be expressed as follows:

What is the value of a specified attribute for a particular entity?

Sometimes it is more convenient to use notation to formulate the query. This query can be written as

$$V \leftarrow (E, A)$$

where V stands for the value, E for entity, and A for attribute, and the variables in parentheses are given.

The question

What did employee number 73712 make in year 2003?

can be stated more specifically as

What is the value of the attribute YEAR-2003 for the entity EMPLOYEE NUMBER 73712?

The record containing employee number 73712 will be found, and the answer to the query will be \$47,100.

EARNINGS-HISTORY							
EMPLOYEE NUMBER	EMPLOYEE NAME	DEPARTMENT	S/H	YEAR-2000	YEAR-2001	YEAR-2002	YEAR-2003
72845	Waters	Outside Sales	S	48,960	51,400	49,050	52,900
72888	Dryne	Outside Sales	S	42,200	44,700	48,020	50,580
73712	Fawcett	Distribution	H	43,500	45,500	46,780	47,100
80345	Well, Jr.	Marketing	S	65,000	71,000	75,000	78,000
84672	Piper	Maintenance	H	40,560	42,340	43,520	44,910
'60	Acquia	Accounting	H	38,755	40,040	41,380	42,540

FIGURE 14.21

It is possible to perform six basic types of queries on a table that contains entities, attributes, and values.

Query Type 2 The intent of the second query type is to find an entity or entities when an attribute and value are given. Query type 2 can be stated as follows:

What entity has a specified value for a particular attribute?

Because values can also be numeric, it is possible to search for a value equal to, greater than, less than, not equal to, greater than or equal to, and so on. An example of this type of query is as follows:

What employee(s) earned more than \$50,000 in 2003?

The notation for query type 2 is

$$E \leftarrow (V, A)$$

In this case, three employees made more than \$50,000, so the response will be a listing of the employee numbers for the three employees: 72845, 72888, and 80345.

Query Type 3 The purpose of this query type is to determine which attributes fit the description provided when the entity and value are given. Query type 3 can be stated as follows:

What attribute(s) has a specified value for a particular entity?

This query is useful when many similar attributes have the same property. The following example has similar attributes (specific years) that contain the annual salaries for the employees of the company:

What years did employee number 72845 make over \$50,000?

or, more precisely,

What attributes {YEAR-2000, YEAR-2001, YEAR-2002, YEAR-2003}
have a value > 50,000 for the entity EMPLOYEE-NUMBER = 72845?

where the optional list in braces ({}) is the set of eligible attributes.

The notation for query type 3 is

$$A \leftarrow (V, E)$$

In this example, Waters (employee number 72845) made over \$50,000 for two years. Therefore, the response will be year 2001 and year 2003. Query type 3 is rarer than the preceding two types due to the requirement of having similar attributes exhibiting the same properties.

Query Type 4 Query type 4 is similar to query type 1. The difference is that the values of all attributes are desired. Query 4 can be expressed as follows:

List all the values for all the attributes for a particular entity.

An example of query type 4 is:

List all the details in the earnings history file for employee number 72888.

The notation for query type 4 is

$$\text{all } V \leftarrow (E, \text{ all } A)$$

The response for this query will be the entire record for the employee named Dryne (employee number 72888).

Query Type 5 The fifth type of query is another global query, but it is similar in form to query type 2. Query type 5 can be stated as follows:

List all entities that have a specified value for all attributes.

An example of query type 5 is:

List all the employees whose earnings exceeded \$50,000 in any of the years available.

The notation for query type 5 is

all $E \leftarrow (V, \text{all } A)$

The response to this query will be 72845, 72888, and 80345.

Query Type 6 The sixth query type is similar to query type 3. The difference is that query type 6 requests a listing of the attributes for all entities rather than a particular entity. Query type 6 can be stated as follows:

List all the attributes that have a specified value for all entities.

The following is an example of query type 6:

List all the years for which earnings exceeded \$40,000 for all employees in the company.

The notation for query type 6 is

all $A \leftarrow (V, \text{all } E)$

The response will be YEAR-2001, YEAR-2002, and YEAR-2003. As with query type 3, query type 6 is not used as much as other types.

Building More Complex Queries The preceding six query types are only building blocks for more complex queries. Expressions, referred to as Boolean expressions, can be formed for queries. An example of a Boolean expression is:

List all the customers who have Zip codes greater than or equal to 60001 and less than 70000, and who have ordered more than \$500 from our catalogs or have ordered at least five times in the past year.

One difficulty with this statement is determining which operator (for example, AND) belongs with which condition; it is also difficult to determine the sequence in which the parts of the expression should be carried out. The following may help to clarify this problem:

LIST ALL CUSTOMERS HAVING (ZIP-CODE GE 60001 AND ZIP-CODE LT 70000) AND (AMOUNT-ORDERED GT 500 OR TIMES-ORDERED GE 5)

Now some of the confusion is eliminated. The first improvement is that the operators are expressed more clearly as GE, GT, and LT than as English phrases, such as “at least.” Second, the attributes are given distinct names, such as AMOUNT-ORDERED and TIMES-ORDERED. In the earlier sentence, these attributes were both referred to as “have ordered.” Finally, parentheses are used to indicate the order in which the logic is to be performed. Whatever is in parentheses is done first.

Operations are generally performed in a predetermined order of precedence. Arithmetic operations are usually performed first (exponentiation, then either multiplication or division, and then addition or subtraction). Next, comparative operations are performed. These operations are GT (greater than), LT (less than),

HEY, LOOK ME OVER (REPRISE)

You have been called back to take another look at Merman's Costume Rentals. Here is part of the database created for Annie Oaklea of Merman's (with whom you last worked in Consulting Opportunities 7.1 and 8.1). The database contains information, such as the cost of the rental, the date checked out, the date due back, and the number of days the costume has been rented since the beginning of the year (YTD DAYS OUT) (see Figure 14.C1).

Analyzing Annie's typical day in the costume rental business, you realize there are several requests she must make of the database so that she can make decisions on when to replace frequently used

costumes or even when to buy more costumes of a particular type. She also needs to remember to keep in the good graces of customers she has previously turned down for a particular costume rental, to know when to recall an overdue costume, and so on.

Formulate several queries that will help her get the information she needs from the database. (*Hint:* Make any assumptions necessary about the types of information she needs to make decisions and use as many of the different query types discussed in this chapter as you can.) In a paragraph, describe how Annie's queries would be different if she were working with a Web-based or hyperlinked system.

COSTUME-RENTAL

COSTUME NUMBER	DESCRIPTION	SUIT NUMBER	COLOR	COST OF	DATE CHECKED OUT	DUE DATE	YTD DAYS OUT	TYPE OF COSTUME	REQUESTS TURNED DOWN
0003	Lady MacBeth F, SM	01	Blue	15.00	10/15	11/30	150	Standard	2
1342	Bear F, MED	01	Dk. Brown	12.50	10/24	11/09	26	Standard	0
1344	Bear F, MED	02	Dk. Brown	12.50	10/24	11/09	115	Standard	0
1347	Bear F, LG	01	Black	12.50	10/24	11/09	22	Standard	0
1348	Bear F, LG	02	Black	12.50	11/01	11/08	10	Standard	0
1400	Goldilocks F, MED	01	Light Blue	7.00	10/24	11/09	140	Standard	0
1402	Goldilocks F, MED	02	Light Blue	7.00	10/28	11/09	10	Standard	0
1852	Hamlet M, MED	01	Dark Green	15.00	11/02	11/23	115	Standard	3
1853	Ophelia F, SM	01	Light Blue	15.00	11/02	11/23	22	Standard	0
4715	Prince M, LG	01	White/purple	10.00	11/04	11/21	145	Standard	5
4730	Frog M, SM	01	Green	7.00	11/04	11/21	175	Standard	2
7822	Jester M, MED	01	Multi	7.50	11/10	12/08	12	Standard	0
7824	Jester M, MED	02	Multi	7.50	11/09	11/15	10	Standard	0
7823	Executioner M, LG	01	Black	7.00	11/19	12/05	21	Standard	0
8645	Mr. Spock N, LG	01	Orange	18.00	09/07	09/12	150	Trendy	4
9000	Pantomime F, LG	01	Red	7.00	08/25	09/15	56	Standard	0
9001	Pantomime M, MED	01	Blue	7.00	08/25	09/15	72	Standard	0
9121	Juggler M, MED	01	Multi	7.00	11/05	11/19	14	Standard	0
9156	Napoleon M, SM	01	Blue/white	15.00	10/26	11/23	56	Standard	1

FIGURE 14.C1

A portion of the database from Merman's Costume Rental shop.

and others. Finally, the Boolean operations are performed (first AND and then OR). Within the same level, the order generally goes from left to right. The precedence is summarized in Figure 14.22.

