Chapter 7 – Designing the User and System Interfaces

Table of Contents

- Chapter Overview
- Learning Objectives
- Notes on Opening Case and EOC Cases

Chapter Overview

This chapter addresses two types of system interfaces: user interfaces and system interfaces. User interfaces are those inputs and outputs that involve a user to enter or review information. System inputs and outputs are either those that go directly to another system without any user involvement, or those that only minimally involve users. For example, the printing and mailing of customer bills is considered a system output because even though it is eventually going to a person, the printing, folding, and mailing is primarily done as part of an automated process.

The user interface design is critical to the success of your project. The user sees, touches, hears, and interacts with the system and therefore that is the system to them. The user interface is the system to them, not anything “under the hood.” They don’t care or even want to understand how the database, network, or programming code is done. They just care about what they see, touch, hear and interact with. An analogy I like to give is this, me and my car. I don’t care how it all works, just that it does what I want it to do. I see the instrument panel. I touch the steering wheel, shift controls, knobs, buttons, levers, etc. This is how I interact with my car. I hear message beeps and see messages and lights on the instrument panel. This is how the car communicates with me. I care about how the instrument panel looks, the colors, organization, ease of reading. I care about the touch and feel of the buttons, levers, etc. and the ease of learning and use of these controls. This is the same with designing user interfaces. Also, if your design is done well --- well organized, consistent, uses standards users will learn how to use the system quicker. And… they will have confidence that “under the hood” your other system components are well designed and trust that the system will work correctly.

Some of you may care about “what’s under the hood.” I understand that, but most users do not. Here’s a funny story. Several years ago I bought a Prius. The Prius is a little different than a regular car as far as user interface and safety. Toyota requires a personal training session when you pick up your car. The trainer opens the hood up and says to me “Now when you get home. Your husband is going to check under the hood and want to touch things. Tell him NOT to touch anything with the red or orange tags on it. They can be dangerous. He could be electrocuted.” Now this sounds a little sexist, but really he did say that. And, by the way every new car I bring home, that is the first thing my husband does. He opens up the hood and fiddles with the car. So….. I get home with the Prius and yes that was the first thing my husband did. He opened up the hood. I of course told him not to touch the red and orange tagged items. I had a good laugh. 😊

User-centered design is an important concept that is addressed early in the chapter. There are various metaphors of how a user accesses and interacts with an automated system. Considering these metaphors assists analysts to define and describe the user interface more effectively. Such metaphors as direct manipulation metaphor, desktop metaphor, document metaphor, and dialog metaphor all
describe the user interface in slightly different terms and help in understanding and specifying an effective user interface.

It is important in the detailed design of the user interface to consider good design principles. Such design principles as visibility, affordance, user shortcuts, feedback, closure, error messages, and reversal of actions are all important no matter what type of user interface is being designed. The judicious use of menus and navigation techniques also make the user interface more usable. Often the best way to organize menu items is to group menu items together by actor or user role.

A powerful technique in the design of the user interface is to use storyboards. Storyboards are nothing more than a set of layouts that mimic the sequence of steps for a particular use case or user process.

Special considerations are required for user interfaces that are accessed via the Web. Additional considerations are also needed for small mobile devices.

The chapter concludes by explaining the principles and techniques that are important for the design of system interfaces. Obviously the first step is to identify all the inputs into and the outputs from the system that are produce in an automated or semi-automated fashion. Often interfaces between systems are defined using a standard transmission format such as Extensible Markup Language (XML), which is an HTML like language that is human readable but also can be specified by custom tags.

**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

**Notes on Opening Case and EOC Cases**

**Opening Case**

**Interface Design at Aviation Electronics:** This case is recap of the development and deployment of a manufacturing support system. The case emphasizes the importance of the user experience and the need to both begin early on the design of the user interface and the importance of continual heavy user involvement. In the case the user/client was powerful enough to get the project manager replaced with one who would work extremely closely with the users to develop a total user experience.
End of Chapter Case

On the Spot Courier Services (running case): On the Spot is a small, but growing, courier service that needs to track customers, package pickups, package deliveries, and delivery routes. There are two assignments for this case. First you develop a simple storyboard of a single use case, Web scenario, Request package pickup. Then you create a dialog between the driver and the system for the use case View scheduled pickups/deliveries using current technology and equipment.

