7

Designing the User and System Interfaces



Chapter Outline

- User and System Interfaces
- Understanding the User Interface
- User-Interface Design Concepts
- The Transition from Analysis to User-Interface Design
- User-Interface Design
- Identifying System Interfaces
- Designing System Inputs
- Designing System Outputs

Learning Objectives

After reading this chapter, you should be able to:

- Describe the difference between user interfaces and system interfaces
- Describe the historical development of the field of human-computer interaction (HCI)
- Discuss how visibility and affordance affect usability
- Describe user-interface guidelines that apply to all types of user-interface types and additional guidelines specific to Web pages and mobile applications
- Create storyboards to show the sequence of forms used in a dialog
- Discuss examples of system interfaces found in information systems
- Define system inputs and outputs based on the requirements of the application program
- Design printed and on-screen reports appropriate for recipients

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

OPENING CASE

Interface Design at Aviation Electronics

Bob Crain was admiring the user interface for the manufacturing support system that was recently installed at Aviation Electronics (AE). Bob is the plant manager for AE's Midwest manufacturing facility, which produces aviation devices used in commercial aircraft. These aviation devices provide guidance and control functions for flight crews, and they provide the latest safety and security features that pilots need when flying commercial aircraft.

The manufacturing support system is used for all facets of the manufacturing process, including product planning, purchasing, parts inventory, quality control, finished goods inventory, and distribution. Bob was extensively involved in the development of the system for several years, including the initial planning and development. The system reflected almost everything he knew about manufacturing. The information systems team that developed the system relied extensively on Bob's expertise. That was the easy part for Bob.

What particularly pleased Bob was the final user interface. He had insisted that the development team consider the entire user experience from the very beginning. He didn't want just another cookie-cutter transaction processing system. He wanted a system that acted as a partner in the manufacturing process—much the way that AE's guidance and control system interfaces acted as a pilot's partner.

The first manager assigned to the project placed a low priority on user-interface design. When Bob asked why user-interface design wasn't a key focus of early iterations, the manager replied, "We'll add the user interface later, after we work out the accounting controls." When Bob insisted that the project manager be replaced, the information systems department sent Sara Robinson to lead the project.

Sara had a completely different attitude; she started out by asking about events that affect the manufacturing process and about cases in which users needed support from the system. Although she had a team of analysts working on the accounting transaction details from the beginning, she always focused on how the user would interact with the system. Bob and Sara conducted meetings to involve users in discussions about how they might use the system, even asking users to act out the roles of the user and the system in carrying on a conversation.

At other meetings, Sara presented sketches of screens and asked users to draw on them to indicate the information they wanted to see and options they wanted to be able to select. These sessions produced many ideas. For example, it appeared that many users didn't sit at their desks all day; they needed larger and more graphic displays they could see from across the room. Many users needed to refer to several displays, and they needed to be able to read them simultaneously. Several functions were best performed by using graphical simulations of the manufacturing process. Users made sketches showing how the manufacturing process actually worked, and the team used these sketches to define much of the interface. Sara and her team kept coming back every month or so with more examples to show, asking for more suggestions.

When the system was finally completed and installed, most users already knew how to use it because they had been so involved in its design. Bob knew everything the system could do, but he had his own uses for it. He sat at his desk and clicked the Review Ongoing Processes button on the screen, and the manufacturing support system gave him his morning briefing.

Overview

Information systems interact with people and with other systems. Because few systems operate autonomously or in isolation, designing the interfaces (inputs and outputs) between a system and its users and environment is an important system development task. Poorly designed interfaces with people can result in a system that operates less than optimally or doesn't fulfill its purpose. For example, a human resources system that has a poorly designed user interface may reduce organizational efficiency and be a source of data entry errors. A customer-facing system with a poorly designed user interface might motivate customers to take their business elsewhere. As with user-oriented interfaces, poorly designed interfaces to other automated systems can be a source of errors or inefficiency. Thus, the design of a system's interfaces is an important part of a system development project.

Inputs and outputs of the system are an early concern of any system development project. The project plan lists key inputs and outputs that the analyst identified when defining the scope of the system. During the analysis phase, analysts will have discussed inputs and outputs early and often with

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

system stakeholders to identify users and actors that affect the system and that depend on the information it produces. Requirements models produced during analysis also emphasize inputs and outputs. For example, use case descriptions define inputs and outputs that occur during a use case. The inputs and outputs are further defined as messages and returns in system sequence diagrams (SSDs).

User and System Interfaces

system interfaces inputs or outputs that require minimal human intervention

A key step in systems design is to classify the inputs and outputs for each event as either a system interface or a user interface. **System interfaces** are inputs and outputs that require minimal human intervention. They might be inputs captured automatically by such special input devices as scanners, electronic messages to or from another system, or transactions captured by another system. Many outputs are considered system interfaces if they primarily send messages or information to other systems (e.g., a pickup notification to a shipping company) or if they produce reports, statements, or documents for external agents or actors without much human intervention (e.g., end-of-month credit card statements e-mailed to cardholders).

user interfaces system interfaces that directly involve a system user

User interfaces are inputs and outputs that more directly involve a system user. User interfaces can be for internal or external users. Their design varies widely depending on such factors as interface purpose, user characteristics, and characteristics of a specific interface device. For example, although all user interfaces should be designed for maximal ease of use, other considerations, such as operational efficiency, may be important for internal users who can be trained to use a specific interface optimized for a specific hardware device (e.g., a keyboard, a mouse, and a large high-resolution display). In contrast, a quite different user interface might be designed for a customer-facing system that assumes a cell phone as the input/output device.

In most system development projects, analysts separate the design of system interfaces from the design of user interfaces because the each requires its own expertise and technology. But as with the design of any system component, considerable coordination is required.

Understanding the User Interface

Many people think the user interface is developed and added to the system near the end of the development process, but the user interface is much more important than that. It is everything that the end user comes in contact with while using the system—physically, perceptually, and conceptually (see **Figure 7-1**).

From a user perspective, the user interface is the entire system. The programs, scripts, databases, and hardware behind the interface are irrelevant. Design techniques that embody this view of user interfaces are collectively called **user-centered design**, which emphasizes three important principles:

user-centered design design techniques that embody the view that the user interface is the entire system

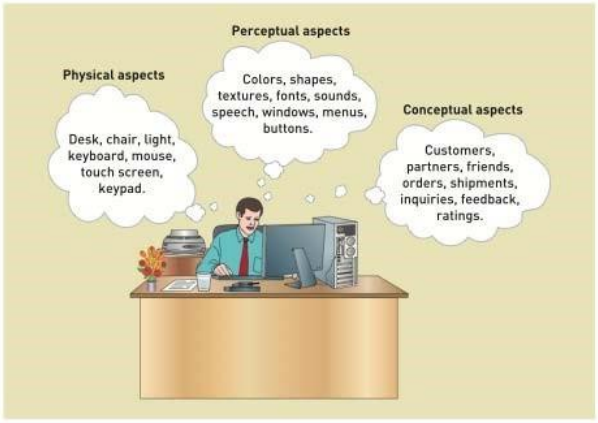
- Focus early on the users and their work.
- Evaluate designs to ensure usability.
- Use iterative development.

The early focus on users and their work is consistent with the approach to systems analysis in this text. User-oriented analysis and design tasks are performed as early as possible and are often given higher priority than other tasks. For example, such user-oriented analysis tasks as stakeholder identification and interviews occur early in the project. User interfaces are designed in early iterations, and user-related design decisions drive other design decisions and tasks.

The early focus on users and their work goes beyond issues of task ordering and priority. It embodies an all-encompassing attempt to understand users and answer such questions as: What do they know? How do they learn? How do

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-1
User-centered design



they prefer to work? What motivates them? The extent of user-oriented focus does vary with the type of system being developed. For example, if the system is a shrink-wrapped desktop application marketed directly to end users, the focus on users and their preferences is intense.

usability degree to which a system is easy to learn and use

The second principle of user-centered design is to evaluate designs to ensure usability. **Usability** refers to the degree to which a system is easy to learn and use. Ensuring usability isn't easy; there are many different types of users with different preferences and skills. Features that are easy for one person to use might be difficult for another. If the system has a variety of end users, how can the designer be sure that the interface will work well for all of them? For example, if it is too flexible, some end users might feel lost. On the other hand, if the interface is too rigid, some users will be frustrated.

Ease of learning and ease of use are often in conflict. For example, menu-based applications with multiple forms, many dialog boxes, and extensive prompts and instructions are easy to learn; indeed, they are self-explanatory. And easy-to-learn interfaces are appropriate for systems that end users use infrequently. But if internal users use the system all day, it is important to make the interface fast and flexible, with shortcuts, hot keys, voice commands, and information-intensive screens. This second interface might be harder to learn, but it will be easier to use after it is learned. Internal users (with the support of their managers) are willing to invest more time learning the system in order to become efficient users.

Developers employ many techniques to evaluate interface designs to ensure usability. User-centered design requires testing all aspects of the user interface. Some usability testing techniques collect objective data that can be statistically analyzed to compare designs. Some techniques collect subjective data about user perceptions and attitudes. To assess user attitudes, developers conduct formal surveys, focus group meetings, design walk-throughs, paper-and-pencil evaluations, expert evaluations, formal laboratory experiments, and informal observation.

The third principle of user-centered design is iterative development—that is, doing some analysis, then some design, then some implementation, and then repeating the processes. After each iteration, the project team evaluates the work on the system to date. Iterative development keeps the focus on the user by

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

continually returning to the user requirements during each iteration and by evaluating the system after each iteration. As with the principle of early focus on users and their work, this principle is reflected throughout this textbook in its approach to system development in general and analysis and design tasks in particular.

Metaphors for Human-Computer Interaction

Widespread use of visually oriented user interfaces debuted in the mass market with the Apple Macintosh in the 1980s and became widespread with various versions of Microsoft Windows in the 1990s. To make computers easier to use and learn, designers of early visually oriented interfaces adopted **metaphors**, which are analogies between features of the user interface and aspects of physical reality that users are familiar with. Metaphors are still widely applied to user-interface design, as described in **Table 7-1**.

metaphors analogies between features of the user interface and aspects of physical reality that users are familiar with

Figure 7-2 is a screen capture of a computer running Windows that illustrates the **direct manipulation**, **desktop**, and **document metaphors**. The entire display is visually similar to the surface of a physical desktop. Icons and pictures for commonly used tools are located on the left and right sides. The icons can be directly manipulated with a mouse or another pointing device. The windows in the center frame are documents that are visually similar to paper pages laid on the surface of a desk, with a sticky note attached to one of the pages.

direct manipulation metaphor metaphor in which objects on a display are manipulated to look like physical objects (pictures) or graphic symbols that represent them (icons)

desktop metaphor metaphor in which the visual display is organized into distinct regions, with a large empty workspace in the middle and a collection of tool icons around the perimeter

document metaphor metaphor in which data is visually represented as paper pages or forms

The direct manipulation, desktop, and document metaphors emphasize displayed objects with which the user interacts. The **dialog metaphor** emphasizes the communication that occurs between a user and a computer, conceptualized as a conversation. In a conversation or dialog between two people, each person listens and responds to questions and comments from the other person, with the information being exchanged in a sequence. The dialog metaphor is another way of thinking about human-computer interaction because the computer "listens to" and "responds" to user questions or comments, and the user "listens to" and "responds" to the computer's questions and comments. **Figure 7-3** illustrates a conceptual dialog between user and computer.