QUERY METHODS

Two popular query methods are query by example and structured query language.

Query by Example Query by example (QBE) is a simple but powerful method for implementing queries in database systems, such as Microsoft Access. The database

Type	Level	Symbol
Arithmetic Operators	1	* *
	2	* /
	3	+ -
Comparative Operators	4	GT LT EQ NE GE LE
Boolean Operators	5	AND
	6	OR

FIGURE 14.22

Arithmetic, comparative, and Boolean operators are processed in a hierarchical order of precedence unless parentheses are used.

fields are selected and displayed in a grid, and the requested query values are either entered in the field area or below the field. The query should be able to select both rows from the table that match conditions as well as specific columns (fields). Complex conditions may be set to select records, and the user may easily specify the columns to be sorted. Figure 14.23 is an example of query by example using Microsoft Access. The query design screen is divided into two portions. The top portion contains the tables selected for the query and their relationships, and the bottom portion contains the query selection grid. Fields from the database tables are dragged to the grid.

The first two rows contain the field and the table in which the field is located. The next row contains sorting information. In this example, the results will be sorted by CUSTOMER NAME. A check mark in the Show box (fourth row down) indicates that the field is to be displayed in the results. Notice that the CUSTOMER NUMBER, CUSTOMER NAME, and STATUS CODE MEANING are selected for the resulting display (other fields are displayed as well, but they do not show in the display). Notice that the ACCOUNT STATUS CODE and ACCOUNT TYPE CODE are not checked and therefore will not be in the final results. In the criteria rows, there is a 1 in the ACCOUNT STATUS CODE (indicating an active record) and a C and D (selecting a General Customer or a Discount Customer) in the

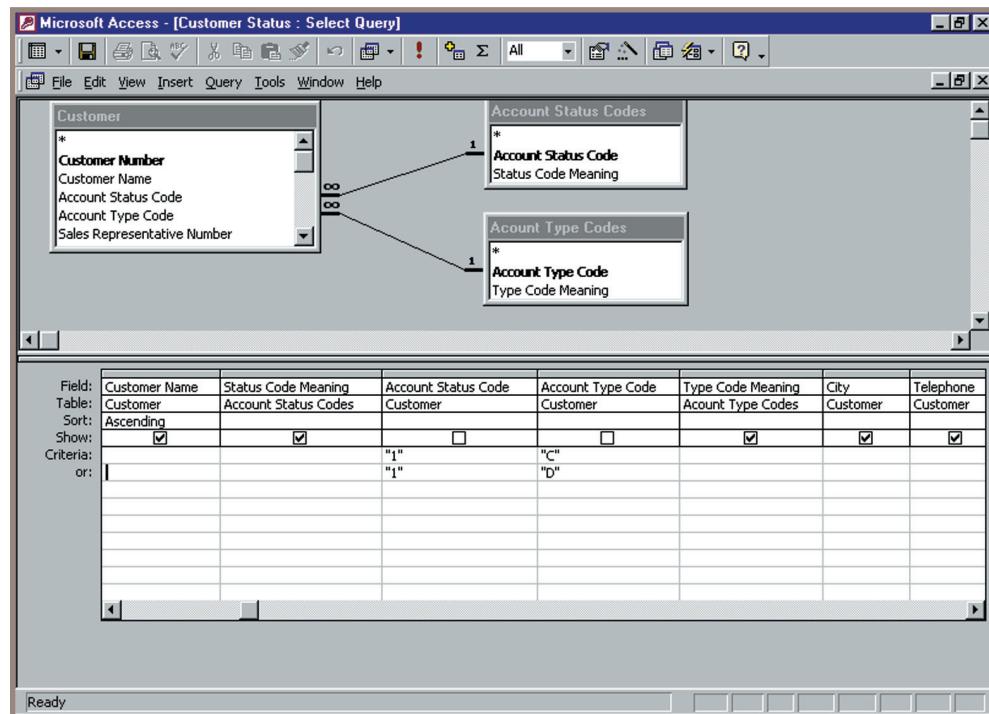


FIGURE 14.23

Query by example using Microsoft Access.

FIGURE 14.24

A query by example for CUSTOMER STATUS yields these results.

The screenshot shows a Microsoft Access window with the title bar 'Microsoft Access - [Customer Status : Select Query]'. The menu bar includes File, Edit, View, Insert, Format, Records, Tools, Window, and Help. The toolbar has various icons for file operations like Open, Save, Print, and Database. The main area displays a data sheet with the following columns: Number, Customer Name, Status Code Meaning, Type Code Meaning, City, and Telephone. There are 9 records listed:

Number	Customer Name	Status Code Meaning	Type Code Meaning	City	Telephone
18325	FilmMagic Video Rental	Active	Discount Customer	Madison	(801) 823-2287
16403	Gordon Builders	Active	General Customer	Sunnyvista	(415) 458-1364
14672	Industrial Cleaning Supply	Active	General Customer	Central Valley	(805) 263-8060
19592	Masterpiece Manuscripts	Active	General Customer	Camden	(000) 000-0000
17507	Music Unlimited	Active	General Customer	New York	(212) 334-9487
19844	Nathan's House of Pets	Active	Discount Customer	Milwaukee	(312) 238-9963
00201	POS Animation, Inc.	Active	Discount Customer	Oakhurst	
09288	Ursa Optical	Active	General Customer	Seattle	(206) 351-4999
19712	Wallaby Outfitters	Active	General Customer	Oakland	(415) 336-1114

Record: 1 2 3 4 5 6 7 8 9 | > >> * of 9
Datasheet View

ACCOUNT TYPE CODE columns. Two conditions in the same row indicate an AND condition, and two conditions in different rows represent an OR condition. This query specifies that the user should select both an Active Customer and either a General or Discount Customer.

The results of a query are displayed in a table, illustrated in Figure 14.24. Notice that the ACCOUNT STATUS CODE and ACCOUNT TYPE CODE do not display. They are not checked and are included in the query for selection purposes only. Instead, the code meanings are displayed, which are more useful to the user. The customer names are sequenced alphabetically.

FIGURE 14.25

A parameter query design screen.

