

CHAPTER 9

Minimizing Cognitive Load

Cognition is the processing of information from the world around us. It includes perception, attention, pattern matching, memory, language processing, decision making, and problem solving. **Cognitive load** is the amount of mental resources needed to perform a given task.

All user interfaces make cognitive demands on users. Users must master special rules of system use, learn new concepts, and retain information in short-term memory. They must create and refine a mental model of how the system works and how they should use it. Systems that use purely auditory interfaces further challenge human memory and attention because they present information serially and non-persistently.

Successful user interface designs must respect the limitations of human cognitive processing. If a design requires the user to hold too many items in short-term memory or to learn a complex set of commands too quickly, it will fail. This chapter describes a number of guidelines for minimizing the cognitive load on users of spoken language interfaces.

There are three cognitive challenges you should consider as your design progresses:

1. **Conceptual complexity:** How complex are the new concepts callers must learn? How well do new mental structures match concepts and procedures that users are already familiar with?
2. **Memory load:** How much information must callers hold in their short-term memory? How much new material (e.g., commands, procedures) must they learn?
3. **Attention:** Is it easy for the caller to attend to the most salient information? Will callers' attention be divided? If they are momentarily distracted (e.g., while driving), can they seamlessly continue their interaction with the system when they are ready?

The following sections discuss each of these potential challenges and present guidelines for handling them.

9.1 Conceptual Complexity

Conceptual complexity is, in part, a product of the new concepts the user must learn and the inherent complexity of those concepts. However, understanding the cognitive challenges goes beyond counting concepts and measuring their complexity. It is also a matter of understanding human capabilities (what is hard and what is easy for humans to understand and learn) and understanding the context in which users will operate (e.g., how the current application will mesh with existing user knowledge, skills, expectations, and mental models).

In this book we do not try to present a theoretical framework that allows precise prediction of the difficulty of particular design decisions. The necessary knowledge to create such a theory is incomplete. Instead, we present a set of guidelines that will help you minimize the cognitive challenges for your callers. This section covers the following guidelines:

- **Establish constancy:** Create **constants**, or universal commands, that are always available to the caller regardless of context. Universal commands should be designed to let callers recover from problems or receive help using the system.
- **Ensure consistency:** Do similar tasks in similar ways throughout the application. For example, whenever a caller traverses different kinds of lists in a large portal application, make the same set of list traversal commands available. In this way, you minimize the amount of new material the caller must learn.

- **Set the context:** Set the context for your callers. Make it clear why the system is taking specific actions. Appeal to their world knowledge and expectations in order to simplify new concepts they must understand.

9.1.1 Constancy

Graphical user interfaces take advantage of the ability to display information, often lots of it at the same time, on the computer screen. For example, many GUIs display a toolbar (see Figure 9-1), usually at the top of the screen. This toolbar, which typically consists of a row of icons representing actions, provides both a visual reminder of popular actions and a physical means to initiate them.

The toolbar is constant: It remains on the screen, and the icons never change. The toolbar's constancy reduces the user's need to memorize a set of actions and commands.

Similarly, VUIs can achieve constancy by using universals: a small set of spoken commands that are always available regardless of context (see section 5.2.2). After callers learn the universal commands, they can use them at any point as well as in future calls. These commands become, in effect, a mental toolbar of always available actions (see Figure 9-2).

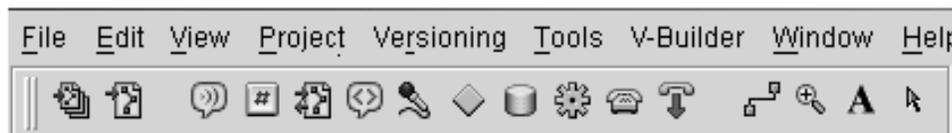


FIGURE 9-1 A GUI toolbar displays unchanging icons.

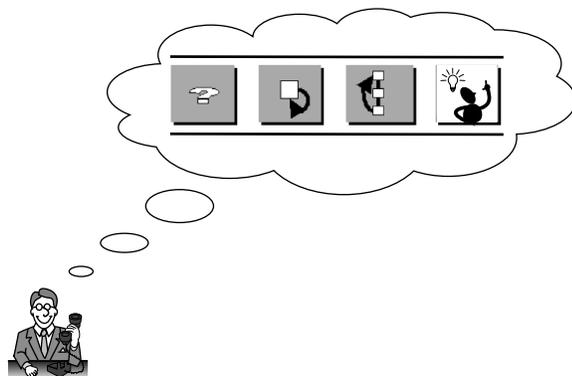


FIGURE 9-2 Callers form a “mental toolbar” of spoken universals.