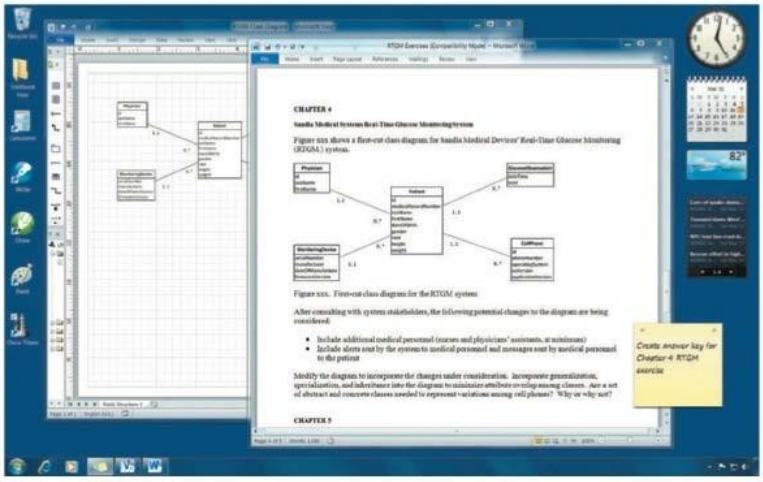
dialog metaphor metaphor in which user and computer accomplish a task by engaging in a conversation or dialog via text, voice, or tools such as labeled buttons

TABLE **7-1** *Commonly used metaphors for user-interface design*

Metaphor	Description	Example
Direct manipulation	Manipulating objects on a display that look like physical objects (pictures) or that represent them (icons)	The user drags a folder icon to an image of a recycle bin or trash can to delete a collection of files.
Desktop	Organizing visual display into distinct regions, with a large empty workspace in the middle and a collection of tool icons around the perimeter	At computer startup, a Windows user sees a desktop, with icons for a clock, calendar, notepad, inbox and sticky notes (the computer interface version of a physical Post-It note).
Document	Visually representing the data in files as paper pages or forms. These pages can be linked together by references (hyperlinks)	The user fills in a form field for a product he or she owns, and the manufacturer's Web site finds and displays the product's manual as an Adobe Acrobat file, which contains a hyperlinked table of contents and embedded links to related documents.
Dialog	The user and computer accomplishing a task by engaging in a conversation or dialog by using text, voice, or tools, such as labeled buttons	The user clicks a button labeled "troubleshoot" because the printer isn't working. The computer prints questions on the display, and the user responds by typing answers or selecting responses from a printed list.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

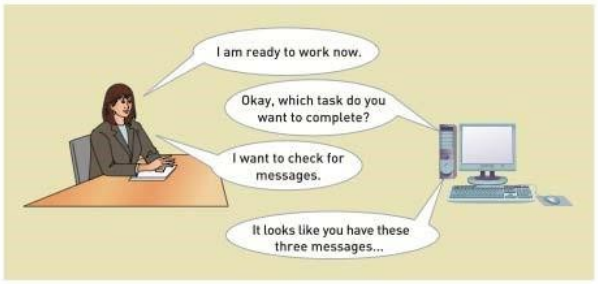
FIGURE 7-2 *The direct manipulation, desktop, and document metaphors on a typical computer display*



The dialog metaphor can be implemented in various ways in user interfaces. A direct approach uses speech generation and recognition over a voice communication channel, as commonly encountered when calling the customer support number of a large company. A computerized voice asks a series of questions, listens for the answer to each question, and responds to the answers. Another implementation of the dialog metaphor uses questions or instructions displayed by a user through text and responses as well as counter-questions displayed by the computer through text. To minimize the need for user typing, responses to computer questions might be limited to a specific set of possibilities displayed to the user in the form of a list from which the user selects the most appropriate response with a mouse click or by touching a display surface.

Regardless of the specific form of implementation, specifying dialogs between user and computer is one of the more powerful tools used by user-interface designers. Although written and spoken languages vary across the globe, conversation and dialog are fundamental and universal human skills. Modeling the

FIGURE 7-3
The dialog metaphor for user-computer interaction



Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

interaction between user and computer as a dialog enables the user to incorporate language and related skills that have been honed from an early age.

User-Interface Design Concepts

Many IT researchers and practitioners have published articles, books, and Web sites that offer guidance in user-interface design. Although some guidelines have changed as user-interface technology has changed, many guidelines are universal, having been around for many decades and technology generations. We review some of these universal guidelines in this section and then move on to user-interface development, with specific guidelines for specific interface types, later in this chapter.

Affordance and Visibility

human-computer interaction (HCI) field of study concerned with the efficiency and effectiveness of user interfaces vis-à-vis computer systems, human-oriented input and output technology, and psychological aspects of user interfaces

Donald Norman is a leading researcher in **human-computer interaction (HCI)**, a field of study concerned with the efficiency and effectiveness of user interfaces to computer systems, human-oriented input and output technology, and psychological aspects of user interfaces. Norman proposes two key principles to ensure good interaction between a person and a machine: affordance and visibility. Both principles apply to user-interface controls, which are elements of a user interface that a user manipulates to perform tasks. Examples of controls include menus, buttons, pull-down lists, sliders, and text entry boxes.

affordance when the appearance of a specific control suggests its function

Affordance means that the appearance of a specific control suggests its function—that is, the purpose for which the control is used. For example, a control that looks like a steering wheel suggests that it is used for turning. Affordance can also be achieved by a user-interface control that the user is familiar with in another context. For example, the media player control icons shown in **Figure 7-4** were first widely used on audiotape and videotape players in the 1970s and have continued to be used in such devices as DVD and portable music players. They are widely incorporated into computer interfaces because so many users are familiar with them.

visibility when a control is visible so that users know it is available, the control providing immediate feedback to indicate that it is responding to the user

Visibility means that a control is visible so users know it is available; it also means that the control provides immediate feedback to indicate that it is responding. For example, the mute button shown in Figure 7-4(a) changes its appearance when the user moves the mouse pointer over it, as shown in Figure 7-4(b). When the user presses the button, it changes its appearance, as shown in Figure 7-4(c).

Visibility and affordance are relatively easy to achieve when the design target is a commonly used platform, such as an iPad, a cell phone running the Android operating system, or a PC running Windows. Such platforms have well-defined user-interface design guidelines and a library of user-interface features and functions that can be reused by application software. When a designer incorporates user-interface objects and styles from these libraries, he or she is tapping into users' experience with similar user interfaces from other applications on those platforms.

Web user-interface design is less standardized because Web browsers are intentionally platform-neutral. Designers can choose from a wide variety of user-

FIGURE **7-4 a-c**
Visibility and affordance in media player controls



Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

interface libraries, each with its own user-interface objects and styles. Attention to visibility and affordance is especially important in Web interface design because there are no real standards that provide a preexisting framework of user familiarity.

Consistency

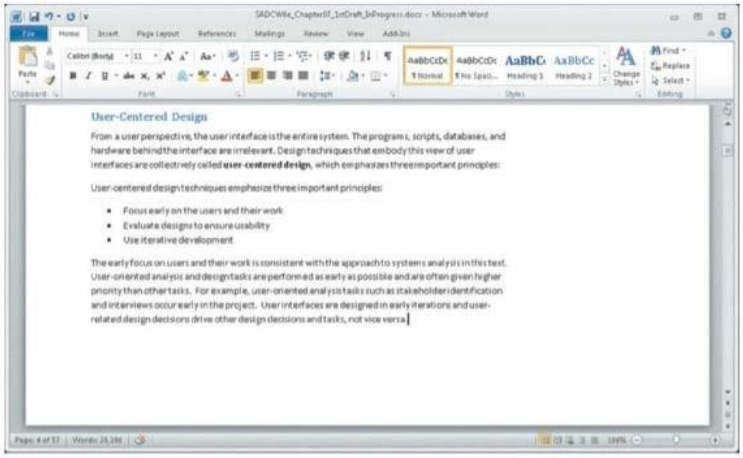
User interfaces should be designed for consistency in function and appearance. The way that information is arranged on forms, the names and arrangement of menu items, the size and shape of icons, and the sequence followed to carry out tasks should be consistent throughout the system. Why? People are creatures of habit. After we learn one way of doing things, it is difficult to change. When we operate a computer application, many of our actions become automatic; we don't think about what we are doing.

Figure 7-5 shows the display of Microsoft Word, which illustrates many aspects of consistency among applications that run under Windows and across the various components of the Microsoft Office suite. Icons that appear in the upper-left and upper-right corners of the window frame are standardized across many Windows applications; thus, users know where to look for them and what their functions are at a glance. Similarly, the scrollbar on the right and the zoom in/out slider and resize handles in the lower-right corner are standard across many Windows applications. The top menus and toolbars are similar across Word, PowerPoint, Publisher, and other programs within Microsoft Office. An experienced user of one program learns others more quickly due to these similarities.

Shortcuts

User interfaces and dialogs designed for novices are often an annoyance and impediment to experienced users' productivity. Users who work with an application repeatedly or for long time periods want shortcuts for frequently used functions, which minimize the number of keystrokes, mouse clicks, and menu selections required to complete tasks. Examples include voice commands as well as shortcut keys, such as Windows keyboard sequences Ctrl+C for copy and Ctrl+V for paste. Application designers should use standard shortcuts when available or build their own.

FIGURE **7-5** *Microsoft Word interface features used in various Windows applications*



Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Feedback

Every action a user takes should result in some type of feedback from the computer so the user knows the action was recognized. Feedback can take many forms in a user interface, including:

- Audible feedback, such as clicking sounds when keys are pressed and beeps when on-screen buttons are pressed
- Visible feedback, such as the icon changes shown in Figures 7-4(b) and 7-4(c), or the progress meter shown during the download of a large file

Feedback provides the user with a sense of confirmation and the feeling that a system is responsive and functioning correctly. Lack of feedback leaves the user wondering whether a command or input was recognized or whether the system is malfunctioning. When subsequent processing is delayed by more than a second or two, users may repeatedly press controls or reenter information, resulting in processing errors and user frustration.

Dialogs That Yield Closure

Each dialog should be organized with a clear sequence—a beginning, middle, and end. Any well-defined task has a beginning, middle, and end, so users' tasks on the computer should also feel this way. The user can get lost if it is not clear when a task starts and ends. In addition, users often focus intently on tasks, so when it is confirmed that a task is complete, the user can clear his or her mind and get ready to focus on the next task.