The screenshot shows a Microsoft Access window with the title bar 'Microsoft Access - [Customer Parameter Query : Select Query]'. The menu bar includes File, Edit, View, Insert, Query, Tools, Window, and Help. The toolbar has various icons for file operations like Open, Save, Print, and Database. The main area displays a parameter query design screen. On the left, there is a 'Customer' table list with fields: Customer Number, Customer Name, Account Status Code, Account Type Code, Sales Representative Number, Street 1, Street 2, City, State, and Zip. On the right, there is a query grid with the following structure:

Field:	Customer Number	Customer Name	City	Telephone	
Table:	Customer	Customer	Customer	Customer	
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:		Like ([Enter a partial Customer Name] & "*")			
or:					

Ready

FIGURE 14.26

Structured Query Language (SQL) for the CUSTOMER NAME parameter query.

```
SELECT DISTINCTROW  
Customer.[Customer Number],  
Customer.[Customer Name],  
Customer.City,  
Customer.Telephone  
FROM Customer  
WHERE (((Customer.[Customer Name])  
Like ([Enter a partial Customer Name] & “*”)));
```

One of the problems encountered when designing queries is that either the user must modify the query parameters or the same conditions are selected each time the query is executed. A solution to this problem is to use a parameter query. This type of query allows the user to enter the conditions in a dialog box each time the query is run. Figure 14.25 illustrates a parameter query. Notice that the Criteria row has the message “Enter a partial Customer Name” included inside brackets. Preceding the message is the word “Like,” and following the message is an ampersand (&), indicating that an exact match is not required. When the query is executed, a dialog box opens with the query message on the top. If the value “ma” is entered in the box requesting a partial customer name, then only customers whose names begin with the letters “Ma” are selected and displayed.

Structured Query Language Structured query language (SQL) is another popular way to implement queries. It uses a series of words and commands to select the rows and columns that should be displayed in the resulting table. Figure 14.26 illustrates the SQL code that is equivalent to the preceding parameter query. The SELECT DISTINCTROW keyword determines which rows are to be selected. The WHERE keyword specifies the condition that the CUSTOMER NAME should be used to select the data entered in the LIKE parameter.

SUMMARY

In this chapter we have examined human–computer interaction (HCI), a variety of interfaces, designing the user interface, designing user feedback, and designing e-commerce Web site feedback and navigation. We have focused on understanding HCI to ensure the functionality and usability of computer systems we design. By doing so we intend to provide effective support for user interaction with technology, making system use a pleasant user experience that serves the well-being of the individual and the objectives of the organization. When analysts create a proper fit among the HCI elements of the human, the computer, and the task, it leads to improved performance and overall psychological and physical well-being of the individual.

Designs focus on such a fit, with one of the main concerns of the HCI designer being the overall psychological and physical well-being of the user. Analysts can use the TAM (Technology Acceptance Model) to organize their thinking about whether users will accept technology and eventually use it, by examining perceived usefulness and perceived ease of use experienced by users. This model has been used to predict the likelihood of acceptance and use, as well as to judge acceptance after implementation.



"I have no problem with using a mouse or any other rodent you throw my way. Really though, whatever Snowden needs is what I try to do. Everyone is different, however. I've seen people here go out of their way to avoid using a computer altogether. Other people would prefer not to talk with a human. In fact, they would be as happy as a puppy chewing on a new bedroom slipper if they could use command language to interact. I have a hunch they would prefer not talking to people at all, but that's just an impression. Most of the folks we have here are open to new things. Otherwise, they wouldn't be here at MRE. We do pride ourselves on our creativity. I have you signed up for a meeting with people from the training group, including Tom Ketcham, Melissa Smith, and Kathy Blandford. You can invite anyone else you think should be included. Snowden may sit in as well, if he has time. That's why he asked me to relay the message, I guess. They'll be very curious to see what kind of interface you are suggesting for them on the new project reporting system."

HYPERCASE QUESTIONS

1. Write a short proposal describing what type of user interface would be appropriate for the users of the project reporting system who are in the training group. Include reasons for your decision.
2. Design a user interface using a CASE tool, such as Visible Analyst, a software package such as Microsoft Access, or paper layout forms. What are the key features that address the needs of the people in the training group?
3. Demonstrate your interface to a group of students who can role-play as members of the training group. Ask for reactions.
4. Redesign the interface based on the feedback you have received. Write a paragraph to say how your new design addresses any comments you have received.



FIGURE 14.HC1

In HyperCase, you can see how users process information in order to create a more effective user interface.

Usability is all about finding out what works for users and what does not. Usability heuristics for judging the usability of computer systems and ecommerce sites include the visibility of system status, the match between the system and the real world, user control and freedom, consistency and standards, error prevention, reconnection rather than recall, flexibility and efficiency of use, aesthetic and minimalist design, help that users recognize, diagnosis and recovery from errors, and help and documentation. It is possible to extend the cognitive capabilities of individuals by creating systems that offer elements such as pivot tables and visual analysis.

Physical considerations of HCI design include vision, hearing, and touch. Physical disabilities and limitations should be taken into consideration during task and interface design. The guidelines for taking an HCI approach to systems design are as follows: (1) Examine the task to be done and consider the fit among the human, computer, and task. (2) Identify what obstacles exist for users in their attempts to accomplish their assigned tasks (keeping in mind the perceived usefulness and perceived ease of use from TAM). (3) Consider usability; examine the usage environment by creating use case scenarios that depict what is going on between users and the technology. (4) Use the information you have gained beforehand to figure out the physical and organizational environmental characteristics. Design with prototyping to accommodate diverse users and users with disabilities. The success of the systems you design depends on user involvement and acceptance. Therefore, thinking about users in systematic and empathic ways is of utmost importance and is not a peripheral issue for systems analysts.

A variety of user interfaces and input devices were covered. Some interfaces are particularly well suited to inexperienced users, such as natural language, question and answer, menus, form-fill and Web-based form-fill, graphical user interfaces (especially on Web pages), the mouse, light pens, the stylus, touch-sensitive screens, and voice recognition systems. Command language is better suited to experienced users.

Combinations of interfaces can be extremely effective. For example, using pull-down menus with graphical user interfaces or employing nested menus in question-and-answer interfaces yields interesting combinations. The Web has posed new challenges for designers, because the user is not known. Web design takes advantage of hyperlinks to allow users to take numerous paths as they interact with the Web site.

Users' need for feedback from the system is also an important consideration. System feedback is necessary to let users know if their input is being accepted, if input is or is not in the correct form, if processing is going on, if requests can or cannot be processed, and if more detailed information is available and how to get it. Feedback is most often visual, with text, graphics, or icons being used. Audio feedback can also be effective.

Special considerations apply to designing ecommerce Web sites. Build improved functionality into the application by eliciting customer feedback through automatic email feedback buttons or by including blank feedback forms on the Web site. In addition, four important navigation design strategies improve the stickiness of ecommerce Web sites: (1) rollover menus, (2) hierarchical displays of links on the entry screen, (3) site maps, and (4) navigation bars that provide one-click navigation, making getting around the site and returning to it as easy as possible for the customer.

Queries are designed to allow users to extract meaningful data from the database. There are six basic types of queries, and they can be combined using Boolean logic to form more complex queries.

KEYWORDS AND PHRASES

application programming interface (API)	one-click navigation
Boolean operators	performance
cognitive considerations of HCI	physical considerations of HCI
command-language interface	psychological considerations of HCI
continuous speech system	pull-down menu
cue cards	query
dialog box	question-and-answer interface
disabilities and design	rollover menu
feedback	site map
feedback for users	speech recognition and synthesis
fit	stickiness
form-fill (input/output form) interfaces	structured query language (SQL)
graphical user interface (GUI)	stylus
intuitive navigation	task
mashup	technology acceptance model (TAM)
menu	template
natural-language interface	touch-sensitive screen
navigation bar	usability
nested menus	Web-based form-fill interface
	wizard

REVIEW QUESTIONS

1. Define HCI.
2. Explain how fit among the HCI elements of the human, the computer, and the tasks to be performed leads to performance and well-being.
3. What are the components of the term *performance* in the HCI context?
4. What is meant by the word *well-being* when used using an HCI approach?
5. What are the two variables of the Technology Acceptance Model (TAM)?
6. List five of the eleven usability heuristics for judging the usability of computer systems and ecommerce Web sites provided by Nielsen and others.
7. Describe some of the ways that a pivot table allows a user to arrange data.
8. List three physical considerations that HCI design addresses.
9. List three ways that analysts can improve task or interface design to help, respectively, a person who is visually impaired, hearing impaired, or mobility impaired.
10. What are the five objectives for designing user interfaces?
11. Define natural-language interfaces. What is their major drawback?
12. Explain what is meant by question-and-answer interfaces. To what kind of users are they best suited?
13. Describe how users use onscreen menus.
14. What is a nested menu? What are its advantages?
15. Define onscreen input/output forms. What is their chief advantage?
16. What are the advantages of Web-based fill-in forms?
17. What are the drawbacks of Web-based form-fill interfaces?
18. Explain what command-language interfaces are. To what types of users are they best suited?
19. Define graphical user interfaces. What is the key difficulty they present for programmers?
20. For what type of user is a GUI particularly effective?
21. What are the three guidelines for designing good screen dialog?