It is not practical to expect a caller to learn more than a very small number of universal commands. This number may become slightly larger if a standard set of universals is established and widely used throughout the speech industry.

Given that the number of universal commands should be small, it is best to associate the commands with functions that a caller can use to get out of trouble—for example, asking for additional help or instruction, moving to a different part of the application, or requesting to speak to a live agent. Successful use of such universals should improve transaction completion rates, automation rates, and user satisfaction.

You should choose command words or phrases for universals that are intuitive and easy to remember (for example, “help”). The commands should have the same meaning, no matter when they are spoken. For example, saying “Help” should always mean that the caller wants more detailed instruction about what can be done at the current point in the dialog. The instruction the caller hears in response will vary depending on the current context, but the universals should always be available.

Two standards bodies have been looking into universals: the Telephone Speech Standards Committee (TSSC 2000) and the European Telecommunications Standards Institute (ETSI 2002). Both committees have solicited input from the design and development community about its experience with universals. The two groups have conducted experiments to see what terminology occurs most naturally to users when they want to elicit the universal behaviors. Both committees have reached similar tentative conclusions.

The following list shows the set of universals that we recommend for all applications. The word or phrase in brackets is the actual command callers would use. Our choice of universals is based on the results of the two standards committees as well as our own experience with deployments. In the future, if an industrywide standard is accepted, we support conformance. The entire industry, and our callers, will benefit if a consistent set of universals is used for all applications.

- Clarification universals
 - [help]: Provide help or additional instruction about the current dialog state.
 - [repeat]: Repeat the most recently played prompt.
- Navigation universals
 - [main menu/start over]: Return to the beginning of the application (following any login process).
 - [go back]: Back up to the preceding step.

- Termination universals
 - [operator]: Transfer to an operator or customer service agent.
 - [good-bye]: Allow the caller to say “Good-bye” and respond appropriately so that the caller is comfortable hanging up.

The good-bye command is included because analysis of deployment data has shown that callers say it even when they are never told it is an available command. Usability studies have suggested that many callers are more comfortable saying good-bye, rather than just hanging up, especially when transactions are involved. It gives them confidence that hanging up does not prevent their transaction from being completed.

In general, callers must be taught about universal commands, or else they won't use them. One approach is to mention the help command in the initial prompt, which the caller hears when first entering the system. Description of the other universals can be included as the last part of help prompts, as well as part of any error recovery prompts. For example, a banking application might have the following initial prompt:

Welcome to Western Valley Bank. If you ever have a problem while you're using this service, just say, "Help." Now, would you like to pay a bill, check your balance, or transfer money?

If the caller says “Help” while in the middle of an operation for obtaining an account balance, the system response might be the following:

Okay, here's some help. You requested an account balance, but I don't know which account. You can say, "Savings" or "Checking." Anytime you like, you can also say "Main menu" or "Operator."

9.1.2 Consistency

You can also reduce callers' cognitive load by giving careful attention to consistency. When you design dialogs, the idea is to let the caller do similar things in similar ways. For example, there may be a number of lists that can be traversed in the course of an application (e.g., the stocks in the caller's portfolio, outstanding buy or sell orders, the caller's stock watch list). Using the same traversal strategy lets callers carry over what they learn from traversing one list to the traversal of all other lists.

Consistency can be applied at many levels. Dialog strategies should be consistent, as should terminology. Grammar coverage also should be consistent; for example, if you use money-amount grammars in two places, don't allow “fifty bucks” in one place but not the other.

In general, callers learn from every interaction. Each exchange between callers and the system refines callers' mental model of the system. That model includes their view of the system's capabilities, the commands available to perform various tasks, the richness and flexibility of grammars, and so on. Being consistent helps make the caller's learning process more efficient and effective. Failing to be consistent contributes to learning inefficiency, leads to confusion, raises the out-of-grammar rate, and provides a much less satisfactory experience for callers.

Let's consider an example of list traversal. In the following dialog with a brokerage system, two types of lists are traversed: the list of companies in the caller's watch list and the list of outstanding orders for buys and sells. Notice that the caller uses the same set of terms for navigating each list.

