GUI

GUI Design Essentials

Chapter 7

Designing for People

Contents

A technology solution is really made up of two systems: the technology system (for example the computer and software) and the human system (the people who use it). When you apply software engineering to designing software you are designing for the technology system. But when do you design for the human half? Making sure your technology solutions take both technology and the human element into account creates a superior system overall.

Humans have strengths and weaknesses. In designing an interface you want to play up the human's strengths, and design to accommodate or minimize human limits. One way that human factors specialists design for people is to take visual, cognitive, and physical considerations into account (Figure 7.1).

The guidelines in this part of the book stem from knowledge accumulated in the last 50 years about people. Specifically, we draw upon our more than 16 years of direct work in the field. Here is an overview

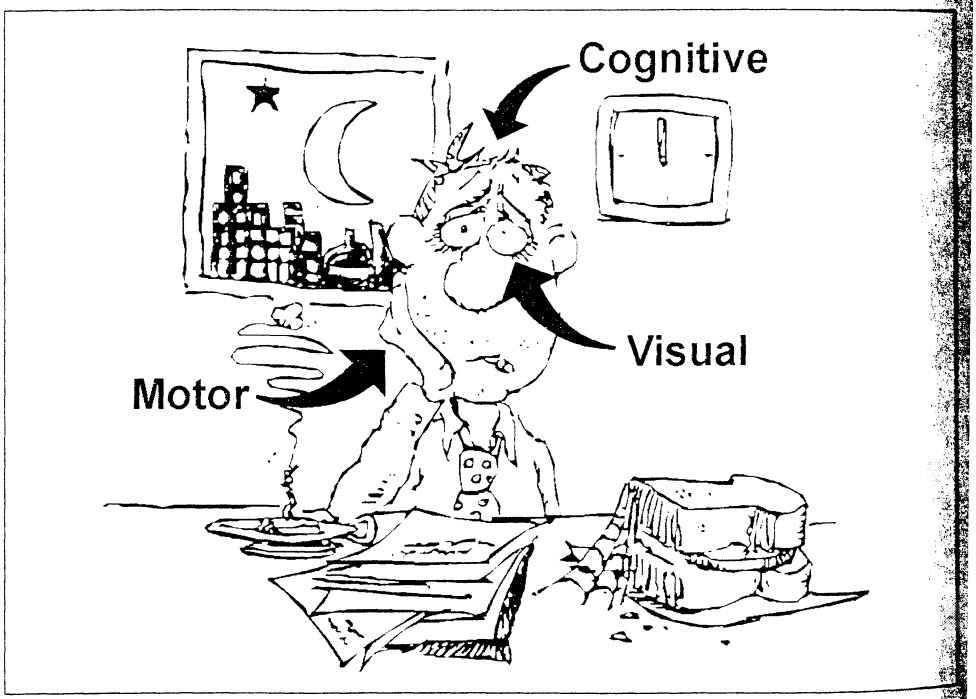

Figure 7.1 Humans have cognitive, visual, and motor limits.

of the guiding principles that form the basis for the interface design guidelines.

Cognitive Considerations

Some of the most important considerations when designing interfaces are the ones involving how people think and learn (cognition). Below are some guiding principles for cognitive processing.

Limit memory loads

People can remember approximately seven new things for about 20 seconds. This is called short-term memory. After 20 seconds they will have lost the information if they cannot quickly store it in long-term memory. Do not require people to remember information longer than this. For example, instead of making a user remember the name of the current customer, show them the name as the user moves from window to window.

One of the powerful aspects of GUIs is that they contain built-in memory aids, for instance, allowing people to choose from a list in a list box rather that typing in data from memory in a data entry box. Make sure you make full use of these opportunities when designing a GUI interface.

Break down decision-making

If users have to make decisions in order to navigate through a task in your software, break down the decision steps into manageable chunks. Have them choose how to sort, for example, and then choose the kind of filtering through a series of controls in a dialog box so that they can think of each decision separately. Use group boxes and labels to help distinguish which decision they should be making at a given point in time.

One of the reasons wizards can be so powerful is that they break down the decision-making to very small steps. This is also one of the

reasons they can be annoying—if users don't want such small
Be conscious of how small a step your user group needs, and
appropriately.