22. What are the roles of icons, graphics, and color in providing feedback?
23. List eight ways for achieving the goal of minimal operator action when designing a user interface.
24. List five standards that can aid in evaluating user interfaces.
25. What are the seven situations that require feedback for users?
26. What is an acceptable way of telling the user that input was accepted?
27. When a user is informed that his or her input is not in the correct form, what additional feedback should be given at the same time?
28. List three ways to notify a Web user that the input is not in the correct form.
29. Why is it unacceptable to notify the user that input is not correct solely through the use of beeping or buzzing?
30. When a request is not completed, what feedback should be provided to the user?
31. Describe two types of Web site designs for eliciting feedback from customers.
32. List four practical ways that an analyst can improve the ease of user navigation and the stickiness of an ecommerce Web site.
33. What are hypertext links? Where should they be used?
34. Describe what a mashup is.
35. List in shorthand notation the six basic query types.

PROBLEMS

1. Manu Narayan owns several first-class hotels worldwide, including properties in Manhattan, Bombay, and even some in suburbia. He wants to make sure that the human-computer interface is appropriate to each culture but wants to be able to share the software among all of his hotel reservations departments. Design a nested menus interface for a check-in and checkout hotel reservation system that can be used internationally. Use numbers to select a menu item. Show how each menu would look on a standard PC display.
2. Stefan Lano needs displays that will show the musical instrument inventory in his chain of music stores that caters to musicians playing in world-class symphony orchestras in Basel, Switzerland; Buenos Aires, Argentina; Philadelphia, USA; and New York. Design a form-fill interface for the inventory control of musical instruments in all four stores that could be used on a PC display screen. Assume that English will be the interface language.
3. Design a Web-based form-fill interface to accomplish the same task as in Problem 2.
 - a. What difficulties did you encounter? Discuss them in a paragraph.
 - b. Of the two designs you did, which would you say is better suited to Mr. Lano's task? Why? List three reasons for your choice. How would you test their usability?
4. A U.K.-based travel agent, Euan Morton, LLC, would like your systems team to design a command-language interface he can use to book seats for airlines to which his firm has solid business ties, such as British Air, RyanAir, and Virgin-Atlantic.
 - a. Show what the interface would look like on a standard display.
 - b. Make a list of commands needed to book an airline seat and write down what each command means.
5. An IT executive, Felicia Finley, from Jersey IT Innovators, Inc., has asked that you design a graphical user interface for an executive desktop to help her in her work. Use icons for file cabinets, a wastebasket, a telephone, and so on. Show how they would appear on the computer display.

6. Nick, a celebrity chef/restaurant owner from Williamsburg, New York, wants to be able to receive clear feedback on the systems used to manage his many “show place” restaurants. Design a display that provides appropriate feedback for a user whose command cannot be executed.
7. Design a screen for a payroll software package that displays information telling Nick from Problem 6 how to get more detailed feedback.
8. Design a Web-based display that shows an acceptable way to tell Nick that input to his system was accepted.
9. Design a feedback form for Nick’s restaurant customers using an ecommerce Web site.
10. Write six different queries for the file in Problem 1 in Chapter 13.
11. Write six different queries for the 3NF relation in Problem 6 in Chapter 13.
12. Design a search that will find potential competitors of a company such as World’s Trend on the Web. Assume you are the customer.
13. Search for World’s Trend’s potential competitors on the Web. (You won’t find World’s Trend itself on the Web. It is a fictional company.) Make a list of those you’ve found.

GROUP PROJECTS

1. With your group members, create a pull-down menu for an employment agency that matches professional candidates to position openings. Include a list of keystrokes that would directly invoke the menu options using the Alt-X format. The menu has the following options:

Add employee	Match employee to opening
Change employee	Print open positions report
Delete employee	Print successful matches report
Employee inquiry	Add position
Position inquiry	Change position
Employer inquiry	Delete position
Add employer	
Change employer	
Delete employer	

2. In a paragraph, describe the problems your group faced in creating this menu.
3. The drag-and-drop feature is used in GUIs and allows the user to move sentences around in a word processing package. As a group, suggest how drag and drop can be used to its fullest potential in the following applications:
 - a. Project management software (Chapter 3).
 - b. Relational database program (Chapter 13).
 - c. Display or forms designer (Chapter 12).
 - d. Spreadsheet program (Chapter 10).
 - e. CASE tool for drawing data flow diagrams (Chapter 7).
 - f. Fax program (Chapter 11).
 - g. File management program (Chapter 14).
 - h. Personal digital assistant (PDA) calendar (Chapter 3).
 - i. Illustration in a drawing package (Chapter 10).
 - j. CASE tool for developing data dictionaries (Chapter 8).
 - k. Decision tree drawing program (Chapter 9).
 - l. Web site for collecting consumer opinions on new products (Chapter 11).
 - m. Organizing bookmarks for Web sites.

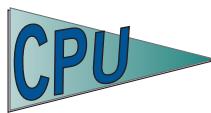
For each solution your group designs, draw the display and show movement by using an arrow.

4. Ask all the members of your group to request a search based on their leisure activities. If there are four people in your group, there will be four unique searches to perform. Now go ahead and do all the searches. Compare your results. Does the person who is involved with the activity have an advantage over the people who know less about it? Explain.
5. Look at the following mashup Web sites and describe how each of them adds value by providing a service.
 - a. Aboutairportparking.com, <http://www.aboutairportparking.com/>
 - b. Baebo, <http://baebo.francisshanahan.com/>
 - c. Bogozo, <http://www.bogozo.com/house/?new+york>.
 - d. Global Incident Map, <http://www.globalincidentmap.com/home.php>
 - e. Hawkee Social Price Comparison, <http://www.hawkee.com/>
 - f. Homethinking, <http://www.homethinking.com/>
 - g. mpire, <http://www.mpire.com/buyer/search.page>
 - h. Streeteasy, <http://www.streeteasy.com/>
6. The following two mashups have political agendas. Suggest three other mashups that attempt to change something by appealing to the public. Health Care That Works, <http://www.healthcarethatworks.org/maps/nyc/> On NY Turf, <http://www.onnyturf.com/citycouncil/freedomzones/>
7. Try these mashups just for fun.
 - a. The Geography of Seinfeld, <http://www.stolasgeospatial.com/seinfeld.htm>
 - b. HBO: The Sopranos, <http://www.hbo.com/sopranos/map/>
 - c. Dig to the Other Side, <http://map.pequenopolis.com/>
 - d. PlotShot, <http://www.plotshot.com/>
 - e. Flickr Sudoku, <http://flickrsudoku.com/>
 - f. liveplasma.com, <http://www.liveplasma.com>
 - g. Ms. Dewey <http://www.msdewey.com/>