If the system requirements are defined initially as events to which the system responds, each event leads to the processing of one specific, well-defined activity. Each use case can be defined as one or more dialogs, each with a flow of steps and well-defined interactions. Event decomposition sets the stage for dialogs with closure.

Error Handling

User errors are a waste of the time to commit and to correct them. A good user-interface design anticipates common errors and helps the user to avoid them. One way to do this is to limit available options, presenting the user with only valid options for a specific point in a dialog. Adequate feedback, as discussed previously, also helps reduce errors.

When errors do occur, the user interface needs mechanisms to detect then. Validation techniques discussed later in this chapter are useful for catching errors, but the system must also help the user correct the error. When the system does find an error, the error message should state specifically what is wrong and explain how to correct it. Consider this error message that occurs after a user has typed in a full screen of information about a new customer:

The customer information entered is not valid. Try again.

This message doesn't explain what is wrong or what to do next. Furthermore, after this message appears, what if the system cleared the data-entry form and redisplayed it? The user would have to reenter everything previously typed but still have no idea what is wrong. The error message didn't explain it, and now that the typed data has been cleared, the user cannot tell what might have been wrong. A better error message would read more like this:

The date of birth entered is not valid. Check to be sure only numeric characters in appropriate ranges are entered in the Date of Birth field.

The system also should streamline corrective actions. For example, if the user enters an invalid customer ID, the system should tell the user that this has occurred and then place the insertion point in the customer ID text box, with the previously typed number displayed and ready to edit. That way, the user

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

can see the mistake and edit it rather than having to retype the entire ID. The system might also suggest valid values based on past experience or other information that the user has already entered.

Easy Reversal of Actions

Users need to feel that they can explore options and take actions that can be cancelled or reversed without difficulty. This is one way that users learn about the system—that is, by experimenting. It is also a way to prevent errors; as users recognize they have made a mistake, they cancel the action. In the game of checkers, a move isn't final until the player takes his or her fingers off the game piece; it should be the same when a user drags an object on the screen. In addition, designers should be sure to include cancel buttons on all dialog boxes and allow users to go back a step at any time. Finally, when the user deletes something substantial—a file, a record, or a transaction—the system should ask the user to confirm the action and, where possible, delay implementing the action.

A key issue in permitting action reversal is structuring dialogs and corresponding system actions. Novice designers and programmers often assume that user dialog structure and the sequence of corresponding system actions must precisely correspond. For example, a complex transaction may require several discrete steps, each of which accepts data from the user and some of which modify data stored by the system. Although the user dialog should reflect this structure, internal programming doesn't always have to process data as it is received. Instead, it might collect the data as the dialog proceeds and build an internal "to do" list. When the user completes the final step in the dialog, the system can then complete processing for the entire dialog at once. If the user decides near the end of the dialog to cancel the entire sequence, there are no internal changes to undo. Also, user-interface performance may be improved because there are fewer potential processing delays between steps.

Reducing Short-Term Memory Load

People have many limitations, and short-term memory is one of the biggest. Psychologists have demonstrated that people can remember only about seven chunks of information at a time. User-interface designers should avoid requiring the user to remember anything from one form to another or from one dialog box to another during an interaction with the system. If the user has to stop and ask "What was the filename? The customer ID? The product description?" then the design is placing too much of a burden on the user's memory. Memory limitations also apply to steps in a complex process. The interface should help users keep track of where they are in a complex process via visual cues and other aids.

The Transition from Analysis to User-Interface Design

The foundation for user-interface design is laid when use cases are identified and documented, as described in Chapter 3. Use cases that require direct user interaction (i.e., interactive use cases) are the starting point for a dialog, and the corresponding use case, activity, and system sequence diagrams are the initial dialog documentation. Interactive use cases may require the user to input choices and data into the system (such as when making an online order) or may generate outputs in response to a user request (such as when tracking a shipment). During design, dialogs for interactive use cases are further refined by developing menus, forms, and other user-interface elements.

Dialog and user-interface design can proceed in either a top-down or bottom-up fashion. In the top-down approach, menus (groups of related use cases, dialogs, and user interfaces) are defined first, followed by a detailed description of each interactive use case dialog and development of the related

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Systems Analysis and Design in a Changing World, 6e https://jigsaw.vitalsource.com/api/v0/books/9781133713371/print?from...

user-interface elements. In the bottom-up approach, interactive use cases are prioritized, and related dialogs and user interfaces are developed one at a time. Menus are added later in the project when related sets of fully implemented user interfaces are completed. Neither approach is inherently better; either or both can be a good match for a specific project.

Use Cases and the Menu Hierarchy

Menus are a way of grouping large numbers of related use cases or dialogs within a user interface. In all but the smallest systems, menus are needed to present the user with a tractable number of choices per screen, to group related functions together so users can more easily locate them, and to properly sequence related interfaces for a complex event decomposition. **Figure 7-6** shows two different menu styles. In Figure 7-6(a), the mouse pointer is positioned over the Shop item of the upper menu in a Web page. Clicking Shop would display a second Web page containing another menu. Figure 7-6(b) shows a more complex menu design, with three menu levels displayed.

How does a designer decide which use cases and user interfaces to include in which menus, which menus are required, and how many menu levels are required? These decisions are primarily driven by the number of uses cases or menu choices and the limits of human cognition. Menus usually contain five to 10 choices in order to avoid overloading the user (see Figure 7-6(a)). With careful design, more choices can be provided, especially for experienced users (see Figure 7-6(b)). For a typical business system, dividing the total number of interactive use cases by five provides an initial estimate of the number of menus that include all interactive use cases and allows for additional menu items, such as setting options or preferences. If the number of menus is greater than 10, higher-level menus that contain links to other menus are required. For example, in Figure 7-6(b), the upper menu options, such as File, Home, and Review, each link to a second-level menu formatted as a toolbar. Some choices within the toolbar are links to lower-level menus, such as the displayed Show Markup menu.

FIGURE **7-6** *Two different menu styles*



Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

TABLE 7-2 *RMO use cases grouped by actor and subsystem*

Subsystem	Use Case	Users/Actors
Sales	Search for item	Customer, customer service representative, store sales representative
Sales	View product comments and ratings	Customer, customer service representative, store sales representative
Sales	View accessory combinations	Customer, customer service representative, store sales representative
Sales	Fill shopping cart	Customer
Sales	Empty shopping cart	Customer
Sales	Check out shopping cart	Customer
Sales	Fill reserve cart	Customer
Sales	Empty reserve cart	Customer
Sales	Convert reserve cart	Customer
Sales	Create phone sale	Customer service representative
Sales	Create store sale	Store sales representative
Order fulfillment	Ship items	Shipping
Order fulfillment	Manage shippers	Shipping
Order fulfillment	Create backorder	Shipping
Order fulfillment	Create item return	Shipping, customer
Order fulfillment	Look up order status	Shipping, customer, management
Order fulfillment	Track shipment	Shipping, customer, marketing
Order fulfillment	Rate and comment on product	Customer
Order fulfillment	Provide suggestion	Customer

Use cases with common actors and event decomposition or that implement CRUD actions for a specific domain class are good candidates to be grouped into a single menu or related group of menus. For example, consider the RMO CSMS use cases shown in **Table 7-2**. An initial grouping of these cases by actor and subsystem is a good starting point for menu design.

Table 7-3 shows a grouping of the use cases in Table 7-2 into four menus. Each menu collects uses cases from one subsystem for a customer or internal sales representative. The number of menu choices ranges from four to seven, which won't overload any one menu and may enable multiple menu levels to displayed at one time. A dialog design is created for each menu option. After dialog design proceeds, the designer may redefine the menu options or structure. In fact, designers often discover missing or incomplete use cases during user-interface design, which results in a brief return to analysis activities to complete the documentation.

Menus usually include options that are *not* activities or use cases from the event list. Many options are related to the system controls, such as account maintenance or database backup and recovery, which are discussed later in this chapter. Other added items include help links as well as links to other menus or subsystems.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

TABLE 7-3 *RMO CSMS use cases grouped into first-cut menus by similar function and user*

Menu Description	Menu Choices (Use Cases)	Intended User(s)
Shopping cart functions (primary or reserve)	■ Search for item ■ View product comments and ratings ■ View accessory combinations ■ Switch carts (primary to reserve or vice versa) ■ Fill shopping cart ■ Empty shopping cart ■ Check out shopping cart	Customer
Sale creation	■ Search for item ■ View product comments and ratings ■ View accessory combinations ■ Create sale	Customer service and store sales representatives
Order shipment	■ Ship items ■ Manage shippers ■ Create backorder ■ Create item return ■ Look up order status ■ Track shipment	Customer service and store sales representatives
Customer order control	■ Look up order status ■ Track shipment ■ Create item return ■ Rate and comment on product ■ Provide suggestion	Customer

Dialogs and Storyboards

After identifying all required dialogs, the designers must document the dialogs. Many options exist; there are no *de facto* standards. One approach is to list the key steps followed for the dialog with a written description of what the user and computer do at each step. The format for writing these steps can follow the activity diagram described in Chapter 2 or the use case description format in Chapter 4 or it can be more free form.

Designers can also document dialog designs by writing out how the user and system might interact if they were two people engaged in conversation. Sometimes, the designer can go right from the dialog to sketching a series of screens that make the design more visible. Consider the *Check out shopping cart* use case for the RMO CSMS. A dialog between the system and user might follow this pattern:

SYSTEM:	What would you like to do?
USER:	I'd like to check out.
SYSTEM:	Okay. What is your e-mail address or account number?
USER:	My e-mail address is nwells22@gmail.com.
SYSTEM:	Fine. You are Nancy Wells at 1122 Silicon Avenue. Correct?
USER:	Yes.
SYSTEM:	All items in your cart are in stock and can be shipped today. Because your order subtotal is over $100, you qualify for free UPS ground shipping (3–5 days). Other shipping options include next day ($35.00), two day ($20.00), and USPS parcel post ($11.70, 5–7 days). How would you like your items to be shipped?

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

USER:	Free UPS ground shipping.
SYSTEM:	We have a shipping address on file for you (1122 Silicon Avenue). Do you want your order shipped to that address?
USER:	No.
SYSTEM:	To what address would you like the order shipped?
USER:	John Wells, 1612 Jefferson Street NE, Albuquerque, NM 87123.
SYSTEM:	Okay, the total charge is $125.56 ($117.90 item subtotal plus $7.66 sales tax). Shall I charge that amount to your credit card on file (a Visa with an account number ending in 0899)?
USER:	Yes.
SYSTEM:	Your payment has been approved, and your order is being prepared for shipment. A confirmation e-mail has been sent to you, and another will be sent with a shipment tracking number when the order is shipped later today. Can I help you with anything else?
USER:	No.

storyboarding sequence of sketches of the display screen during a dialog

There are many possible implementations of this scenario. Before deciding on a specific implementation, an analyst often uses a technique called **storyboarding**—that is, showing a sequence of sketches of the display screen during a dialog. The sketches don't have to be very detailed to show the basic design concept. Designers can implement a storyboard with a visual programming tool, such as Visual Basic, but using simple sketches drawn with a graphics package can help keep the focus on the fundamental design ideas and avoid biasing the design to the capabilities of one specific application development tool.