Provide context

In order for people to be able to understand and remember what is
ing communicated to them, they need context. Context provides
cific meaning and interpretation. If you present information out
context, the material will take longer to be understood and may
be remembered.

Some ways of providing context in a user interface are:

- Using titles on screens and windows
- Using labels on screen titles, buttons, menus, and group frames that are easily understood (not cryptic abbreviations)
- Using menus to show what is available

Be aware of user's mental model

As soon as users begin working with a system, go through training, read documentation, or talk to other people about the system, they begin to create a mental picture or model of how the system works (Figure 7.2). This is a rapid, and not entirely conscious, process. They then use this mental model to predict how the system will behave and to decide what they should do next.

If a mental model is difficult to create, because the interface is confusing and unpredictable, or the conceptual model it uses is a poor match to how users think about their work, then users will find it hard to learn the system, will make more errors, and will be less satisfied with the system.

To help users learn and use the system, you need to design with the mental model in mind. Part I of this book, especially the section on Interface Design, is concerned mostly with creating a match between the mental model of the user and the conceptual model of the interface.

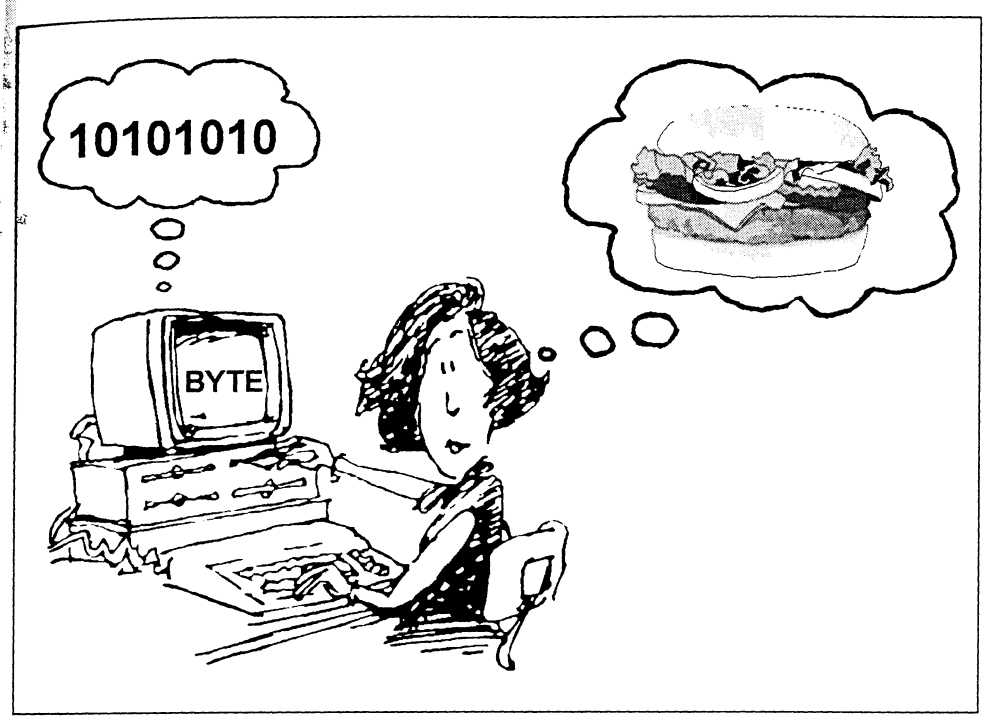

Figure 7.2 Users create mental models.

Be consistent

One of the ways to facilitate a good mental model quickly is through consistency. People rely on consistency in order to find information quickly, create an accurate mental model, and make decisions. The push to consistency is one reason that platform- and enterprise-specific guidelines are becoming common and important.

Be forgiving

People like to explore and try things out. They will be pushing buttons on the mouse, opening windows, and pressing keys. Make sure your interface allows them to explore without doing damage. Build in ways for users to cancel out, go back, and undo actions.

Visual Considerations

There has been a lot of research on how people scan, read, an information on screens. If you can reduce the amount of work to visually use a screen, you can save users time and ensur they'll see critical data. Some overall principles we use in deve guidelines in this area are described below.

Minimize eye movement

Design your screens and windows so that users can start at th and work their way down without having to move their eyes back forth or up and down a lot.