SELECTED BIBLIOGRAPHY

- Adam, A., and D. Kreps. "Web Accessibility: A Digital Divide for Disabled People?" In *Societal and Organizational Implications for Information Systems*, IFIP International Federation for Information Processing, Vol. 208, edited by E. Trauth, D. Howcraft, T. Butler, B. Fitzgerald, and J. DeGross, pp. 217–228. Boston: Springer, 2006.
- Barki, H., and J. Hartwick. "Measuring User Participation, User Involvement, and User Attitude." *MIS Quarterly*, Vol. 18, No. 1, 1994, pp. 59–82.
- Berstel, J., S. C. Reghizzi, G. Roussel, and P. San Pietro. "A Scalable Formal Method for Design and Automatic Checking of User Interfaces." *ACM Transactions on Software Engineering and Methodology*, Vol. 14, No. 2, April 2005, pp. 124–167.
- Bort, J. "Navigation: An Art for E-Com Sites." *Microtimes.com*, Issue 201, December 1999. Available at <microtimes.com/201/ecombert201a.html>. Last accessed February 9, 2001.
- Butler, T., and B. Fitzgerald. "A Case Study of User Participation in the Information Systems Development Process." In *Proceedings of the Eighteenth International Conference on Information Systems*, pp. 411–426. Atlanta, GA: ICIS, 1997.
- Carey, J., D. Galletta, J. Kim, D. Te'eni, B. Wildemuth, and P. Zhang. "The Role of Human–Computer Interaction in Management Information Systems Curricula: A Call to Action." *Communications of the Association for Information Systems*, Vol. 13, 2004, pp. 357–379.
- Davis, F. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology." *MIS Quarterly*, Vol. 13, No. 3, 1989, pp. 319–340.
- Davis, G. B., and M. H. Olson. *Management Information Systems: Conceptual Foundations, Structure, and Development*. New York: McGraw-Hill, 1985.

- Hornbaek, K., and E. Frokjaer. "Comparing Usability Problems and Redesign Proposals as Input to Practical Systems Development." *CHI 2005*, April 2–7, 2005, pp. 391–400.
- Laudon, K. C., and J. P. Laudon. *Management Information Systems*, 9th ed. Upper Saddle River, NJ: Prentice-Hall, 2005.
- Mantei, M. M., and Teorey, T. J. "Incorporating Behavioral Techniques in the System Development Lifecycle." *MIS Quarterly*, Vol. 13, No. 3, September 1989, pp. 257–267.
- Nielsen, J., R. Molich, C. Snyder, and S. Farrell. *E-Commerce User Experience*. Fremont, CA: Norman Nielsen Group, 2001.
- Nielsen, J., and R. L. Mack. *Usability Inspection Methods*. New York: John Wiley, 1994.
- Rubin, J. *Handbook of Usability Testing*. New York: John Wiley, 1994.
- Sano, D. *Designing Large-Scale Web Sites: A Visual Design Methodology*. New York: John Wiley, 1996.
- Schneiderman, B., and C. Plaisant. *Designing the User Interface: Strategies for Effective Human–Computer Interaction*. New York: Addison-Wesley, 2005.
- Te'eni, D., J. Carey, and P. Zhang. *Human Computer Interaction: Developing Effective Organizational Systems*. New York: John Wiley, 2007.
- U.S. Department of Health and Human Services. "Usability Guide" (for developing Web sites). Available at www.usability.gov. Last accessed August 7, 2006.
- U.S. Equal Employment Opportunity Commission Web site listing employer obligations under the Americans with Disabilities Act. Available at www.eeoc.gov/types/ada.html. Last accessed September 24, 2006.
- UsabilityNet. "Overview of the User Centred Design Process." Available at: http://www.usabilitynet.org/management/b_overview.htm. Last accessed August 6, 2006.
- Venkatesh, V., M. G. Morris, G. B. Davis, and F. D. Davis. "User Acceptance of Information Technology: Toward a Unified View." *MIS Quarterly*, Vol. 27, No. 3, pp. 425–478.
- Zhang, P., J. Carey, D. Te'eni, and M. Tremeaine. "Integrating Human–Computer Interaction Development into the Systems Development Life Cycle: A Methodology." *Communications of the Association for Information Systems*, Vol. 15, 2005, pp. 512–543.



ALLEN SCHMIDT, JULIE E. KENDALL, AND KENNETH E. KENDALL

UP TO THE USERS

14

"Let's take our prototypes and some new displays, reports, and forms to create the final user interface," Anna says to Chip.

"It's about time, isn't it?" replies Chip. He was all too aware of the importance of designing a good interface.

After talking, they set up the following display dialog guidelines:

1. Well-designed displays should:

Communicate actions and intentions clearly to users.

Show options available to operators. Examples are:

MAKE CORRECTIONS OR PRESS ESC TO CANCEL

ENTER HARDWARE INVENTORY NUMBER

PRESS ENTER KEY

PRESS ENTER TO CONFIRM DELETE, ESC TO CANCEL

Buttons that say OK or Cancel

Standardize use of any abbreviations.

Avoid the use of codes, substituting the code meaning.

Provide help screens for complicated portions of the dialog.

Provide tool-tip help for toolbar icons.

2. Feedback should be provided to the users. Feedback includes:

Titles to show the current page.

Actions successfully completed messages, such as:

RECORD HAS BEEN ADDED

RECORD HAS BEEN CHANGED

Error messages. Examples are:

INVALID DATE

CHECKDIGIT IS INVALID

SOFTWARE IS NOT ON FILE

An invalid data dialog box, with an OK button on a graphical user interface display.

Processing delay messages similar to:

PLEASE WAIT—REPORT IS BEING PRODUCED

An hourglass turning upside down on a graphical user interface.

3. There should be consistency in the design, including:

Location of the OPERATOR MESSAGE or FEEDBACK MESSAGE on the bottom of the display or in the status line area.

Date, time, system name, and display reference number appearing in heading lines.

Consistent exit of all displays, such as through the use of the same function key.

Standard use of keys, such as PgDn and PgUp, to display a next or previous page in a multiple-page display.

Consistent method of canceling an operation, such as through the use of the Escape key (Esc).

Standardized use of color and high-intensity display, such as all error messages appearing in red.

Standardized use of icons on a GUI display.

Standardized pull-down menus on a GUI display.

14

4. Minimum operator actions should be required to use the system. Some examples are:

The use of Y and N as yes and no replies. The use of the plus (+) and minus (−) signs on the number pad as a substitute for Y and N.

When changing or deleting records, only the record key need be specified. The system would obtain the record and display pertinent information.

When names are required as key entries, only the first few letters of the name need be entered. The program should find all matching record key names and present them for selection by the operator.

Data entry displays should allow the entry of codes.

All numeric entries may omit leading zeros, commas, or a decimal point.

As each data field is completed, the cursor should advance to the next entry field.

After each option is completed, the same display, with blank entry areas, should be redisplayed until the Exit key is pressed.

When an option is exited, the previous menu should be displayed.

Drop-down list boxes should be used whenever possible on GUI displays.

Check boxes and radio buttons should be used to make selections whenever possible.

Default buttons should be outlined so that the user can press the Enter key to select them.

5. Data entering the system should be validated. Guidelines are:

Specific fields should be verified according to edit criteria.