Figure 7-7 shows the storyboard for the dialog based on the *Check out shopping cart* use case. The screen formats are primitive though sufficiently detailed to show all the information presented to and entered by the user. The storyboard can be reviewed by users and designers to identify missing or extraneous information and to discuss various options for final implementation, which might be based on a Web page shown on a large display, a traditional Windows dialog, or the user interface for a mobile device app.

User-Interface Design

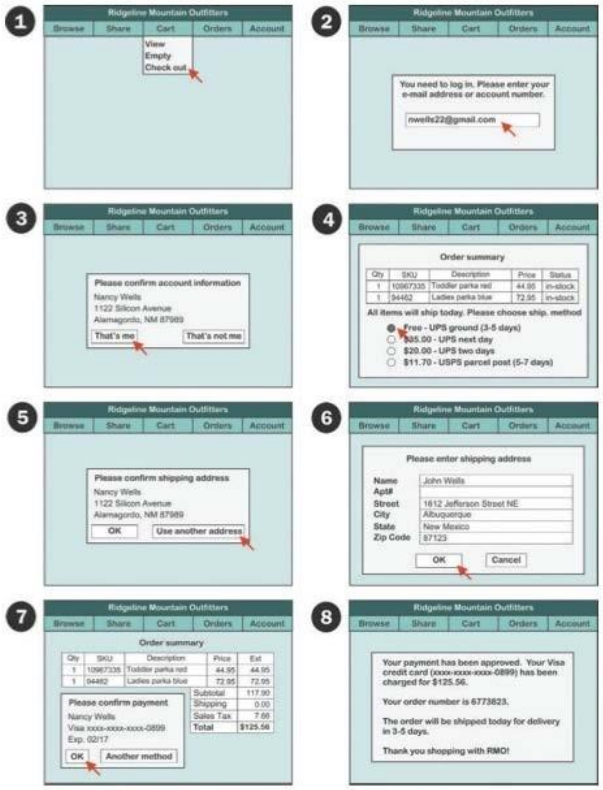
As computing devices have proliferated, available devices and technologies for user interfaces have grown more powerful and diverse. Displays range from large-format, flat-panel monitors used with desktop computers to much smaller displays on tablets and cell phones. Displays can be supplemented with such simple sounds as clicks and beeps, music, or speech. User input can be captured via speech, touch screen, keyboard and mouse, or digital imaging (scanning).

As the range of user-interface technologies has increased, so has the need to create multiple user interfaces. For example, e-commerce applications typically have one user interface for desktop and laptop computers with large displays, another for cell phones with small displays, and sometimes a third for midsized displays, such as an iPad's. There may even be variations in the user interface among similar devices. For example, a user interface intended for a cell phone might be a generic version designed to run within the phone's Web browser or it might be a customized app with different versions for iPhones and Android-based phones.

Despite user interfaces' wide range of sizes and capabilities, some of their features are used on nearly all computing devices. We begin with a discussion of the common features and then delve into differences.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-7
Storyboard for the Check out shopping cart *dialog*



Guidelines for Designing Windows and Forms

After identifying dialogs, menus, and forms with a storyboard or another technique, the system developer can construct the user interface by using one of the many prototyping tools available. Major issues to consider at this stage of interface design include interface layout and formatting, data keying and entry, and navigation controls.

Interface Layout and Formatting

High-quality interfaces are well laid out, with the fields easily identified and understood. One of the best methods to ensure that interfaces are well laid out is to prototype various alternatives and let users test them. Users will let you know which characteristics are helpful and which are distracting. As you design your interfaces, you should think about these:

- Consistency—All the forms within a system need to have the same look and feel. Consistent use of function keys, shortcuts, control buttons, color, and layout makes a system much more useful and professional looking. If

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-8
The RMO home page



designing for an OS-supported interface (for example, Windows, iPhone, or Android), follow published guidelines to improve app and form consistency.

- Labels and headings—Labels should also be easy to identify and read. A clear, descriptive title at the top of the interface helps to minimize confusion about a form's use.
- Distribution and order—Related fields are usually placed next to each other and can be grouped within a box. Tab order (movement of the cursor or input focus) should follow the users' usual reading order (left to right and top to bottom in the United States and Europe). Blank space should be used so content is easy to distinguish and read.
- Fonts and colors—Variations in font face and size can help users distinguish different parts of the form, but only a handful of font and size variations should be used for larger screens and as few as possible should be used for small screens. Too much variation is visually distracting and can cause eyestrain. Follow similar guidelines for colors. Avoid using too many colors and ensure that they are complementary. Also, be careful about mixing blue with yellow or mixing red with green, lest you make it difficult for color-blind users.

Figure 7-8 shows the home page displayed when a customer views the RMO Web site. The form includes two menu bars near the top that group related functions within the same part of the page. If the user points to the item Shop for Clothing, a submenu is displayed immediately below it with a similar color scheme and font. If the user points to the item Shop for Gear, Shop for Clothing is displayed in black text against a white background, Shop for Gear is displayed in white text with a blue background, and the latter's submenu items are displayed instead. Labels for the menu items are widely spaced and use an easy-to-read font. Except for the logo and picture, the page uses a small number of complementary colors. The title is positioned near the top of the picture and stands out well from the picture background and other page elements.

Data Entry

text box a rectangular box that accepts text typed on a keyboard or recognized from speech input

list box a text box that contains a list of predefined data values

Several types of data-entry controls are widely used in user interfaces, including:

- **Text box**—a rectangular box that accepts text typed on a keyboard or recognized from speech input. Example: Product ID in **Figure 7-9**.
- **List box**—a text box that contains a list of predefined data values. Example: Size in Figure 7-9.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-9
The RMO Product Detail form illustrating typical Microsoft Windows data-entry controls



combo box a text box that contains a predefined list of acceptable entries but permits the user to enter a new value when the list doesn't contain the desired value

radio buttons a group of choices from which the user selects only one; the system then automatically turns off all other buttons in the group

check boxes similar to radio buttons, but the user can select multiple items within the group

- **Combo box**—a text box that contains a predefined list of acceptable entries but permits the user to enter a new value when the list doesn't contain the desired value.
- **Radio buttons**—a group of choices from which the user selects only one; the system then automatically turns off all other buttons in the group. Example: Shipping Method in screen #4 of Figure 7-7.
- **Check boxes**—similar to radio buttons, but the user can select multiple items within the group.

These data-entry controls were developed for the Apple Macintosh and later adopted by Windows and other operating systems. Early Web-browser standards had limited support for data-entry controls, but current versions support the controls described previously and many others. iPhone and Android interfaces support similar controls.

The form in Figure 7-9 could be used by an RMO employee to look up information about a product or to modify information in the catalog. Notice how the title and labels make the form easy to read. The natural flow of the form is top to bottom, with related fields placed together. Navigation and close buttons are easily found but aren't in the way of data-entry activity. The form uses standard Microsoft Windows controls, including text, list, and combo boxes. Though not visible in the figure, the form includes features that optimize it for frequent users, including standard Windows keyboard shortcuts, top-to-bottom and left-to-right tab order, and autocompletion of some fields based on database lookups and patches to partially entered text.

Navigation and Support Controls

Standard window interfaces provide several controls for navigation and window manipulation. For Microsoft applications, these controls consist of the Minimize, Maximize, and Close buttons in the upper-right corner, horizontal and vertical scrollbars, and so forth (see Figure 7-5 for specific examples). To maintain consistency across applications, it is a good idea to use built-in or standardized navigation controls whenever possible.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Additional Guidelines for Web Browser User Interfaces

Most user-interface designers first learn to develop Web-based interfaces that operate within a Web browser, such as Internet Explorer, Mozilla, Chrome, or Safari. As Web technologies and standards have matured, the differences in capability between browser-based interfaces (e.g., see Figure 7-11) and those that use operating system support libraries (e.g., see Figure 7-9) have all but disappeared. In many respects, browser-based interfaces have become more powerful. Nonetheless, there are some differences that should be considered when designing Web pages and browser-based forms.

Consistency

Consistency is especially critical within Web sites because most sites contain a large number of pages that serve many different purposes and audiences. For example, a typical corporate Web site provides e-commerce functions (e.g., online ordering), information to investors, directory and public contact information, and such product information as specifications and manuals. In essence, a corporate Web site is the gateway to a large collection of systems serving many different users and tasks. Despite the wide variety of users and tasks, the site as a whole is a single system that should support a single look and feel and should project a consistent, appealing, and desirable image for the corporation as a whole.

Most corporations spend considerable resources developing and maintaining their Web pages and ensuring consistency among them. Thus, user-interface designers of specific parts of the Web site must operate within the constraints of a corporate-wide design. **Cascading style sheets (CSS)** are a Web page encoding standard, and they enable a Web site designer to specify parts of a page that will always look the same and parts that will vary by task or audience. They can also constrain choices within the "variable" parts of a page, including placement and appearance of toolbars and menus, fonts, colors, and background images.

cascading style sheets (CSS) Web page encoding standard that enables a Web site designer to specify parts of a page that will always look the same and parts that will vary by task or audience

Figures 7-10 and 7-11 show additional pages from the RMO Web site that are displayed as the customer searches for items and completes an order. The menus, the outline surrounding detailed content, and the color and font options are all constrained by CSS. When the user selects menu items or clicks links or controls, the pages that are displayed reuse these elements to ensure consistent appearance and user interaction.

FIGURE **7-10**
Product detail page from RMO's Web site



Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-11
Shopping cart page from RMO's Web site



Performance Considerations

Web sites in general and browser-based forms in particular are sensitive to application design and to the quality of the network connections between the user's computing device and the servers that host the site. When a user clicks a hyperlink or a control that acts like a hyperlink, the browser sends information entered by the user (if any) to a Web server, along with a request for a new page. That information traverses multiple networks, is received and processed by a server, and then the response (a new page to be displayed) is sent back over the network. The delay between clicking a hyperlink and the display of the requested page depends on the amount of data to be transmitted, the display and network connection speed of the user computing device, the capacity of the networks that carry the messages, and the number of other users and applications that are competing for that network capacity.

A trade-off exists between the amount of information transmitted between the user's computing device and the server and the time it takes for the page to refresh; the more information that is transmitted, the longer the delay. That trade-off is especially important for communication over the Internet, although it is also a significant issue within corporate networks when user desktops and servers share high-speed connections.