Adhere to principles of good format and layout

Research in the field of human factors and screen design has resu in sound advice on information placement, grouping, and orde formation should be placed to follow the pattern of reading, for e ple, people who read English will tend to look at the top left of screen, and then move both left to right and top to bottom placement of information on the screen leads them towards either left to right (horizontal) or top to bottom (vertical) first. Hence will see guidelines in the chapters below that say that either a h zontal or a vertical flow is acceptable, but to make sure it is c which one you are using.

Use color and highlighting judiciously

Don't abuse or overuse color and highlighting. Techniques such underlining, using boxes, or color can be very powerful ways to v ally grab attention. But because they are so powerful, you must sure you are using these techniques effectively and not overus them. Every time you use color or highlighting you should be able

state the reason for the particular use. If you cannot find a reason then maybe you shouldn't be using that technique.

Use visual coding

Use visual coding, such as graying out unavailable options to provide visual meaning to data on specific areas on a screen.

Don't assume people will read everything on a screen

If people see things over and over again they *don't* see them anymore—this is called "gating" information. This is a useful function of our brains so that we will notice what is different. But it presents possible problems for the interface designer. You can't assume that people will see or read something just because you put it somewhere on the screen.

ysical Considerations

Sometimes we forget that we require users to interact physically when we design software, for example, manipulating the keyboard, a mouse, touch screen, and so on. Below are some of the critical physical demands to watch out for.

Limit key combinations

Limit the number of combination key presses you require, for example, "CTRL + V." Not only are they hard to remember, they are harder to physically act on.

Avoid difficult combinations

Avoid key combinations that are particularly difficult, for example, requiring users to press three keys simultaneously, or two-key combinations that use only one hand.

Pay attention to touch-typing skills

If users are touch typists they will prefer to keep their hands on the home row. Don't require them to take their hands off this row. If your users are not touch typists, then try to use numbers or function keys rather than letters on the keyboard. People who are not touch typists may have a hard time locating a specific letter.

Avoid a 50/50 split

Avoid forcing users to use a keyboard 50% of the time and a mouse, pointing device, or touch screen 50% of the time, with frequent switching. Although you can use options and choices, you should design for 80% use one way or the other for a particular task.

Watch out for repetitive motion syndrome

Wrist and finger problems resulting from improper use of a keyboard or a mouse are real and significant. If users will be typing or using a pointing device for long periods (greater than one hour), make sure someone will take care of proper support, such as wrist rests, and rest breaks.

Make sure users get training on devices

Don't assume everyone knows how to use a trackball or mouse. Make sure someone is handling training, especially if the device is not standard, such as the pointing device for notebook computers.

Environmental and Social Issues

Don Norman from Apple talks and writes extensively on the ways that computers and technology fit in or don't fit in the social aspects of people's lives. These social considerations are probably some of the

most powerful ones in software design, and the most overlooked. In his book *Turn Signals are the Facial Expressions of Automobiles*, Norman comments that people have come to expect that computers will follow the same rules of communication that exist for human to human interaction. In other words, people expect a relationship with computers and with their software. They expect your interface to communicate with them, be flexible in working with them, give them feedback, and so forth. These expectations only increase as interfaces improve.

Many years ago, people were impressed and satisfied if a computer could crunch numbers and produce the correct answer. Now people expect their computers to respond to them conversationally, to be predictable and reliable, and to be friendly and intelligent. This places an even greater amount of work and careful design onto the shoulders of the people who design and create software.

In addition, we need to consider the social, psychological, and physical environment that people are in when they are using our software. Following are some ideas to think about.

Gauge emotions

People react to everything with emotion (Figure 7.3). They decide that they "like" or "trust" your interface, even if they can't tell you exactly what that means. You need to take these emotions into account when you design. Users may have had a negative or positive experience with a previous system and tend to generalize those emotions to your new system. Knowing about likely emotional responses ahead of time will help you plan for them and deal with them up front.

Build in predictability

Although people may want part of their lives to be spontaneous, they want their computers to be very predictable. If they are trying to get work done they will want to be able to predict how the computer will respond to each of their actions.