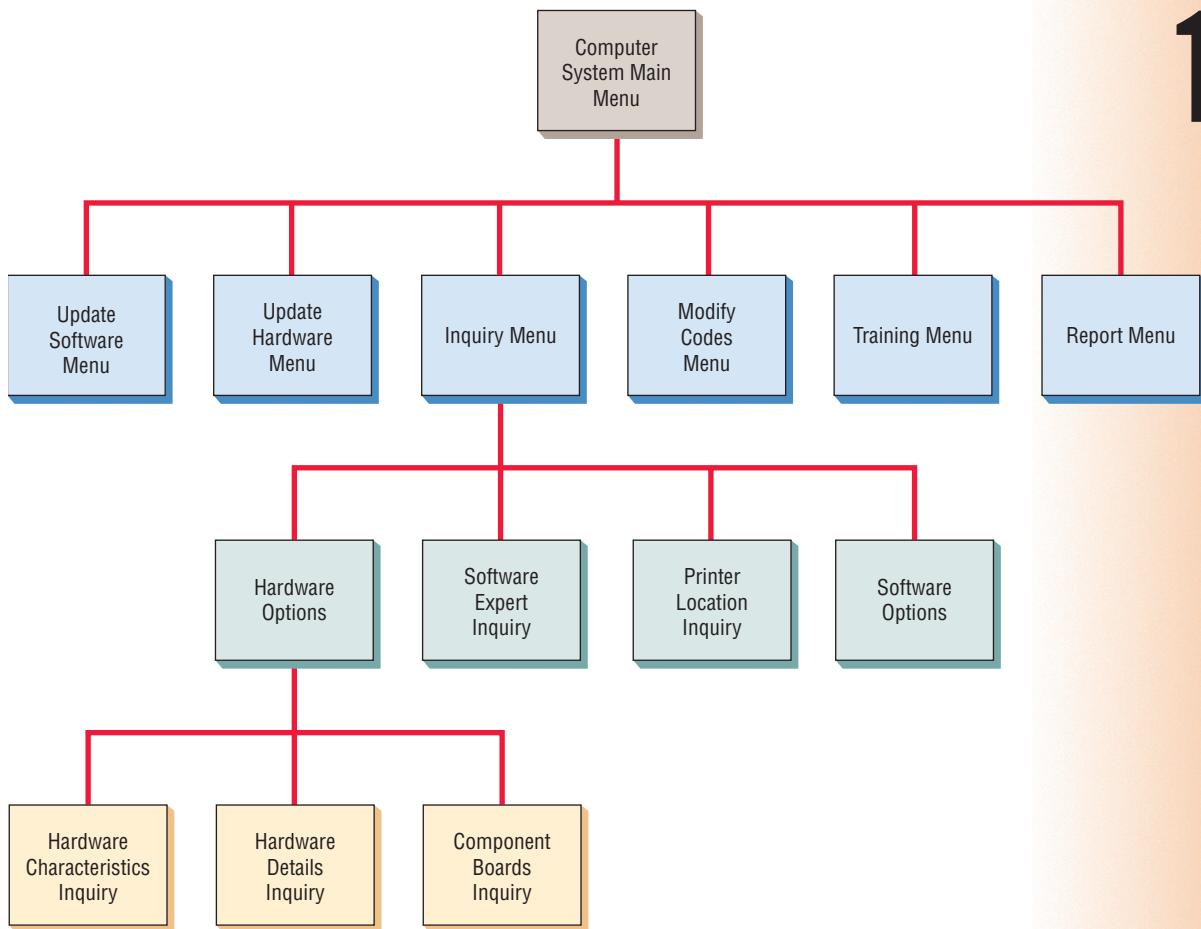
As errors are detected, operators should be given a chance either to correct the error or to cancel the transaction.

When no errors have been detected in a transaction, the entry should be presented to the operator for visual confirmation. The operator should have the opportunity either to accept it or to make corrections to the data entered.

Upon examining the many displays and reports (over 30 in all), Chip and Anna decide to split the menu into several functions. “How do we divide these various functions into a set of menus?” asked Chip.

“Why don’t we use a decomposition diagram to organize the functions into a hierarchy,” replied Anna. Chip and Anna begin working on the diagram. The menu interactions will be represented in a hierarchical structure, with options shown as rectangles and the overall menu represented by the rectangle on the top. Each secondary menu will be shown beneath the primary menu, with screen programs at the lowest level. The main menu will have six main choices, as illustrated in Figure E14.1: (1) **Update Software**, (2) **Update Hardware**, (3) **Inquiry**, (4) **Modify Codes**, (5) **Training**, and (6) **Report**. Each of these options is further subdivided into smaller menus or individual functions. The Inquiry Menu is subdivided into two smaller menus, **Software Options** and **Hardware Options**, as well as options for running the **Software Expert Inquiry** and the **Printer Location Inquiry**.

The rectangles on the functional decomposition diagram are implemented using a series of pull-down menu lists, which are shown in Figure E14.2. Notice that the Inquiry menu has functions corresponding to the rectangles on the previous figure. A row of buttons for common functions is included below the menus. The menu functions are included as a set of buttons in the main area of the screen, and these buttons may be clicked to run corresponding programs. It was decided that the **Add Computer**,

**FIGURE E14.1**

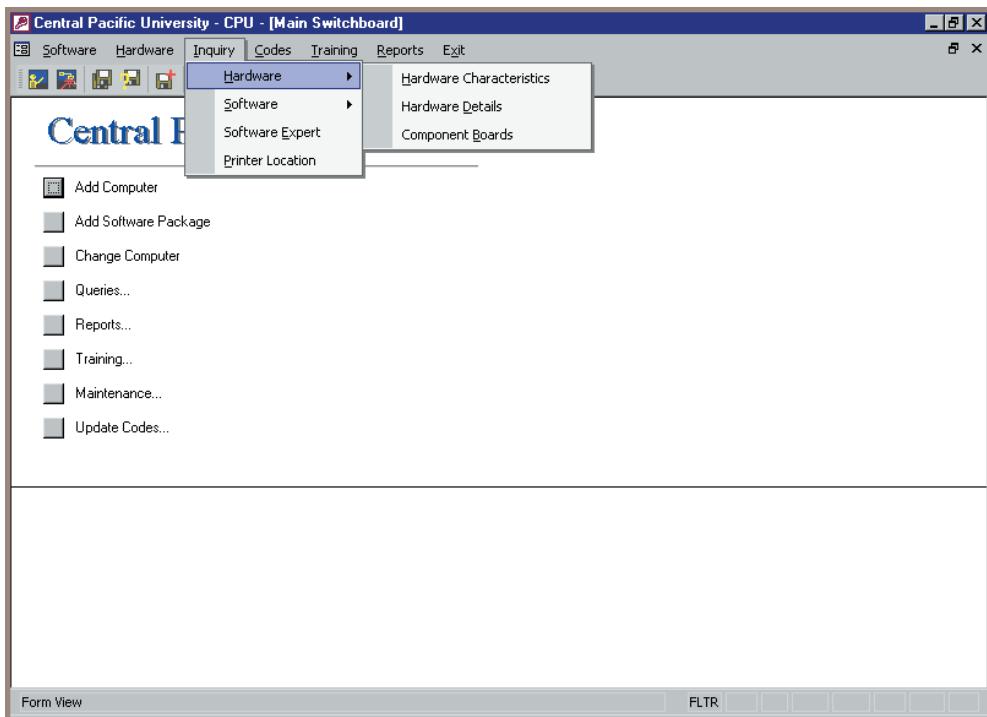
Screen hierarchy for the computer system.

Add Software Package, and **Change Computer** programs would be run directly from the main menu. Clicking the other buttons causes selection dialog boxes to display, with choices for selecting programs. Figure E14.3 shows the dialog box for the **Reports** option. All the reports are listed, with **Print Preview**, **Print**, and **Close Form** buttons for selecting actions.

"Here's what I think the guidelines for the update programs should be," Anna tells Chip. "The key focus is on accuracy, with comprehensive editing for each data field. Add programs will display an entry page and allow either hardware or software records to be created. After all entries are complete, a user should double-check the data and click the **Add Software Record** button. Any data that are already in the system should be implemented using drop-down lists. There are also buttons to undo changes, move to different records, print the record, save the changes, and exit the page. A record could be added only if the primary key for the record does not already exist."

"Delete displays must have a simple, primary key entry, such as the COURSE DESCRIPTION in the DELETE SOFTWARE COURSE display," Anna continues.

14

**FIGURE E14.2**

The main menu for the computer system.

The DELETE SOFTWARE COURSE display uses a Find button (the binoculars) to help locate the desired record. The corresponding record is read and the information is displayed. Users click the Delete button and are prompted to confirm the delete. If the user clicks Cancel, the delete action is canceled. How does all that sound?" she asks Chip.