There is also a trade-off between the amount of information and other data contained within a Web page and a Web-based application's performance. Pages with extensive information content or with embedded programming can avoid or postpone page refreshes. For example, a page containing a blank order form may be quite small. But many page refreshes will be required if the browser must interact with the server to validate every input as the user enters it. If the page containing the form also contains embedded validation programs, then many server interactions and page refreshes can be avoided. However, the initial download of the page will take longer because there is more content and the validation programs may be slow if the user computing device isn't very powerful (e.g., a relatively inexpensive cell phone).

Designers of Web-based user interfaces must perform a careful balancing act, providing embedded "intelligence" within a page to avoid refreshes but not overloading page content so as to avoid long delays when the user moves from page to page. Thorough testing is the best way to ensure that the right balance

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-12
RMO's home page, with three menu levels



has been found. For example, in **Figure 7-12,** the RMO Web page includes the list of menu items for each of the main menus (e.g., Shop for Clothing) and submenus (e.g., Women's Apparel). As the user points to menu items, submenu content appears automatically without the need for a page refresh. Contrast this to the menu style shown in Figure 7-6(a), where the user must click a menu item and wait for a new page with the next level menu to be downloaded and displayed. The RMO page takes longer to download due to its embedded menu content and related programming, but it avoids page refreshes and their associated delays when displaying submenus.

Pictures, Video, and Sound

Web-based interfaces are often preferred for their inherent ability to mix text, images, and sound. Powerful and compelling interfaces can be constructed, and they are especially important in customer-facing systems. However, heavy use of sound and images exacerbates the previously discussed performance issues and also creates issues of compatibility. The performance implications are most significant for video and high-resolution still images, which consume large amounts of network capacity. For example, the background image in Figure 7-8 would require several hundred kilobytes of downloaded data for high-quality display on a laptop or desktop computer. A smaller image could be substituted for the smaller screen of a cell phone.

Compatibility issues arise for sound and video because there are so many ways in which they can be encoded. Most Web browsers rely on add-on components (sometimes called plug-ins) to play sounds and music and to display video. Unfortunately, all plug-ins don't work with all browsers, particularly older browser versions. Thus, Web site designers must carefully choose which formats and plug-ins will be used. In many cases, designers must create different pages for different plug-ins and write complex programs to query browsers for which plug-ins are present so the proper page can be downloaded.

Users with Disabilities

Designers of all user interfaces must be sensitive to the special needs of persons with disabilities. Because the Web is such a fundamental resource in the modern world, standards have been developed to ensure maximal usability for those who are visually impaired or have limited dexterity. Visually impaired users typically interact with Web pages via text-to-speech software that examines the content of

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

a web page and reads it aloud. Users with limited dexterity often used voice-recognition software to navigate through elements in a page and perform tasks normally done with a keyboard, mouse, or touch screen. Both types of software are examples of a general software class called **assistive technologies**.

assistive technologies software (such as text-to-speech and voice-recognition utilities) that adapts user interfaces to the special needs of persons with disabilities

The World Wide Web Consortium (W3C) is an organization that sets standards for many aspects of the Web, including compatibility with assistive technologies. As of June 2010, it has published a working draft of the User Agent Accessibility Guidelines (UAAG), version 2.0. Although the draft hasn't received final approval as of this writing, many organizations use its guidelines or those of earlier standards to guide the development of their Web-based user interfaces.

Additional Guidelines for Handheld Devices

Designing Web and app-based user interfaces for handheld devices presents additional design challenges, including:

- Small screen size
- Small keyboards and touch screens
- Limited network capacity
- App design guidelines and toolkits

As of 2012, the size of a typical mobile phone screen is approximately 3.5×2.25 inches and approximately 480×320 pixels. The small screen area provides relatively limited space in which to display content. Thus, designers must pare down the user-interface content to ensure readability and to avoid cluttering the screen. **Figure 7-13** shows a sample mobile RMO Web page. Compared to the larger Web pages shown earlier, the mobile page eliminates many elements, including all images except a scaled-down logo. The remaining textual content is abbreviated, and special attention has been paid to contrast and layout in order to ensure maximal readability.

Small keyboards and touch screens also provide limited capabilities for user input. Mobile device user interfaces must avoid detailed textual input whenever possible and must provide touch controls that are well spaced and easy to locate. On many phones, speech-to-text capabilities provided by the phone's operating system can be used to streamline data entry and navigation, although the state of this technology still makes errors relatively common. Thus, designers must not rely too heavily on speech recognition and must ensure that errors are easily detected and corrected.

As of 2012, most mobile phones include connectivity through cell phone networks and Wi-Fi. Most current cell phone networks, described as third-

FIGURE **7-13**
RMO Web page layout for a mobile phone



Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

generation (3G) networks, were originally designed for voice communication, with data communication grafted on as an afterthought. The throughput for 3G networks is usually no more than one-tenth the throughput of a Wi-Fi network. In 2010, deployment of 4G networks began in the United States. 4G networks increase data throughput to approximately that of Wi-Fi, although many users compete for access to that bandwidth.

Because mobile phone data throughput is much more limited than for other computing devices, the performance issues described earlier become much more significant design constraints. Page sizes must be limited to achieve acceptable download and page-refresh rates. High-resolution graphics are used only when absolutely necessary, and bandwidth-consuming video is typically avoided entirely. For RMO's mobile Web site, background graphics are completely avoided and high-resolution images are only used when a customer wants to view product details.

Some organizations deploy custom-developed apps that users can install on their mobile devices. Those apps run within a mobile operating system, such as the iPhone OS, iPad OS, or Google Android OS. Each OS provides a toolkit for user-interface developers and a set of development guidelines that ensure maximal compatibility among apps. Whenever possible, user-interface developers should use these toolkits and guidelines.

Identifying System Interfaces

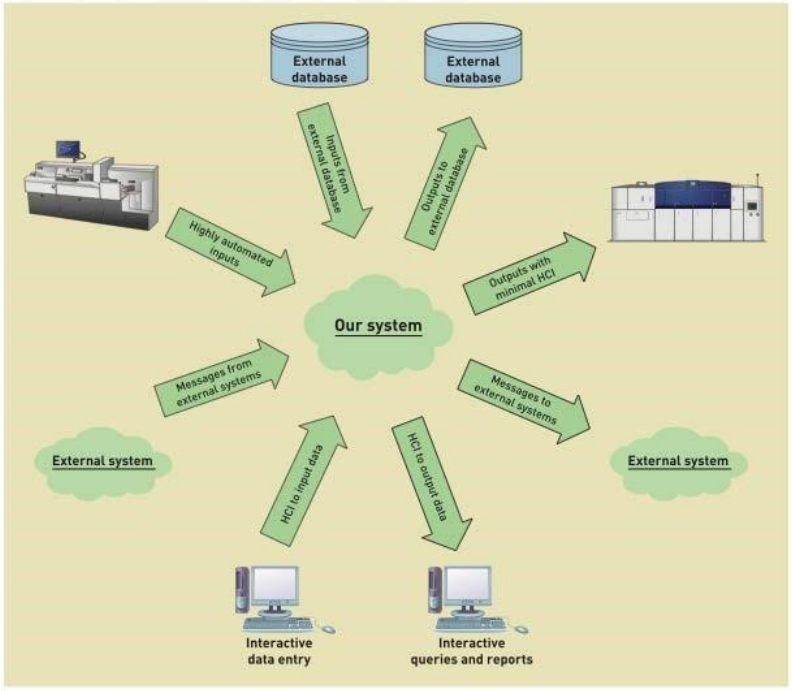
The user interface includes inputs and outputs that directly involve system users. But there are many other system interfaces that process inputs, interact with other systems in real time, and distribute outputs with minimal human intervention. We define *system interfaces* broadly as any inputs or outputs with minimal or no human intervention. Included in this term are displayed and printed outputs for people, such as billing notices, reports, printed forms, and electronic outputs to other automated systems. Inputs that are automated or come from nonuser-interface devices are also included. For example, inputs from automated scanners, bar-code readers, optical character recognition devices, and other computer systems are included as part of a system interface.

The full range of inputs and outputs in an information system is illustrated in **Figure 7-14** and described here:

- Inputs from and outputs to other systems—These are direct interfaces with other information systems, normally formatted as network messages. Electronic data interchange (EDI) and many Web-based systems are integrated with other systems through direct messaging. For example, in RMO's integrated supply chain management system and its customer support system, the arrival of inventory items from a supplier might trigger the shipment of a back-ordered item to a customer.
- Highly automated inputs and outputs—These are captured by devices (such as scanners) or generated by persons who start a process that proceeds without further human intervention. For example, an item in a warehouse might pass a bar-code scanner that records its location as it moves by on a conveyor belt. Also, monthly statements can be printed and mailed through highly automated systems that place the statements within envelopes, apply postage, presort them by ZIP code, and batch them for delivery to the post office.
- Inputs and outputs to external databases—These can supply input to or accept output from a system. EDI messages are more commonly used, but direct interaction with another system's database may be more efficient. For example, RMO's purchasing system could directly place product orders into a supplier's database.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-14 *The full range of inputs and outputs in an information system*



One of the main challenges of EDI is defining the format of the transactions. For example, General Motors—one of the early users of EDI—has thousands of suppliers and thousands of different transaction types, each in a different format. To complicate the situation further, each of these suppliers may be linked via EDI with tens or hundreds of customers, many of whom may also use EDI. So, a single type of transaction may have a dozen or more defined formats. It is easy to see why it is so costly to set up and maintain EDI systems. Even so, EDI is much more efficient and effective than paper transactions, which must be printed and reentered.

Extensible Markup Language (XML) extension of HTML that embeds self-defining data structures within textual messages

XML tags character sequences (such as <name> and </name>) that define the beginning, end, and meaning of the text that appears between them

Modern EDI messages are generally formatted in **Extensible Markup Language (XML)**. XML is an extension of HTML that embeds self-defining data structures within textual messages. So, a transaction that contains data fields can be sent with XML codes to define the meaning of the data fields. Many newer systems are using XML to provide a common system-to-system interface. **Figure 7-15** illustrates a simple XML transaction that can be used to transfer customer information between systems. Data are surrounded by **XML tags**, such as <name> and </name>, that define the beginning, end, and meaning of the text that appears between them.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-15
Customer information formatted as an XML message

```
<customer record>
     <accountNumber>RMO10989</accountNumber>
     <name>William Jones</name>
     <billingAddress>
          <street>120 Roundabout Road</street>
          <city>Los Angeles</city>
          <state>CA</state>
          <zip>98115</zip></billingAddress>
     <shippingAddress>
          <street>120 Roundabout Road</street>
          <city>Los Angeles</city>
          <state>CA</state>
          <zip>98115</zip></shippingAddress>
     <dayPhone>215.767.2334</dayPhone>
     <nightPhone>215.899.8763</nightPhone>
</customer record>
```

XML is called an extensible language because users can define any tags they want to use. For XML-based EDI, both systems must recognize the tags, but after a complete set of codes is established, transactions can include many different formats and still be recognized and processed. Many industries and professional organizations have standards committees that define tags used for EDI.