**(1)**

SYSTEM: What would you like to do?

CALLER: Play my watch list.

SYSTEM: International Business Machines, eighty two, up one point three six. Amazon, thirty six, down . . .

CALLER: Next.

SYSTEM: Ford Motor Company, ten point . . .

CALLER: Next.

SYSTEM: Charles Schwab Corporation, eleven . . .

CALLER: Finished.

SYSTEM: What would you like to do next?

CALLER: Tell me my open orders.

SYSTEM: You have five open orders. Here's the first one: sell one hundred shares of Microsoft . . .

CALLER: Next.

SYSTEM: Buy two hundred shares of Cisco Systems at seventeen . . .

CALLER: Next.

SYSTEM: Buy five hundred shares of International Business Machines at eighty two . . .

CALLER: Finished.

SYSTEM: What would you like to do next?

9.1.3 Context Setting

Context setting is another important concept that can be applied to the design of voice user interfaces to minimize cognitive load (Weinschenk and Barker 2000). Psychological research has shown that people can understand and remember infor-

mation more easily when it is presented with the appropriate context. For example, consider the following passage (Bransford and Johnson 1973):

The procedure is actually quite simple. First you arrange things into different groups. Of course, one pile may be sufficient depending on how much there is to do. If you have to go somewhere else due to lack of facilities, that is the next step; otherwise you are pretty well set.

When asked to remember as many ideas in this passage as possible, subjects remembered only about three. However, when told beforehand that the passage is about laundry, they recalled twice as many ideas. Context helps people to associate new information with familiar concepts, easing cognitive load.

One way to set the context in a user interface is to use a metaphor. As discussed in Chapter 4, a metaphor is a familiar object or schema that is used to help facilitate understanding in another domain. You may recall the examples of the desktop metaphor and the shopping cart metaphor.

To investigate whether metaphors actually help users of voice interfaces, researchers at British Telecom conducted a study that compared three shopping applications equipped with a voice interface (Dutton, Foster, and Jack 1999). One system had no metaphor and simply described merchandise through a menu structure. Another had a department store metaphor in which the callers used a virtual elevator (with sound effects) to move from floor to floor. The system described the merchandise on display for that floor. A third system had a catalog magazine metaphor in which the system described merchandise presented in magazine pictures.

Callers rated the system with the department store metaphor significantly higher than the system with no metaphor. (The system with the magazine metaphor scored between these two.) In addition, callers were better able to navigate the system when a metaphor was present. These findings indicate that context setting through the use of a metaphor can improve user satisfaction and effectiveness.

9.2 Memory Load

Callers cannot process large amounts of new information at one time and will not remember new information if it is not immediately useful to them. There are a number of techniques for creating menus, wording prompts, and providing instruction that help minimize the load on a caller's memory.

9.2.1 Menu Size

In an influential article titled “The Magical Number Seven, Plus or Minus Two,” Miller (1956) described a pattern of human short-term memory in which people can store seven, plus or minus two, items. Often, designers use this as a guideline for the number of items to put in lists, menus, and so on. However, listening to sentences over the phone while trying to extract and remember information from these sentences is much more taxing than the tasks Miller used in the lab. The caller’s task is more akin to the listening task in which subjects are asked to listen to a series of sentences and remember the last word of each sentence (Daneman and Carpenter 1980). In experiments using this completely auditory approach in which sentence comprehension is also taking place, people can remember only about three items on average.

Other research on human memory has shown that people naturally cluster items in threes and that recall is best when information is divided into groups of three or four items (Broadbent 1975; Wickelgren 1964). Taken together, the research results suggest that the caller’s memory load should be kept quite small. A reasonable guideline is to limit menus to three or four items. Both Gardner-Bonneau (1992) and Schumacher, Hardzinski, and Schwarz (1995) make the same recommendation of four or fewer items in a menu.

9.2.2 Recency

When you write prompts, if callers are told specific words to use in response, make that the last thing they hear. For example, “To hear the list again, say, ‘Repeat list’” is better than, “Say, ‘Repeat list’ to hear the list again.” It taxes callers’ memory less if the item they need to remember is the last thing they hear. This effect is often referred to as the **recency effect**. This ordering of *function* and then *action* in prompt wording has been adopted from standards of touchtone systems (Balentine 1999).

In addition to the recency effect, there are linguistic reasons to structure spoken sentences with the information of importance at the end. These are covered in Chapter 10.