User and System Interfaces

Key Terms

- **system interfaces** – inputs or outputs that require minimal human intervention
- **user interfaces** – system interfaces that directly involve a system user

System inputs are normally inputs captured automatically by such special input devices as scanners, electronic messages to or from another system, or transactions captured by another system. System outputs are considered system interfaces if they primarily send messages or information to other systems. Also, included are printed forms in this category because they do not require human interaction with the system. Such things as monthly billing statements that are printed, machine folded, and mailed to customers.

**User interfaces** are inputs and outputs that more directly involve a system user. Their design varies widely depending on such factors as interface purpose, user characteristics, and characteristics of a specific interface device.

Understanding the User Interface

Key Terms

- **user-centered design** – design techniques that embody the view that the user interface is the entire system
- **usability** – degree to which a system is easy to learn and use
- **metaphors** – analogies between features of the user interface and aspects of physical reality that users are familiar with
- **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
- **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

Experienced developers recognize the importance of all aspects of the user interface and recommend using several techniques that place the user at the center of the development process. These techniques
are referred to collectively as user-centered design. User-centered design techniques emphasize three principles:

- Focus early on 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.

The goal of user-centered design is usability, the degree to which a system is easy to learn and use. Ease of learning and ease of use sometimes conflict. For example, a system can be easy to learn but be cumbersome to use all day. Usability is not always easy to achieve.

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 continually returning to the user requirements during each iteration and by evaluating the system after each iteration.

**Metaphors for Human-Computer Interaction (HCI)**

There are many ways to think about human-computer interaction, including metaphors or analogies. Each metaphor provides an analogy to a different HCI concept.

**Direct manipulation:** A metaphor of HCI in which the user interacts directly with objects on the display screen. The desktop metaphor is a direct manipulation approach in which the display screen includes an arrangement of common objects found on a desk.

**Document metaphor:** A metaphor of HCI in which interaction with the computer involves browsing and entering data on electronic documents. Hypertext and hypermedia approaches are based on the document metaphor. Hypertext documents allow the user to click on a link and jump to a different part of the document or to another document. Hypermedia extends the hypertext concepts to include multimedia content such as graphics, video, and audio.

**Dialog metaphor:** A metaphor of HCI in which interacting with the computer is much like carrying on a conversation or dialog. The user asks the computer for something, and the computer responds. The computer might then ask the user for something, and the user responds. The text provides an example that describes a manager and an assistant carrying on a conversation about messages (see Figure 7-3). This dialog would be basically the same if the manager were interacting with a computer that talked, or if the manager were interacting with a computer with menus and dialog boxes. Therefore, designing a natural sequence of interactions, like a dialog, will help ensure usability in a final physical design.

**User-Interface Design Concepts**

**Key Terms**

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

- **affordance** – when the appearance of a specific control suggests its function 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

**Affordance and Visibility**

There are two key principles that ensure good interaction between a person and a machine: visibility and affordance. **Visibility** means that a control should be visible so users know it is available, and that the control should provide immediate feedback to indicate it is responding. **Affordance** means that the appearance of the control should suggest its functionality—that is, the purpose for which the control is used. For example, a button affords clicking, and a scroll bar affords scrolling. Most users are familiar with windows controls, but they may not be familiar with the many new types of controls that are possible on Web sites.

**Consistency**

A consistent-appearing and consistent-functioning interface makes the system easy to learn and use. The way 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.

**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.

**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 provides the user with a sense of confirmation and the feeling that a system is responsive and functioning correctly. Feedback can take several forms including audible sounds, such as clicks or beeps or even voice comments, and visible feedback such as icon changes, color changes, progress meters, etc.

**Dialogs That Yield Closure**

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.

**Error Handling**

A good user interface design anticipates common errors and helps the user to avoid them. When errors
do occur, the user interface needs mechanisms to detect them. When the system does find an error, the error message should state specifically what is wrong and explain how to correct it. 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.

**Easy Reversal of Actions**

Users need to feel that they can explore options and take actions that can be canceled or reversed without difficulty. This is one way that users learn about the system—that is, by experimenting. 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 the system should ask the user to confirm the action and, where possible, delay implementing the action.

**Reducing Short-Term Memory Load**

People have many limitations, and short-term memory is one of the biggest. 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.

**The Transition from Analysis to User-Interface Design**

**Key Terms**

- **storyboarding** – sequence of sketches of the display screen during a dialog

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.