Designing System Inputs

When designing inputs for a system, the system developer must focus on three areas:

- Identifying the devices and mechanisms that will be used to enter input
- Identifying all system inputs and developing a list with the data content of each
- Determining what kinds of controls are necessary for each system input

Automated Input Devices

The primary objective is to enter or update error-free data into the system. The key term here is *error-free*. Several good practices can help reduce input errors:

- Use electronic devices and automatic entry whenever possible.
- Avoid human involvement as much as possible.
- If the information is available in electronic form, use that instead of reentering the information.
- Validate and correct information at the time and location it is entered.

Automating data entry and avoiding human involvement are essentially different sides of the same coin, although using electronic devices doesn't automatically avoid human involvement. When system developers think carefully about minimizing human input and using electronic input media, they can design a system with fewer electronic input forms and avoid one of the most pervasive sources of input error: typing mistakes by users. Here are some of the more common devices used to avoid human keystroking:

- Magnetic card strip readers
- Bar-code readers

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- Optical character recognition readers and scanners
- Radio-frequency identification tags
- Touch screens and devices
- Electronic pens and writing surfaces
- Digitizers, such as digital cameras and digital audio devices
- Speech-recognition software

The next principle of error reduction is to reuse the information already captured in automated form whenever possible. For example, consider the automated airline check-in process for customers without checked baggage. The customer swipes a credit card or driver's license, and the system queries its own database or an external database to identify the customer and retrieve the reservation information. The retrieved information is displayed to the customer for confirmation. Because the retrieved information is almost always correct, the task of data entry is reduced to a card swipe and the press of a button on a touch screen, eliminating manual data entry and its associated error rate. If the displayed data is incorrect, the final principle of error reduction is applied by having the customer directly enter corrected data.

Defining the Details of System Inputs

The fundamental approach that analysts use to identify user and system inputs is to search documents developed during analysis activities for information flows that cross the system boundary. The analyst examines system sequence diagrams to identify the incoming and outgoing messages for each activity or use case, and the design class diagrams to identify and describe the data content.

Figure 7-16 is a partial system sequence diagram for an object-oriented version of a payroll system. Various use cases have been combined to consolidate the major inputs on a single diagram. The messages that cross the system boundary identify inputs—system inputs and user-interface inputs. Three inputs cross the system boundary:

- updateEmployee (empID, empInformation)
- updateTaxRate (taxTableID, rateID, rateInformation)
- inputTimeCard (empID, date, hours)

The first input is part of a user interface. The other two inputs are from external systems and don't require user involvement. The information from the tax bureau can be sent as a set of real-time messages or in the form of a downloaded input file. The time card information could come into the system in various formats. Perhaps physical time cards are entered via an electronic card reader. Or an input from a subsystem, such as an electronic employee ID card reader, might send time card information at the end of every workday. These last two input messages need to be precisely defined, including their transmission methods, contents, and formats. The point to note here is that a sequence diagram provides a detailed perspective of the user and system inputs to support the use case and the corresponding business event.

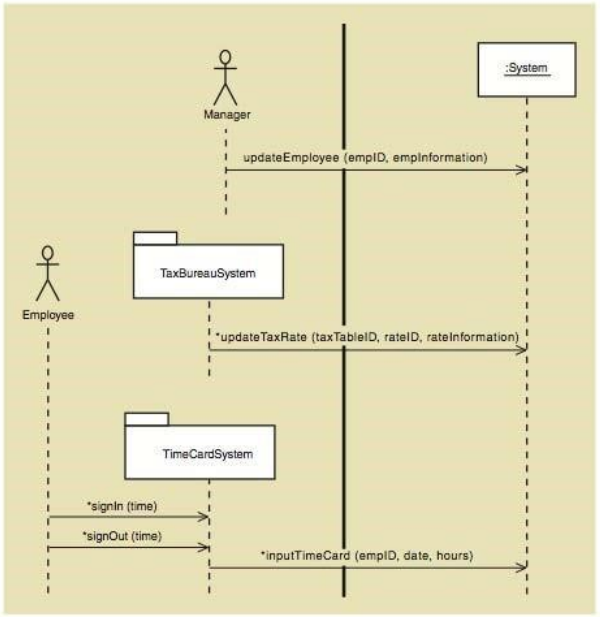
Designing System Outputs

The primary purpose of system outputs is to present information in the right place at the right time to the right people. The tasks in this activity focus on four areas:

- Determining the type of each system output
- Making a list of required system outputs based on application design

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-16
Partial system sequence diagram for payroll system use cases



- Specifying any necessary controls to protect the information provided in the output
- Designing and prototyping the output layout

The purpose of the first two tasks is to evaluate the various alternatives and design the most appropriate approach for each needed output. The list of required system outputs is normally specified during the analysis activities as part of modeling system requirements. During design, the task is to coordinate the production of those outputs with the methods that are identified during the application architecture design.

The third task ensures that the designer evaluates the value of the information to the organization and protects it. Frequently, organizations implement controls on the inputs and system access but forget that output reports often have sensitive information.

As with system inputs, outputs are indicated by messages in sequence diagrams that cross the system boundary—originating from an internal system object and flowing to an external actor. Output messages that are based on an individual object (or record) are usually part of the methods of that object class. To report on all objects within a class, a class-level method is used. A class-level method is a method that works on the entire class of objects, not a single object. For example, a customer confirmation of an order is an output message that contains information for a single order object. However, to produce a summary report of all orders for the week, a class-level method looks at all the orders in the Order class and sends output information for each one with an order date within the week's time period.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Designing Reports, Statements, and Turnaround Documents

Modern information systems have made information much more widely available, with a proliferation of all types of reports—paper and electronic. One of the major challenges organizations face today is to organize the overwhelming amount of information in order to support managerial decision-making. One of the most difficult aspects of output design is to decide what information to provide and how to present it to avoid a confusing mass of complex data.

Report Types

There are four types of output reports commonly provided by an information system:

detailed reports reports that contain specific information on business transactions

summary reports reports that summarize detail or recap periodic activity

exception reports reports that provide details or summary information about transactions or operating results that fall outside a predefined normal range of values

executive reports reports used by high-level managers to assess overall organizational health and performance

- **Detailed reports**—These contain specific information on business transactions—for example, a list of all overdue accounts, with each line of the report presenting information about a particular account. A credit manager could use this report to research overdue accounts and determine actions to collect past-due amounts.
- **Summary reports**—These are often used to recap periodic activity. An example of this is a daily or weekly summary of all sales transactions, with a total dollar amount of sales. Managers often use this type of report to track departmental or division performance.
- **Exception reports**—These provide details or summary information about transactions or operating results that fall outside of a predefined normal range of values. When business is progressing normally, no report is needed. For example, a manufacturing organization might produce a report that lists parts that fail quality control tests more than 0.2 percent of the time.
- **Executive reports**—These are used by high-level managers to assess overall organizational health and performance. They thus contain summary information from activities within the company. They might also show comparative performance with industry-wide averages. Using these reports, executives can assess the competitive strengths or weaknesses of their companies.

Internal versus External Outputs

internal outputs reports or other outputs produced for use within the organization

external outputs reports or other outputs produced for use by people outside the organization

turnaround documents external outputs that includes one or more parts intended to be returned with new data or information

Printed outputs are classified as either internal outputs or external outputs. **Internal outputs** are produced for use within the organization. The types of reports just discussed fall under this category. **External outputs** include statements, notices, and other documents that are produced for people outside the organization. Because they are official business documents for an outside audience, they need to be produced with the highest-quality graphics and color. Examples include monthly bank statements, late notices, order confirmation and packing slips (such as those provided to Ridgeline Mountain Outfitters' customers), and legal documents (such as insurance policies). Some external outputs are referred to as **turnaround documents** because they are sent to a customer but include a tear-off portion that is returned for input later, such as a bill that contains a payment stub to be returned with a check. All these printed outputs must be designed with care, but organizations have many more options for printed output. Today's high-speed color laser printers enable all types of reports and other outputs to be produced.

An example of a detailed report for an external output is shown in **Figure 7-17**. When a customer places an order on the Web, the system will be able to print the order information as a confirmation. Of course, a user can always print the Web screen display by using the browser's print capability, but doing so is time consuming because it includes all the graphics and

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-17 *RMO shopping cart order report*

Ridgeline Mountain Outfitters—Shopping Cart Order

Customer Name: Fred Westing
Customer Number: 6747222

Order Number: 4673064
Today's Date: May 18, 2013

Shipping Address:
936 N Swivel Street
Hillville, Ohio 59222

Billing Address:
936 N Swivel Street
Hillville, Ohio 59222

Qty	Product ID	Description	Size	Color	Price	Extended Price
1	458238WL	Jordan Men's Jumpman Team J	12	White/ Light Blue	$119.99	$119.99
1	347827OP	Woolrich Men's Backpacker Shirt	XL	Oatmeal Plaid	$41.99	$41.99
2	87594255H	Nike D.R.I. – Fit Shirt	M	Black	$30.00	$60.00
1	5858642OR	Puma Hiking Shorts	L	Tan	$15.00	$15.00
					Subtotal	$236.98
					Shipping	$8.50
					Tax	$11.25
					Total	$256.73

Shipping Information:
Shipping Method: Normal 7–10 day
Shipping Company: UPS
Tracking Number: To be sent via email
Email Address: FredW253@aol.com

Payment Information:
American Express ☐ MasterCard ☐ VISA ☒ Discover ☐
Account Number
X X X X - X X X X - X X X X - 5 7 8 4
MO YR
Expiration Date 05 / 15

Thank you for your order. It is a pleasure to serve you.
Check back next week for new weekly specials!!

index links on the page. It is much more user friendly to provide shoppers with a "printer friendly" order confirmation in addition to a Web-based display, as shown in Figure 7-11.

Figure 7-18 is an example of an internal output based on inventory records. The report includes detailed and summary sections, although the figure doesn't show the summary section. A control break is the data item that divides the detailed section into groups. In this example, the control break is on the product item number—called ID on the report. Whenever a new value of the ID is encountered on the input records, the report begins a new control break section. The detailed section lists the transactions of records from the database, and the summary section provides totals and recaps of the information. The report is sorted and presented by product. However, within each product is a list of each inventory item showing the quantity currently on hand.