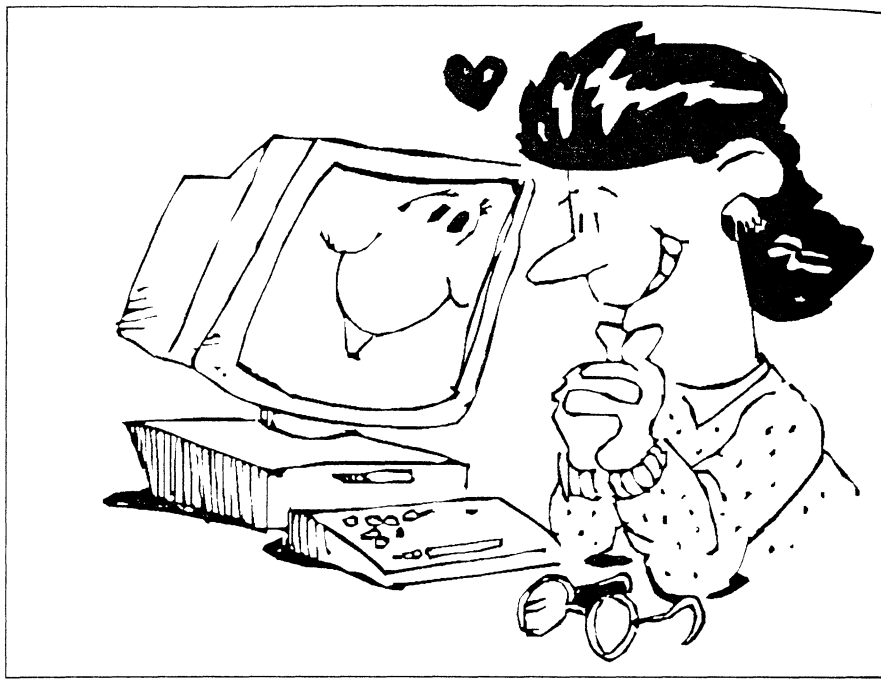

Figure 7.3 People get emotional about their system.

Watch out for stressful environments

People react to all stimuli differently if they are under stress (Fig 7.4). A screen that may be fairly easy to use can become difficult if person has someone on the phone yelling at them. Analyze amount of stress that will be in the environment when people are ing your interface. Then analyze the amount of information and type of activities your interface requires. If users are under stress may need to make design decisions (such as amount of informat on the screen or amount of navigation back and forth) to compen for the confusing effect the stress will add.

In 1989 an American Navy ship accidentally shot down a com cial Iranian jet. The Navy people on the ship mistakenly identified jet as a military plane. Shortly after this incident the then head o Navy said on the radio that all the information the Navy crew ne to make a correct decision was on the computer terminal screen on

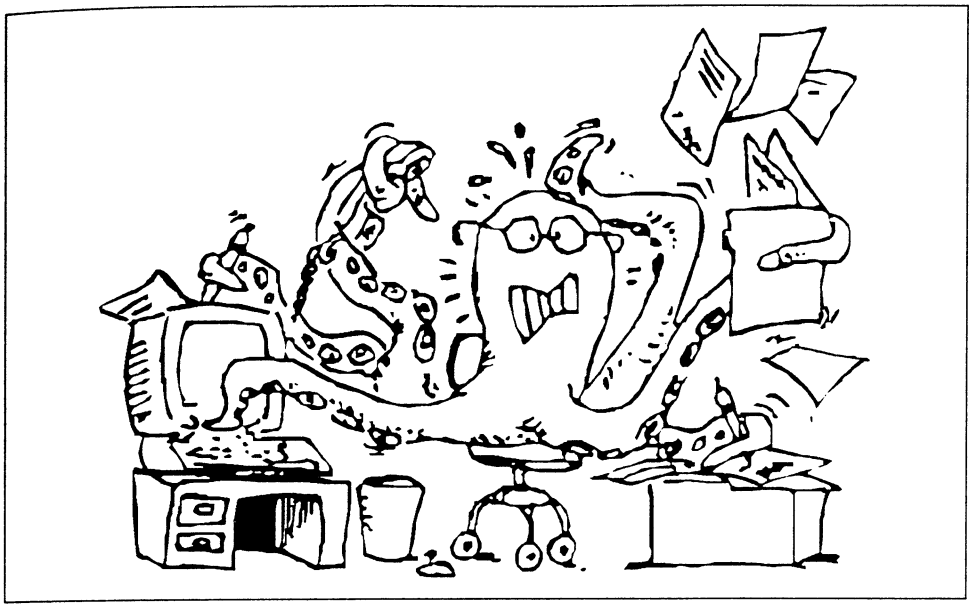

Figure 7.4 Stress affects how people react to stimuli.

ship. But in the stress of the situation they could not find the critical information. "We need human factors engineering on our computer screens," he said. This is an example of how a stressful environment can affect use of a screen.