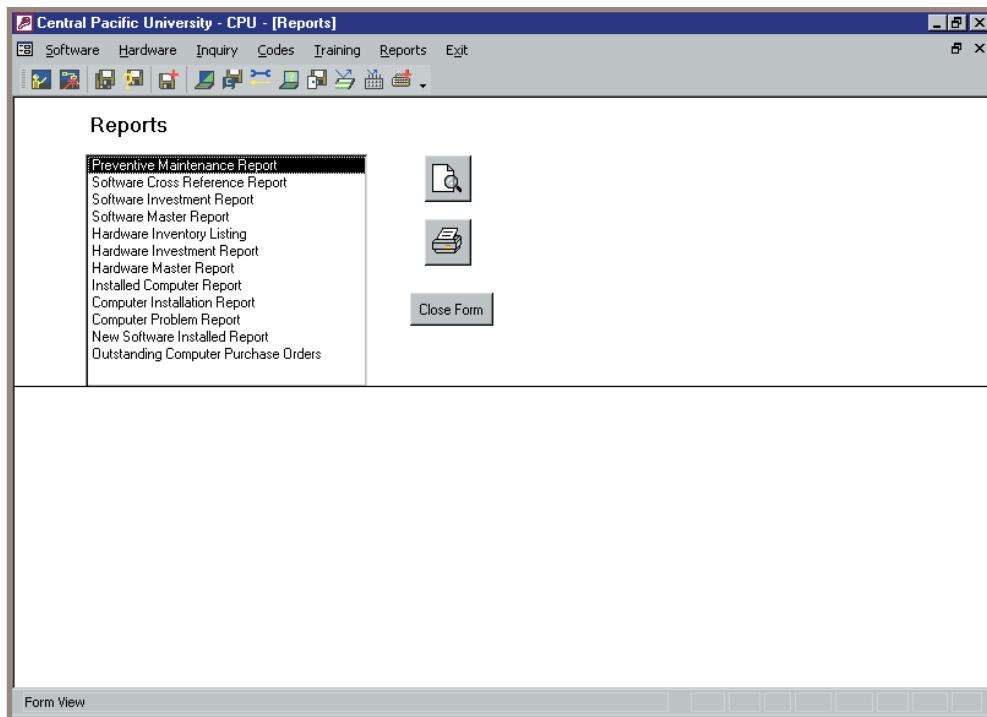
"So far, so good," he replies. "Anything about onscreen change displays?"

"Yes. They have a primary key for the record entered and the matching record read. Record information is to be displayed that allows the operator to overtype the data with changes. All changes are to be validated with full editing. When all change fields are valid, the user must click a button to save the changes. Is that clear enough for the user?" Anna asks.

"I think it's very good," Chip acknowledges.

Chip is responsible for the inquiry portion of the system. The focus on these programs is speed. A short entry is obtained from the user, and the corresponding records are read. Information is formatted for maximum communication and displayed. "I've met with various users," he tells Anna. "Here's a list of inquiry programs." Each of the inquiry displays is designed, along with the database tables needed and possible errors that could occur.

"The first display I designed was the HARDWARE INQUIRY," Chip continues. "I used the description of the display that we had put into the Visible Analyst repository after the prototypes had been created." The Notes area contains information about how it should operate. An INVENTORY NUMBER or partial INVENTORY

**FIGURE E14.3**

A dialog box for the computer system report menu.

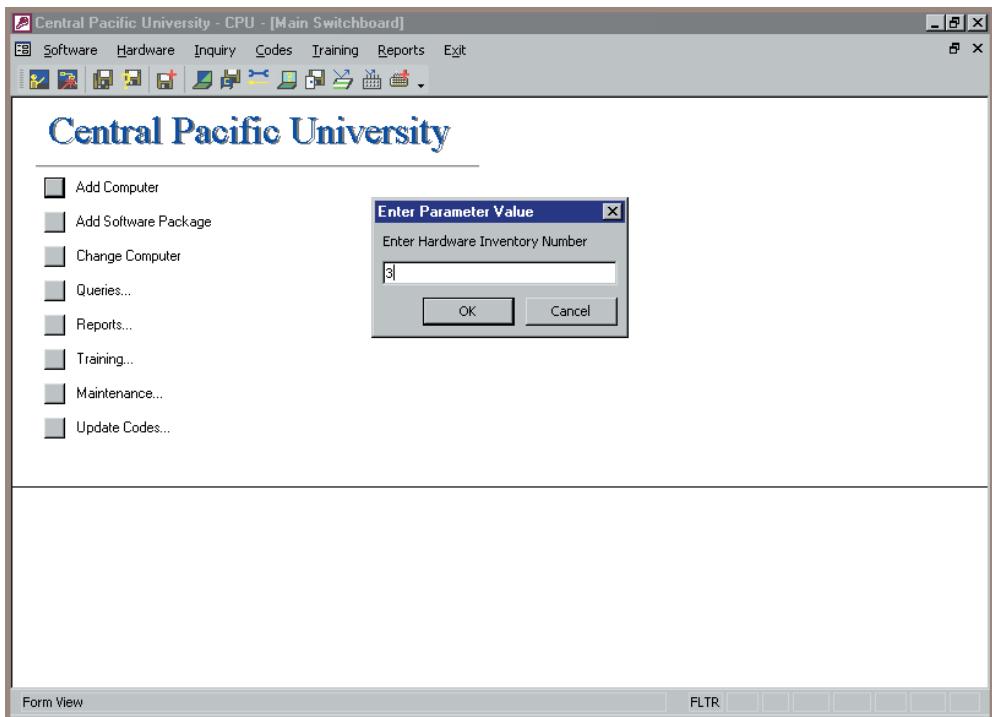
NUMBER should be entered. The first matching record is read (in the case of a partial INVENTORY NUMBER), and the user can scroll to the next record or to previous records.

"I produced a rough layout and met with Dot to obtain feedback on the design," Chip says. "After pointing out some minor corrections, she mentioned that the maintenance details should be included, providing complete information for each computer."

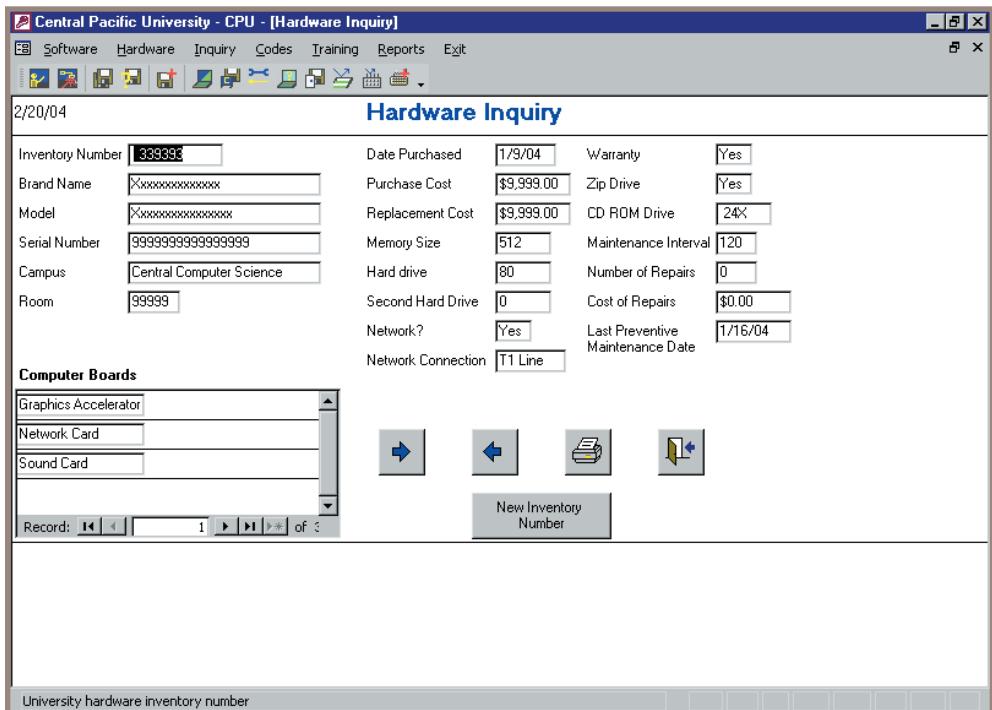
The program logic is to use the HARDWARE INVENTORY NUMBER as the entry field of a **Parameter Value** dialog box (illustrated in Figure E14.4), with the starting number of 3 entered. The record is searched for in the database. If it is not found, a message is displayed. Once the record is located, the matching board records are read. Board records contain a code for the type of board, and the BOARD CODE TABLE is searched for the matching code. The meaning of the code is formatted on the screen. The resulting display is shown in Figure E14.5. Notice that there are buttons for printing the current record on the form and closing the form. The **New Inventory Number** button redisplays the **Parameter Entry** dialog box, and it allows the user to choose a new record.