External outputs can consist of complex, multiple-page documents. A well-known example is the set of reports and statements that you receive with your car insurance statement. This statement is usually a multipage document consisting of detailed automobile insurance information and rates, summary pages, turnaround premium payment cards, and insurance cards for each automobile. Another example is a report of employment benefits, with multiple pages of information customized to the individual employee. Sometimes, the documents are printed in color, with special highlighting or logos. **Figure 7-19** is one page of an example report for survivor protection from an employee benefit booklet. The text is standard wording, and the numbers are customized to the individual employee.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-18 *RMO inventory report*

Ridgeline Mountain Outfitters — Products and Items

ID	Season	Category	Supplier	Unit Price	Special Price	Discontinued
RMO12587	Spr/Fall	Mens C	8201	$39.00	$34.95	No

Description Outdoor Nylon Jacket with Lining

Size	Color	Style	Units in Stock	Reorder Level	Units on Order
Small	Blue		691	150	
	Green		723	150	
	Red		569	150	
	Yellow		827	150	
Medium	Blue		722	150	
	Green		756	150	
	Red		698	150	
	Yellow		590	150	
Large	Blue		1289	150	
	Green		1455	150	
	Red		1329	150	
	Yellow		1370	150	
Xlarge	Blue		1498	150	
	Green		1248	150	
	Red		1266	150	
	Yellow		1322	150	

ID	Season	Category	Supplier	Unit Price	Special Price	Discontinued
RMO28497	All	Footwe	7993	$49.95	$44.89	No

Description Hiking Walkers with Patterned Tread Durable Uppers

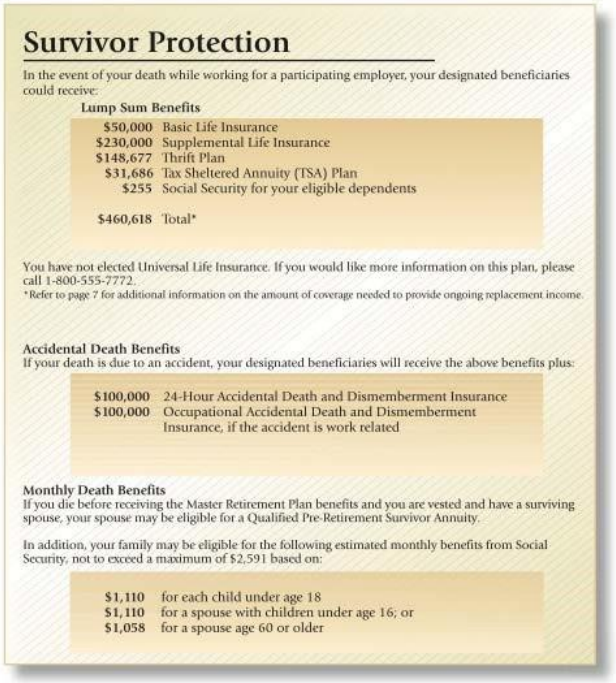
Size	Color	Style	Units in Stock	Reorder Level	Units on Order
7	Brown		389	100	
	Tan		422	100	
8	Brown		597	100	
	Tan		521	100	
9	Brown		633	100	
	Tan		654	100	
10	Brown		836	100	
	Tan		954	100	
11	Brown		862	100	
	Tan		792	100	
12	Brown		754	100	
	Tan		788	100	
13	Brown		830	100	
	Tan		921	100	

Electronic Reports

Organizations use various types of electronic reports, each serving a different purpose and each with its respective strengths and weaknesses. Electronic reports provide great flexibility in the organization and presentation of information. In some instances, screen output is formatted like a printed report but displayed electronically. However, electronic reports can also present information in many other formats. Some have detailed and summary sections, some show data and graphics together, others contain boldface type and highlighting, others can dynamically change their organization and summaries, and still others contain hotlinks to related information. An important benefit of electronic reporting is that it is dynamic; it can change to meet the specific needs of a user in a particular situation. In fact, many systems provide powerful

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-19 *A sample employee benefit report*



Survivor Protection

In the event of your death while working for a participating employer, your designated beneficiaries could receive:

Lump Sum Benefits

$50,000	Basic Life Insurance
$230,000	Supplemental Life Insurance
$148,677	Thrift Plan
$31,686	Tax Sheltered Annuity (TSA) Plan
$255	Social Security for your eligible dependents
$460,618	Total*

You have not elected Universal Life Insurance. If you would like more information on this plan, please call 1-800-555-7772.

*Refer to page 7 for additional information on the amount of coverage needed to provide ongoing replacement income.

Accidental Death Benefits

If your death is due to an accident, your designated beneficiaries will receive the above benefits plus:

$100,000	24-Hour Accidental Death and Dismemberment Insurance
$100,000	Occupational Accidental Death and Dismemberment Insurance, if the accident is work related

Monthly Death Benefits

If you die before receiving the Master Retirement Plan benefits and you are vested and have a surviving spouse, your spouse may be eligible for a Qualified Pre-Retirement Survivor Annuity.

In addition, your family may be eligible for the following estimated monthly benefits from Social Security, not to exceed a maximum of $2,591 based on:

$1,110	for each child under age 18
$1,110	for a spouse with children under age 16; or
$1,058	for a spouse age 60 or older

drill down user-interface design technique that enables a user to select summary information and view supporting detail

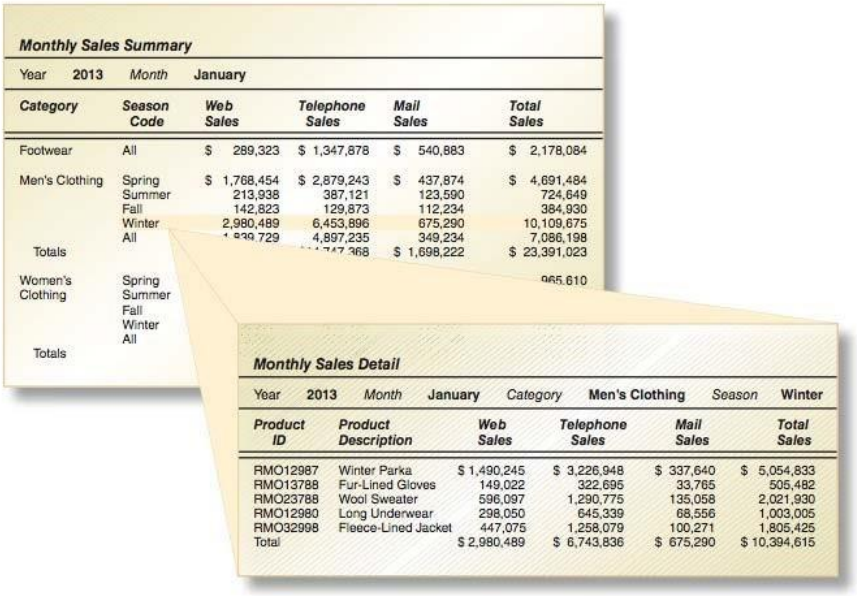
ad hoc reporting capabilities so users can design their own reports on the fly. For example, an electronic report can provide links to further information. One technique, called **drill down**, allows the user to activate a "hot spot hyperlink" on the report, which tells the system to display a lower-level report that provides more detailed information. For example, **Figure 7-20** contains a monthly sales summary. The report provides sales totals grouped by product category and season. However, if the user clicks the hotlink for any season, a detailed report pops up with more detailed sales data.

Another variation of this hotlink capability lets the user correlate information from one report to related information in another report. Most people are familiar with hotlinks from using their Internet browsers. In an electronic report, hotlinks can refer to other information that correlates or extends the primary information. This same capability can be very useful in a business report that, for example, links the annual statements of key companies in a certain industry.

Another dynamic aspect of electronic reports is the capability to view the data from different perspectives. For example, it might be beneficial to view sales commission data by region, by sales manager, by product line, or by time period

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-20 *RMO summary report with drill down to the detailed report*



Monthly Sales Summary

Year 2013 Month January

Category	Season Code	Web Sales	Telephone Sales	Mail Sales	Total Sales
Footwear	All	$ 289,323	$ 1,347,878	$ 540,883	$ 2,178,084
Men's Clothing	Spring	$ 1,768,454	$ 2,879,243	$ 437,874	$ 4,691,484
	Summer	213,938	387,121	123,590	724,649
	Fall	142,823	129,873	112,234	384,930
	Winter	2,980,489	6,453,896	675,290	10,109,675
	All	1,839,729	4,897,235	349,234	7,086,198
Totals			747,368	$ 1,698,222	$ 23,391,023
Women's Clothing	Spring				965,610
	Summer				
	Fall				
	Winter				
	All				
Totals					

Monthly Sales Detail

Year 2013 Month January Category Men's Clothing Season Winter

Product ID	Product Description	Web Sales	Telephone Sales	Mail Sales	Total Sales
RMO12987	Winter Parka	$ 1,490,245	$ 3,226,948	$ 337,640	$ 5,054,833
RMO13788	Fur-Lined Gloves	149,022	322,695	33,765	505,482
RMO23788	Wool Sweater	596,097	1,290,775	135,058	2,021,930
RMO12980	Long Underwear	298,050	645,339	68,556	1,003,005
RMO32998	Fleece-Lined Jacket	447,075	1,258,079	100,271	1,805,425
Total		$ 2,980,489	$ 6,743,836	$ 675,290	$ 10,394,615

or to compare the current data with last season's data. Instead of printing all these reports, you can use an electronic format to generate the different views as needed. Sometimes, long or complex reports include a table of contents, with hotlinks to the various sections of the report. Some report-generating programs provide an electronic reporting capability that includes all the functionality found on Web pages, including frames, hotlinks, graphics, and even animation.

Graphical and Multimedia Presentation

The graphical presentation of data is one of the greatest benefits of the information age. Tools that permit data to be presented in charts and graphs have made information reporting much more user friendly for printed and electronic formats. Information is being used more and more for strategic decision-making as businesspeople examine their data for trends and changes. In addition, today's systems frequently maintain massive amounts of data—much more than people can review. The only effective way to use much of this data is by summarizing and presenting it in graphical form. **Figure 7-21** presents a pie chart and a bar graph—two common ways to present summary data.

Multimedia outputs have become available recently as multimedia tool capabilities have increased. Today, it is possible to see a graphical (possibly animated) presentation of the information on a screen and have an audio description of the salient points. Combining visual and audio output is a powerful way to present information. (Of course, video games are pushing the frontier of virtual reality to include visual, audio, tactile, and olfactory outputs.)

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

FIGURE 7-21 *Sample pie chart and bar graph reports*



As the design of system outputs progresses, it is beneficial to evaluate the various presentation alternatives. Reporting packages can be integrated into the system to provide a full range of reporting alternatives. Developers should carefully analyze each output report to determine the objective of the output and to select the form of the output that is most appropriate for the information and its use.

Chapter Summary

Inputs and outputs can be classified as system interfaces or user interfaces. The user interface is everything the user comes into contact with while using the system—physically, perceptually, and conceptually. There are many different ways to describe the user interface, including the desktop metaphor, the document metaphor, and the dialog metaphor. Dialog design starts with identifying dialogs based on activities or use cases. A storyboard showing sketches of screens in sequence can be drawn to convey the design for review with users, or prototypes can be created using a tool such as Visual Basic. The object-oriented approach provides UML models that can document dialog designs, including sequence diagrams, activity diagrams, and class diagrams.