Consider social interaction

If people are using your software while they are also interacting with another person, you have to be aware of the impact your interface has on the social interaction of the situation. For example, a salesperson speaking with a customer does not want to be distracted or have the customer be distracted by the computer. If the interface becomes a distraction to the interaction between the humans, then it has meddled in the social interaction in a negative way (Figure 7.5). Think carefully how introducing a computer into an existing social interaction may affect the interaction as a whole.

Give some thought as to how people are interacting with others as well; for example, is your user on the phone with a customer while us-

Figure 7.5 Computers should not be a distraction in social situations.

ing the software you are designing? If so, the customer will expect a fast response time from the user. The user may get flustered if it takes a long time to navigate through screens to provide an answer. This itself can set up a stressful situation, especially if the customer is impatient or upset.

The Purpose of the Interface

For many years users have viewed software as a tool to get their work done. And for many years designers have tried to design good tools. Thinking of software as a tool is still important in many of today's interfaces. Most of Part II of this book deals with guidelines for designing a good tool.

Over the last several years, however, there has been an increase in the variety of purposes people have in using computers. Users may not only want to get their work done, but also want to search, brow

buy, learn, and be entertained. Below are some design guidelines to follow when trying to match the user's purpose.

As a tool

If your users are using your software as a tool to get a much larger job done, they want a useful, reliable, and nondistracting tool. As the designer, you'll tend to see the interface and the software as an integrated, important world. You need to shift your mindset so you can view your interface as a tool, and ask whether it is being useful as a tool. When designing a tool interface, remember to:

- *Reduce the demands on the user.* Design the screens, windows, flow, and navigation to minimize the number of decisions, amount of searching, amount of thinking and remembering the user has to do. When the purpose of an interface is a tool to get work done, simple is better, boring is good.
- *Match the user's work flow.* Make sure you know how the user is going to do the work when the new software is in place, and design for that work flow. Don't make them change optimal flow just to fit your order of screens.
- *Follow standards.* Minimize the learning time by incorporating industry-wide and enterprise standards for screens, menus, and controls.

Searching

Sometimes the user's main purpose at a given moment is to search for information. This might be searching for a particular file or report, searching for data, or searching for information through online catalogs. When users are searching they need to be able to change their minds frequently, go back and forth between detail and high level, and change strategies. If the user's main purpose for an interface is to search for information:

- *Plan for flexibility.* The user may want to search narrowly one time and broadly the next. They may want to start narrow and go broad or vice versa. Different users will have different pre-

ferred searching strategies. Users may want to save their own preferences to reuse.

- *Build in mechanisms to "go back."* Users may try a search and not like the result. Make it easy for them to go back and retry.
- *Plan for switching purposes.* After they search what will users want to do? Keep in mind that searching may be a temporary purpose. Users might be switching from search to another purpose. Does your interface allow them to quickly change gears?
- *Recognize human limits.* Don't overwhelm users with hundreds of items to search through. Provide filters to let them narrow further. Don't assume that they remember what they have searched on. Show them not only results but a summary of the criteria.

Browsing

Sometimes the user's main purpose at a given moment is to browse. This is a "window shopping" version of using a computer. They might be browsing through a tutorial, or browsing an online document, or browsing sites on the Internet. When users are browsing they want to wander at whim, switch direction quickly, or sometimes stay and linger for awhile. If the user's main purpose in using the interface is to wander through information:

- *Provide different layers of structure.* Provide high-level summaries that can be quickly scanned so that the user can decide if they want to browse deeper or keep moving.
- *Separate your purpose from theirs.* You may want them to stop and search, and stay and buy, but they may want to only pass through. Be respectful of their desire to get out quickly or they will just become annoyed and will not come back.
- *Grab their attention.* Use methods such as color, graphics, and corporate identity to grab their attention and draw them to what is critical. Don't overuse these techniques or they will become saturated and not pay attention to any one thing.
- *Make navigation easy.* Make sure that navigating through, in and out of your interface is easy to learn and use. You want

them to concentrate on content and not get frustrated over how to get in, out, and around.