"I selected the SOFTWARE LOCATION inquiry as the next display to develop," Chip tells Anna. "After talking at length with Cher, I produced the details and documented them in the Visible Analyst repository. The entry field is a partial software TITLE, entered in a Parameter Value dialog box. The first record matching the partial TITLE is displayed, and, because there are different operating systems and versions of

14

**FIGURE E14.4**

HARDWARE INVENTORY NUMBER parameter value dialog box.

**FIGURE E14.5**

HARDWARE INVENTORY NUMBER inquiry screen as represented in Microsoft Access.

the software, the user can click buttons to advance to the next (or previous) record. Five columns of information are displayed: HARDWARE INVENTORY NUMBER, BRAND NAME, MODEL, CAMPUS, and ROOM. Cher can quickly locate a machine containing the desired software. She seems to be happy with this idea so far," Chip adds.

The program locates the SOFTWARE MASTER record using the alternate key TITLE. If the matching record is not found, an error message is displayed. Because there may be several versions, the Next Record button may be clicked until the correct OPERATING SYSTEM and VERSION NUMBER are obtained.

Once the correct software has been obtained, the relational file is used to find the matching SOFTWARE INVENTORY NUMBER. This relational file contains the SOFTWARE INVENTORY NUMBER and the matching HARDWARE INVENTORY NUMBER, which is used to locate the matching record in the COMPUTER MASTER. For each matching machine, the CAMPUS table is used to locate the CAMPUS LOCATION code and to display the matching CAMPUS DESCRIPTION. The area for displaying the machines containing the software is a scroll region, because it may contain more machines than will fit on a single screen.

"I think we've got a good start on designing our user interfaces," Anna comments. Chip nods in agreement.

EXERCISES

-  E-1. Use Microsoft Access to view the menu options for the computer system.
- E-2. Examine the HARDWARE INQUIRY. Explain the inquiry type using the value, entity, and attribute (V, E, A) notation.
- E-3. In a paragraph, explain why a data entry display should emphasize accuracy, whereas an inquiry display emphasizes how fast results may be displayed.
- E-4. Modify and print the hierarchy chart representing the **Update Hardware** menu.

Add rectangles to represent the following menu options:

CHANGE COMPUTER
DELETE COMPUTER RECORD
UPDATE INSTALLED COMPUTER

-  E-5. Use the **Functional Decomposition** feature of Visible Analyst to draw a hierarchy chart representing the options found on the **Update Software** menu. Start with the top rectangle representing the **Update Software** menu.

ADD SOFTWARE PACKAGE
CHANGE SOFTWARE RECORD
DELETE SOFTWARE RECORD
UPGRADE SOFTWARE PACKAGE

- E-6. Chip and Anna realize that the menu that has been designed is for the users involved in the installation and maintenance of computer hardware and software. This menu would not be suitable for general faculty and staff members, because they should not have the ability to update the records. Design a menu, either on paper or using software with which you are familiar, that would provide the general user with the ability to perform inquiries and reports.

 The exercises preceded by a Web icon indicate value-added material is available from the Web site at www.prenhall.com/kendall. Students can download a sample Visible Analyst Project and a Microsoft Access database that can be used to complete the exercises. Visible Analyst software can be packaged with this text for an additional fee.

14

- E-7. Discuss in a paragraph why the users would need to move to another page (by pressing the **Next Record** button) to display the correct record for the SOFTWARE LOCATION inquiry.
-  E-8. Design the SOFTWARE DETAILS inquiry display. The entry field is SOFTWARE INVENTORY NUMBER, and all software information, with the exception of EXPERT and MACHINES INSTALLED ON, should be displayed. Refer to the Visible Analyst SOFTWARE DETAILS data flow repository entry.
-  E-9. When scheduling classrooms for student use, Cher Ware needs to know all the software packages in a given room. She would like to enter the CAMPUS LOCATION and the ROOM on an inquiry display. The fields would be TITLE, VERSION, SITE LICENSE, and NUMBER OF COPIES.
Design the SOFTWARE BY ROOM inquiry, which is described as a data flow in the Visible Analyst repository.
-  E-10. Mike Crowe needs to know which component boards are installed in each machine. Use Visible Analyst to view the data flow entry for COMPONENT BOARD and to design the COMPONENT BOARD inquiry. The input field is the HARDWARE INVENTORY NUMBER. Output fields are BRAND NAME, MODEL, and a scroll region for BOARD. The logic is to read the COMPUTER MASTER using the HARDWARE INVENTORY NUMBER. If the record is not found, display an error message to that effect. Find the matching BOARD records. Use value, entity, and attribute (*V, E, A*) notation for the type of inquiry.
-  E-11. Every so often, Hy Perteks receives a request for help concerning a given software package. Staff members and students need to perform advanced options or transfer data to and from different packages, and they are having difficulties. Hy would like to enter the software TITLE and VERSION NUMBER. The resulting display would show the SOFTWARE EXPERT NAME and his or her CAMPUS LOCATION and ROOM NUMBER. Design the screen for the LOCATE SOFTWARE EXPERT inquiry. Describe the logic and files needed to produce the inquiry. Use value, entity, and attribute (*V, E, A*) notation for this inquiry. The details for this inquiry are included in the Visible Analyst SOFTWARE EXPERT data flow repository entry.
-  E-12. In a follow-up interview with Cher Ware, it was determined that she needs to know what machines are available to install any software package, given the package's graphics requirements. Produce an inquiry that would allow Cher to enter the DISPLAY CODE and, optionally, a GRAPHICS BOARD and CAMPUS LOCATION for the software. Four columns should be displayed:

HARDWARE INVENTORY NUMBER
 CAMPUS LOCATION
 ROOM LOCATION
 GRAPHICS BOARD

Refer to the Visible Analyst MONITOR REQUIRED data flow. Write a paragraph describing the logic involved in obtaining the results. Include the type of inquiry using value, entity, and attribute (*V, E, A*) notation.

-  E-13. Both Cher and Hy expressed an interest in finding machines of a specified brand connected to different printers. Sometimes the engineering students need a plotter, whereas other situations demand a color laser or portable printer.

Design an inquiry that would have the PRINTER and BRAND NAME of the computer as input fields. Output would be two columns: CAMPUS LOCATION (full name, not a code) and ROOM LOCATION. Refer to the Visible Analyst PRINTER LOCATION data flow.

Briefly describe the logic used in producing the output. Would this inquiry need a scroll region to display all the information? Why or why not? Use a paragraph to describe the type of inquiry using value, entity, and attribute (V, E, A) notation.

- www E-14. Hy receives a number of requests for training classes. He would like to plan training and place the upcoming classes on the intranet so that faculty would have an adequate amount of lead time to schedule a class. Design the SOFTWARE TRAINING CLASSES inquiry. The details may be found in the Visible Analyst data flow repository entry called SOFTWARE TRAINING CLASSES.