9.2.3 Instruction

Applications that have many features and capabilities, especially those that will be used repeatedly by the same callers, often include instructional modes as part of the

interactive application. Instruction sheets sent through the mail, or reminder cards with lists of commands, are not very effective. Most users do not read the instructions before attempting to use the system. Thus, applications need to be self-explanatory. It should be possible for an inexperienced user to get all needed help while using the service. We discuss two approaches here.

Tutorials

Some systems offer online tutorials, demos, or a combination of both. The option of hearing a tutorial is typically offered the first time a user calls the system. This approach is used mainly in subscription services or services that expect a lot of repeat use (for example, personal agents, banking, or brokerage account access). A **tutorial** consists of step-by-step instructions for using certain features of the system. A **demo** consists of a recorded interaction between a simulated caller and the system. The system voice speaks the prompts that are played during real system use.

Tutorials that present a demo of a user interacting with the system (CCIR-4 1999) and interactive tutorials (Kamm, Litman, and Walker 1998) have been shown to be effective instructional tools for new users. However, if too much information is provided or if the information is only described (rather than presented in an interactive mode), users have a hard time digesting it (Balogh, LeDuc, and Cohen 2001). There are two rules of thumb about tutorials:

1. Teach only a very small number of concepts.
2. Make it interactive. Have the caller actually perform the action.

Just-in-Time Instruction

When you need to describe a large amount of functionality, there are drawbacks to relying solely on a tutorial or demo. First, a caller will have difficulty following a lengthy description of diverse functionality. Second, if a feature is not exercised immediately, it is likely to be forgotten. In general, callers will not have the patience to listen to a lengthy description, especially if it is not helping meet their immediate needs.

The notion of **just-in-time instruction** can address these two limitations of tutorials (Cohen 2000). The basic idea is to provide only the instruction relevant to the task immediately at hand, just before the caller needs to perform the task. The

amount of new information introduced at that point is small, and it is exercised immediately.

For example, consider a personal agent application with a rich set of capabilities. Rather than hear a detailed tutorial the first time they use the system, callers are instructed about particular capabilities the first time they access them. For example, the first time callers ask for traffic information they might get the following message:

You can get up-to-the-minute traffic reports for the major roadways in any city simply by saying the city name. You can also save time by saying the name of the road or highway *and* the city name. For example, you can say, "Highway 101 in San Francisco."

Just-in-time instruction can be offered the first time a caller exercises a particular capability. Additionally, you can offer instruction if a caller is having problems using the system, such as frequent rejects, timeouts, misrecognitions, and so on. If the caller is not making optimal use of system features (for example, not using shortcuts or not taking advantage of the richness of a grammar to efficiently input data), you can offer instruction.

The next two examples compare two approaches to instruction. In (2), first-time callers are offered a tutorial. The tutorial is lengthy and covers a large amount of information. It is doubtful that a caller will retain very much of what is covered, especially in future calls to the system. In (3), the application uses just-in-time instruction. The same material is covered (this example shows only the instruction on quotes and watch lists). However, it is offered in small pieces, at the time when that information is relevant and useful to the caller's current activity.

(2)



SYSTEM: Welcome to Princeton Brokerage. I can help you get quotes, set up a watch list, trade equities or options, and access your account information. To get a quote, simply say the name of the company. To set up a watch list, say, "Set up watch list," and then you will be asked for each company you want to put in your list. After you set up a watch list, you can get quotes for all of the companies on your list simply by saying, "Watch list." To make additions or deletions, say, "Edit watch list." To make a trade, say, "Buy" or "Sell," followed by the name of the company you want to trade, the number of shares, and the price. For example, you can say, "Buy a hundred shares of Intel at eighty-two," or "Sell fifty shares of Apple at the market." To trade options, say, "Options." To access your account information, say, "Account information." You will then be able to ask for your cash position, your holdings on any particular company, or a summary of all of your holdings.

**(3)**

SYSTEM: Welcome to Princeton Brokerage. I can help you get quotes, set up a watch list, trade equities and options, and access your account information. What would you like to do?

CALLER: I'd like a quote.

SYSTEM: To get a quote, just say the name of the stock, mutual fund, or index name.

CALLER: IBM.

SYSTEM: International Business Machines is selling at eighty two, up one point three six.

CALLER: Cisco.

SYSTEM: Cisco is selling at seventeen point two five, up point one seven.

CALLER: I wanna set up a watch list.