Buying

When the user's purpose is to buy or place an order, for instance ordering a product from an online catalog, they are very focused. They may need to search or browse first, but you need to make sure they can easily complete the order. They expect the process to be as quick and easy as paying for an item at a checkout counter. If the user's purpose is to actually buy or place an order:

- *Allow them to switch purposes quickly.* Once the user has decided to make a purchase you need to stop selling. They are not browsing or searching anymore. They want a tool that makes it easy to place an order. After they order an item, make sure it is easy for them to switch back to browsing or searching in case they want to buy more.
- *Build in a quick and easy ordering process.* Make sure that it is very fast and easy to place an order. Do not require them to move back and forth between screens. When they are ready to order they should have all the information they need to complete the order, unless they change their minds.
- *Make it easy to change buying decisions.* Make sure it is easy for them to change their decision, for example, adding an item, or changing the quantity. Don't make them cancel the current order and start all over if they change their mind.
- *Follow "tool" guidelines.* Realize that once they start to place an order, they are actually using the computer as a tool. All of the comments above on designing a tool apply to good order forms.

Learning

Learning is a different experience than either a tool, browsing, searching, or buying. If users are using the computer as a learning tool, for instance taking a computer-based tutorial, they will need to have their attention caught and retained. This can be difficult if there are other

distractions in the environment, such as telephones or other interruptions. If the user's purpose is to learn:

- *Keep their interest.* Use graphics, color, and animation to make the information interesting and keep them from getting distracted by their environment.
- *Use instructional design principles.* Learn about and use principles of instructional design to chunk information into meaningful bits. Don't overwhelm them with too much at once.
- *Minimize teacher talk.* Use boxes with text sparingly. People do not read large blocks of text on computer screens.
- *Build with easy in's and out's.* Make it easy for users to stop where they are and come back in later at the same place. Make it easy for them to start a small section over again without going back to the beginning.

Entertaining

Sometimes users are using a computer to be entertained, for example, playing a game. If the user's purpose is to be entertained:

- *Make the interface challenging.* Almost the opposite of a good tool, an interface for entertaining should be challenging and even difficult. The user needs to feel they always have something new to do, or a new challenge to overcome.
- *Watch out for assumptions.* Your idea of what is entertaining might be boring to someone else. Be specific in your assumptions and test them.
- *Pull out all the stops.* This is the time to use color, graphics, animation, sound, video, and so on.

About Guidelines

Part I of this book detailed the process of gathering and analyzing data, and creating usable designs. There comes a time in the design of an interface, however, that you must make concrete decisions about a particular screen, for instance, where to place a particular button, and what to name it. How can you ensure that you are making those cor-

rect detailed decisions? How can you ensure that all the developers on a project or across the enterprise are making correct decisions? The is the role of Part II of this book.

When standards and guidelines are followed during the design process, benefits for users include:

- *Reduced user's work time.* When applications are consistent in look and feel both within an application and from one application to another, users won't be distracted by the interface and will be able to get on with their work.
- *Reduced training time.* Users do not need to take time to learn the look and feel of each application.
- *Users become more involved in application design.* When guidelines are in place, and users know what to expect for the look and feel of the interface, they are better able to concentrate on the specific interface to get their job done, and give the most useful feedback to the design team.

When guidelines are used for design, benefits for developers include:

- *Reduced decision time.* If developers do not have to make decisions for each part of the interface, it frees them up to concentrate on one or more of the many other technical decisions they need to make.
- *Sound basis for decisions regarding users.* Rather than having to rely on the rationale "I just think it would be better," they can have the weight, research, and opinion of others with them through corporate guidelines.
- *Reduced programming time.* When guidelines include examples and templates, they can save time in programming, since developers are able to open and modify an existing template as needed.

Finally, two of the most overlooked benefits of using guidelines—items that benefit the entire enterprise—are the issues brought to the forefront, and the discussions that result from the work of multifunctional, multidisciplinary teams. The guidelines process gets players from different teams talking—talking not only about guidelines, but about projects, tools, challenges, and solutions.