Each screen and form used in a dialog needs to be designed, and there are guidelines for the layout, selection of input controls, navigation, and help. These guidelines apply to windows forms and to browser forms used in Web-based systems. Designing a dialog for a Web site is similar to creating any other dialog, except users need more information and more flexibility. Additional Web design guidelines apply to designing for the computer medium, designing the whole site, and designing for the user. In addition, because a Web site reflects the company's image to customers, graphic designers and marketing professionals should be involved.

When designing system inputs, the developer identifies the input devices and identifies all system inputs and lists the data content of each. To develop a list of the inputs to the system, designers use sequence and design class diagrams. The process to design the outputs from the system consists of the same steps as for input design.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Key Terms

affordance 193
assistive technologies 207
cascading style sheets (CSS) 204
check boxes 203
combo box 203
desktop metaphor 191
detailed reports 213
dialog metaphor 191
direct manipulation metaphor 191
document metaphor 191
drill down 216
exception reports 213
executive reports 213
Extensible Markup Language (XML) 209
external outputs 213
human-computer interaction (HCI) 193
internal outputs 213
list box 202
metaphors 191
radio buttons 203
storyboarding 200
summary reports 213
system interfaces 189
text box 202
turnaround documents 213
usability 190
user interfaces 189
user-centered design 189
visibility 193
XML tags 209

Review Questions

1. Why is interface design often referred to as dialog design?
2. What are the three aspects of the system that make up the user interface for a user?
3. What are some examples of the physical, perceptual, and conceptual aspects of the user interface?
4. What are the three metaphors used to describe human-computer interaction?
5. A desktop on the screen is an example of which of the three metaphors used to describe human-computer interaction?
6. What type of document allows the user to click a link and jump to another part of the document?
7. List and briefly describe four guidelines for interface layout and formatting that apply to all types of user display and input devices. What additional guidelines apply to Web sites/pages and user interfaces for mobile computing devices?
8. What is the technique that shows a sequence of sketches of the display screen during a dialog?
9. What UML diagram can be used to show how the interface objects are plugged in between the actor and the problem domain classes during a dialog?
10. What are some of the input controls that can be used to select an item from a list?
11. What two types of input controls are included in groups?
12. What popular analogy is used for direct customer access with a Web site when customers shop online?
13. What does XML stand for? Explain how XML is similar to HTML. Also discuss the differences between XML and HTML.
14. How do you identify the data fields of a system interface by using UML and the object-oriented approach?
15. What are the different considerations for output screen design and output report design?
16. What is meant by drill down? Give an example of how you might use it in a report design.
17. What is the danger from information overload? What solutions can you think of to avoid it?

Problems and Exercises

1. Think of all the software you have used. What are some examples of ease of learning conflicting with ease of use?
2. Visit some Web sites and then identify all the controls used for navigation and input. Are they all obvious? Discuss some differences in visibility and affordance among the controls.
3. A common maxim for designing a man-machine interface is that it is better to change the machine than to try to change the human to accommodate

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

the machine. Are there machines (or systems) that you use in your daily life that have room for improvement? Are the current generations of Windows PC and Apple Mac as usable as they might be? If not, what improvements can you suggest? Is the World Wide Web as usable as it might be? If not, what improvements can you suggest? Are we just beginning to see some breakthroughs in usability, or have most of the big improvements already been made?

4. Download and install App Inventor from Google. Use it to develop a prototype interface that implements the storyboard in Figure 7-7.
5. Evaluate the course registration system at your university. List the basic steps of a user's dialog with the system. What are some of the problems with the system from the perspective of ease of learning and ease of use? In what ways is the system inflexible? In what ways is needed information not available? Is too much information provided that distracts from the task at hand?
6. Evaluate the online catalog system at your university library. Write a dialog that shows the interaction between the user and the system. Rewrite the dialog to improve it. Create a storyboard to show how your design would look and feel.
7. Find a Web site with direct customer ordering. Browse through some product descriptions and note the design of the dialog and the Web pages. What do you like and dislike about the design? Evaluate the Web site based on visibility and affordance. Does the site achieve an optimal balance between the number of page refreshes and the delay between page refreshes? Would your answer change if you were using a different computing device, a different network, or accessing the system at a different time of day?

Case Study

Video Rental Service

Develop a storyboard that encompasses the following sample dialog. The system supports rental and download of movies or television shows to a computer or digital video recording and playback device. A dialog between the system and user might follow this pattern:

SYSTEM:	What would you like to do?
USER:	I'd like to rent a movie.
SYSTEM:	Okay. What is your e-mail address or account number?
USER:	My e-mail address is nwells22@gmail.com.
SYSTEM:	Fine. You are Nancy Wells at 1122 Silicon Avenue. Correct?
USER:	Yes.
SYSTEM:	What movie do you want to rent?
USER:	*Titanic*.
SYSTEM:	I have two available movies named *Titanic*. The first was released in 1953 and stars Barbara Stanwyck and Clifton Webb. The second was released in 1997 and stars Leonardo DiCaprio and Kate Winslet. Which do you want to rent?
USER:	Both.
SYSTEM:	Fine. That is *Titanic* (the 1953 release) with a rental charge of $2.99 for seven days and *Titanic* (the 1997 release) with a rental charge of $5.99 for seven days. Would you like to add more rentals?
USER:	That's all.
SYSTEM:	Okay. The total charge is $8.98. Shall I charge that amount to your credit card file (a Visa with an account number ending in 0899)?
USER:	Yes.
SYSTEM:	Your movies are downloading to your computer now. They will expire at midnight seven days from now.
USER:	Okay. Thanks.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

RUNNING CASE STUDIES

Community Board of Realtors

One of the key use cases for the multiple listing service system is *Create new listing*, where the realtor enters all the important information about a new listing he or she has obtained. Realtors want to be able to create a new listing as soon as possible so other realtors and potential buyers will find the listing online. Some like to enter the information while talking with the owner or while inspecting the property. Realtors are rarely in their offices these days, so being able to create a new listing on a mobile device is a key feature of the multiple listing service system.

Consider the information that must be entered when creating a new listing, and list the dialog steps that are necessary. Keep in mind that when designing for a smartphone, less information can be entered in each step compared to a full screen Web application.

Also keep in mind that typing is error-prone and awkward for many users, so think about opportunities to use check boxes, radio buttons, and list boxes to aid selection. Create a storyboard of this use case for a mobile device, showing each step of the dialog that maximizes the use of check boxes, radio buttons, and list boxes.

On the Spot Courier Services

Review the case description and your solution for the Web scenario of the use case *Request package pickup* from Chapter 5. Then, using a presentation tool, such as Microsoft PowerPoint or Apple Keynote, create a storyboard of the Web pages necessary to support the use case.

The case description in Chapter 5 also identified a new use case, which we can call *View scheduled pickups/deliveries*. Based on current technology, write a dialog showing how this might be supported with a portable digital device. You may use any current technology that you deem applicable, such as GPS tracking, map and directions software, and real-time updates of pickup locations. Consider the possibility that the driver may want to get an overview of his or her stops for the entire run, view the next few stops, or just get directions to the next stop.

The Spring Breaks 'R' Us Travel Service

The Spring Breaks R Us social networking subsystem requires an intuitive and engaging user-interface design for mobile devices. But the social networking subsystem can also play an important role in resort security. For example, each resort could use traveler location, interests, activities, and "likes"—all of which are available through the application—to monitor the well-being of travelers staying at the resort. Most spring break travelers are young, and their parents are concerned about their safety—particularly at isolated resorts in foreign countries. SBRU and the participating resorts could keep track of where travelers are and who they are near, monitor messages about activities and parties, and anticipate crowded conditions or vulnerable travelers wandering off-site. Alerts could notify security if conditions veer away from normal or if messages indicate there are problems. For example, if the pool is overcrowded, some action can be taken. If messages refer to places off-site that are known to be dangerous, security can make an extra patrol. Although many people find this use of private information objectionable, others—particularly parents—find it essential.

Imagine resort security with a large, wide-screen monitor tracking traveler activities. Design a main screen that includes multiple locations, paths and roads, traveler location and status, messages traveling from traveler to traveler, and other features that security should monitor. Create a storyboard that shows an example of a pop-up alert and a menu of options that security might select after an alert. Should you also show security the staff members' locations and status? How about clicking security staff members to send them a message? How about clicking a location to turn up the lights or to close a security gate? Be creative as you think through the design possibilities. You should include four or five screen layouts for the storyboard.

(continued on page 222)

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

(continued from page 221)

Sandia Medical Systems Real-Time Glucose Monitoring

Developers have made these choices regarding format and transmission methods between the cell phone app and server-based software components:

- Glucose-level readings within a normal range will be sent to the server once per hour as XML messages via secure HTTP. Because glucose levels are stored every five minutes by the cell phone app, a normal message to the server will contain 12 glucose levels and time stamps. If more than two glucose levels in a row are outside the normal range, the cell phone app will transmit them to the server immediately.
- Physicians and other medical personnel will initiate direct contact with a patient via text message, voice message, or real-time voice call.
- Changes to alert conditions and updates to software will be sent from the server to a patient's cell phone app as XML messages. The XML messages will be sent immediately after the cell phone app has transmitted glucose levels to the server.

No industry group has yet defined XML tags suitable for the RTGM application. Thus, designers must develop appropriate tags for this application.

If you aren't already familiar with XML, do at least 30 minutes of background research on the Web. Refer to the class diagram in the Chapter 4 RTGM case to determine required data content. Then, design XML tags and message formats suitable to transmit time-stamped glucose levels from the cell phone app to the server and to transmit updated alert conditions from the server to the cell phone app.

If medical personnel choose to send a text message to the patient, how will the server transmit that message? To help answer that question, research Short Message Service (SMS), Enhanced Messaging Service (EMS), and SMS gateways.

Further Resources

Randolph G. Bias and Deborah J. Mayhew, *Cost-Justifying Usability: An Update for the Internet Age* (2nd ed). Morgan Kaufmann, 2005.

Paul C. Brown, *Implementing SOA: Total Architecture in Practice*. Addison-Wesley, 2008.

Patrick Carey, *New Perspectives on Creating Web Pages with HTML, XHTML, and XML* (3rd ed.). Cengage Learning, 2010.

Donald Norman, *The Design of Everyday Things*. Basic Books, 2002.

Janice Redish, *Letting Go of the Words: Writing Web Content that Works*. Morgan Kaufmann, 2007.

Ben Shneiderman, Catherine Plaisant, Maxine Cohen, and Steven Jacobs, *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (5th ed.). Addison Wesley, 2009.

Joel Sklar, *Principles of Web Design* (5th ed.). Cengage Learning, 2012.

Copyright 2011 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.