SYSTEM: Sure, watch list. But first, let me tell you a bit about how it works. I'm going to help you put some companies on your list. In the future, you can get quotes on your companies simply by saying, "Watch list." Now, what's the first company to add to your watch list?

CALLER: Cisco.

SYSTEM: What's the next company name? Or, you can say, "Finished."

CALLER: IBM.

SYSTEM: Tell me the next company name, or say, "Finished."

CALLER: Intel.

SYSTEM: Next one?

CALLER: America Online.

SYSTEM: Next?

CALLER: <pause>

SYSTEM: Sorry, what's the next company name? <pause> Or just say, "Finished."

CALLER: I'm finished.

SYSTEM: Okay! Now whenever you want to get quotes for the companies on your watch list, just say, "Watch list." And to add or delete a company, say, "Edit my watch list." Now, what would you like to do next?

9.3 Attention

Attention is the process of selecting what to focus on, given the multitude of possibilities available (Preece, Rogers, and Sharp 2002). The way you present information to callers can have a significant effect on how easily they can attend to the information of greatest interest.

Let's look at an example of reading back flight information in a travel planning application. Imagine that a caller has requested a flight from New York to

Boston “this afternoon.” Assuming that the system has found four flights in its database that match the caller’s criteria, it must inform the caller of the possibilities and request a selection. Imagine the following presentation of the list of four possible flights:



(4)

SYSTEM: United Airlines flight 47 leaves New York Kennedy Airport at 1 p.m. from gate 36 and arrives at Boston Logan at 1:45 p.m. at gate 22. American Airlines flight 243 leaves New York Kennedy Airport at 2:15 p.m. from gate 12 and arrives at Boston Logan at 3 p.m. at gate 47. American Airlines flight 260 leaves New York Kennedy Airport at 3:45 p.m. from gate 15 and arrives at Boston Logan at 4:30 p.m. at gate 42. United Airlines flight 52 leaves New York Kennedy Airport at 5 p.m. from gate 38 and arrives at Boston Logan at 5:45 p.m. at gate 31. Which would you like?

Whether or not one of these flights meets the caller’s needs, the presentation of the information is so cluttered as to derail the caller’s ability to attend to the information that is most important for the decision. Now imagine the following alternative:



(5)

SYSTEM: There are flights at 1 p.m., 2:15, 3:45, and 5. Which would you like?

CALLER: How about the 2:15 flight.

SYSTEM: American Flight 243 leaves New York Kennedy Airport at 2:15 p.m., arriving at Boston Logan at 3. Would you like to book this flight?

In this case, only the information of greatest concern to the caller is provided, easing the decision. If your application will present possibly complex information, you should note that fact during requirements definition. You should make sure you understand the prospective caller’s goals, priorities, and decision criteria. In this way, you can optimize the presentation of information and not unduly challenge the caller’s ability to attend to the information of greatest concern.

In some cases, divided attention is inevitable. For example, when users are driving, situations may arise that temporarily demand their complete attention. Systems designed for use while driving must accommodate this need by providing the caller with control over the pacing of the interaction. Users might exert such control by, for example, using a pause/resume feature. Or the application might clarify the conversational context after a timeout so that the caller can seamlessly continue with the dialog. (Note: The issue of VUI design for driver safety is an important area that needs careful research. The suggestions in this paragraph are

meant to be illustrative only. They are not based on research with subjects in real or simulated driving situations.)

In other cases, you may desire to communicate information without derailing the caller's current activity. Imagine a personal agent application that manages callers' phone calls and voice mail and can read them their e-mail over the phone. If a new voice mail message arrives while users are listening to e-mail, the application can use a recognizable earcon to notify them without interrupting the flow of e-mail reading. The message "New voice mail has arrived" is communicated via the earcon without disrupting the caller's attention.

In general, the first step in facilitating attention is to understand the caller's goals and priorities. Then you can design strategies to make only the pertinent information available. At the same time, the system can accommodate the caller's needs to attend to information and events outside the purview of the application.

9.4 Conclusion

We have reviewed guidelines to help you deal with cognitive challenges, including conceptual complexity, memory load, and attention. As with all design guidelines, they must be applied with careful consideration of context.

The next two chapters discuss the accommodation of conversational expectations. Chapter 10 discusses issues of prompt wording (*what* the system says), and Chapter 11 discusses issues of prompt prosody (*how* the system says it).