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. In the bottom-up approach, interactive use cases are prioritized, and related dialogs and user interfaces are developed one at a time followed by the menus that tie all the user interface components all together.

**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.

A designer must decide which use cases and user interfaces to include in which menus, which menus are required, and how many menu levels are required. Menus usually contain five to 10 choices in order to avoid overloading the user. Multiple levels of menus helps to keep the number of individual menu items at a workable number.

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.
Dialogs and Storyboards

After identifying all required dialogs, the designers must document the dialogs. 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. Designers can also document dialog designs by writing out how the user and system might interact if they were two people engaged in conversation.

There are many possible implementations of a dialog 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. Figure 7-7 shows a detailed storyboard example.

User-Interface Design

**Key Terms**

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

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. As the range of user-interface technologies has increased, so has the need to create multiple user interfaces.

**Guidelines for Designing Windows and Forms**

*Interface Layout and Formatting:* High-quality interfaces are well laid out, with the fields easily identified and understood. 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.
- Labels and headings—Labels should also be easy to identify and read.
- Distribution and order—Related fields are usually placed next to each other and can be grouped within a box.
• 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.

**Data Entry:** Several types of data-entry controls are widely used in user interfaces, including **Text boxes, List boxes, Combo boxes, Radio buttons,** and **Check boxes.**

### Additional Guidelines for Web Browser User Interfaces

**Consistency:** Consistency is especially critical within Web sites because most sites contain a large number of pages that serve many different purposes and audiences. **Cascading style sheets (CSS)** are a Web page encoding standard, and they enable a Web site designer to specify parts of the site's pages that will always look the same and parts that will vary by task or audience.

**Performance considerations:** Performance considerations are also an important issue in Web applications. 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. Designers of Web-based user interfaces must perform a careful balancing act, providing embedded programming code 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.

**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. Compatibility issues arise for sound and video because there are so many ways in which they can be encoded. Thus, Web site designers must carefully choose which formats and plug-ins will be used.

**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. Providing this kind of support is part of the class of software called **assistive technologies.**

### 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

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. Mobile device user interfaces must avoid detailed textual input whenever possible and must provide
touch controls that are well spaced and easy to locate.

The throughput for 3G networks is usually no more than one-tenth the throughput of a WI-Fi network. 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.

Identifying System Interfaces

**Key Terms**

- **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

The authors define system interfaces broadly as any inputs or outputs with minimal or no human intervention such as the following:

- 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.
- 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.
- 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.

One of the main challenges of EDI is defining the format of the transactions. 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 using **XML tags**. XML files are in human readable form.

Designing System Inputs

**Key Terms**

none

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. 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.

Here are some of the more common devices used to avoid human keystroking:

- Magnetic card strip readers
- Bar-code readers
- 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

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 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. There are three inputs defined with message names and input parameters. These three inputs identify user inputs and system inputs.

Designing System Outputs

Key Terms

- **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

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

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
- 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 third task ensures that the designer evaluates the value of the information to the organization and protects it.

### Designing Reports, Statements, and Turnaround Documents

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

- **Detailed reports**—These contain specific information on business transactions.
- **Summary reports**—These are often used to recap periodic activity.
- **Exception reports**—These provide details or summary information about transactions or operating results that fall outside of a predefined normal range of values.
- **Executive reports**—These are used by high-level managers to assess overall organizational health and performance.

**Internal versus External Outputs:** 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. 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.

**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.

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 ad hoc reporting capabilities so users can design their own reports on the fly. 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.

*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.

**Final Thoughts**

The basic concepts in the chapter are not particularly difficult. The major issue in this chapter is that there is a lot of information and material to cover. User interface design, and system interface design, is not a complex topic, but it is very time consuming. Good interface screens and reports require a tremendous attention to detail and frequently many iterations with the user. The basic concepts taught in this chapter will help you get started, but working with the user to actually develop screens, reports or Web pages is a demanding job.

One area that you might have problems with is identifying the user interface based on the information available in system sequence diagrams and/or activity diagrams. A system sequence diagram is more oriented to individual input and output transactions, where the user interface requires many more concepts and considerations. I recommend that you review this part of the chapter carefully.

The other area that may be new for you is how to create a storyboard. In fact, that is part of the same problem – to take a single use case or message in a use case and expand it to cover various steps of validating input data, navigating around the screen, and finally submitting the data. Review carefully the storyboard example in the chapter. You will be creating a storyboard for the On the Spot Courier system as part of your assignment.