Overview

Unit 12 is organized into eight sections. Following this overview, introduction and objectives, the second section introduces general principles of human-computer (or user) interface design. In section three, you will learn how to design forms, reports and screens and online dialogues. Sections four and five discuss input and output designs, respectively, with their particular focus and requirements. Section six covers the methods for documenting your design. The seventh section demonstrates user-interface design for the continuing case study on the Orient-Pacific Insurance Corporation. Finally, the last section contains a unit summary, list of references, glossary of terms and solutions for the self-tests and activities.
Introduction

Once a systems architecture is adopted and a decision is made to develop a custom, in-house system (rather than purchase a packaged system), detailed technical specifications of system components are required. Systems design is the process of preparing these specifications. This unit focuses on the human-computer interface (other names include user-interface design and socio-technical design).

In user interface design, you concentrate on human interfaces with your information system. You specify external appearances of your system and how users organize themselves to interact effectively with it. You create a physical information system that works for users. Your starting point is your requirements specification and your chosen system architecture. You design the forms, reports, screens, online dialogues and human operations necessary to make your system meet those requirements within that architecture.

A foremost principle of design is that an information system must match the use to which it is put. As a systems designer, you must put yourself in the position of a person who will use your system — you must design your system from the user’s perspective. It is only from this perspective that you can answer such questions as: ‘Will my system really work?’; ‘What will my system look like to a user?’; ‘How will users communicate with my system?’; ‘How will my system communicate with users?’. To produce a successful system, you have to create a ‘friendly interface’ between the system and its users. Friendly systems design requires attention to human factors with regard to the interactions between users and systems.
Objectives

By the end of Unit 12, you should be able to:

1. Demonstrate a knowledge of human factors that have an impact on the development and operation of information systems.
2. Describe strategies for developing interfaces to information systems.
3. Identify the issues surrounding terminal-based and graphical user interface (GUI)-based interface designs.
4. Explain the difference between data capture, data entry and data input.
5. Identify and describe several automatic data collection technologies.
6. Apply human factors to the design of computer inputs.
7. Design internal controls for computer inputs.
8. Use appropriate GUI controls for input screen design.
9. Identify basic types of computer output.
10. Design an appropriate output format based on user requirements.
11. Describe basic principles of good output design.
12. Design forms, screens and reports that effectively meet information and human requirements.
13. Document a system design using a user and technical manuals.
General human-computer interface design

You need to consider some general issues before you begin detailed human-computer interface design.

Using a requirements specification and systems architecture

A requirements specification contains details of data flows and processes in the form of data flow diagrams (DFDs), a data dictionary and process descriptions. A systems architecture specifies general physical characteristics of flows and processes. It describes where they are located, whether they are manual or computer-based, online or batch, and timing considerations. These two documents supply most of the information you need to create your designs.

For example, consider a policy application from a client to Orient-Pacific Insurance Corporation (OPIC). In Diagram 0 of your OPIC DFD in Unit 5 (Figure 2.9), you find a data flow PROPOSAL from an external entity, CLIENT, to a process APPROVE PROPOSAL. You find a full description of this flow in your Data Dictionary. From your systems architecture, you see that this flow comes from an external source and is outside the human-computer boundary. Consequently, it will be a manual form filled in by a CLIENT and sent to OPIC. Data that need to be on this form are specified in your Data Dictionary. Your DFD shows that data from this form are entered into your system online in a process called UNDERWRITE POLICY. You have to design this form so that it is suitable for both a CLIENT to fill in and an operator to enter into your system. You also need to design appropriate screen and computer dialogue. In general:

- Flows from external entities
  These are forms if they are outside the human-computer boundary; for example, PROPOSAL in OPIC. Alternatively, if proposal details are communicated by telephone, data should be entered directly into your system online (as manual recording followed by data entry can introduce transcription errors). As a general rule, data should be captured as close to its source as possible.

- Flows to external entities
  There are reports if they are outside the human-computer boundary — e.g., EXCEPTION REPORT in OPIC. These may be on standard computer printouts or on special forms/stationery — e.g., invoices, statements or cheques.

- Flows that cross your human-computer boundary
  These translate between the bits and bytes that a computer uses and the letters and numbers that humans use. Flows into a system are input either off-line through a data entry operation or optical character
recognition (OCR) system for **batch input**, or **online input** directly through a keyboard and screen. Flows out of a system are output to a printer, usually off-line, or directly to a screen, online.

- Flows within your human-computer boundary

  There are electronic communications between modules, programs, systems or different computers.

One piece of information that may not be available from a requirements specification or systems architecture is a clear picture of user characteristics. If a systems analyst has done a thorough job, notes about users are recorded in the analysis documentation. However, a designer is responsible for ensuring that a design fits its users’ style.

**User characteristics and human factors**

‘User friendly’ is a very common expression, but what does it mean? You may think that a computer that addresses you in a chatty, casual, human manner is friendly. But if a user has a job to do, this type of interface can be frustrating and very counter-productive. A computer is a machine, not a person and it must be designed to reflect the critical function it plays in an organization. ‘User friendly’ means that a system matches its users’ characteristics.

Your first considerations about a user should be their:

- degree of familiarity with a system
- frequency of use of a system.

A highly-trained data entry operator who uses a system for hours every day (e.g., an operator who enters PROPOSALS at OPIC) needs very fast response, minimal instructions and streamlined procedures for error handling. On the other hand, an occasional user such as an OPIC manager making ad hoc enquiries needs guidance, complete instructions and explanations of options.

Your second consideration is the physical environment in which a system operates. A truck driver filling in a log of trips made during a day needs a form that will stand up to very rough treatment. Data entry operators need screens, lighting and chairs that can be adjusted to their size.

You need to take account of the users’ general intellect, personality and social factors. A great difficulty here is that a design must be sufficiently flexible to accommodate changes of staff and changes in work style. One approach is to have a training pack associated with each design to ensure that a user has a prerequisite level of skill before using the design. Perhaps a better way is to design a system that can be accessed at various levels of skill and allows users to choose the level that suits them best.

A final factor is the organization of work flow. Completion of coherent units of work in a small span of time makes for effective information processing and satisfying human work.
Coherent units of work (task analysis)

Single processes on a data flow diagram often become single processes in a physical system. For example, PRODUCE PRODUCT REPORT at OPIC (see Figure 5.12 of Unit 5) is a process that will become a computer program; UNDERWRITE POLICY (see Figure 5.9 of Unit 5) is a process that will involve a person completing a policy. Some functions need to be packaged together because they are done in one place, at the same time, by the same person. For example, a clerk who is at a counter in an OPIC office might deal with a PROPOSAL followed by a PREMIUM, then two general enquiries, followed by another PREMIUM. These are all examples of coherent units of work.

From a computer perspective, a general rule of thumb is that a unit of work is coherent if, when a computer malfunction occurs, a transaction is not partly processed. That is, it is either completed or not there at all; this makes system recovery much easier. A transaction is one instance of a process being completed. For example, an UNDERWRITE POLICY may be processed several times in a day, but each one of these instances is a transaction.

From a human perspective, this unit of work may be too small to be coherent. A data entry operator may enter 100 PROPOSALS an hour. It is difficult to say that each one is a satisfying unit of work. Perhaps an hour’s work can be made coherent by giving the operator different positive feedback on the work or by rotating data entry work with other duties. The study of how people can be organized into satisfactory working relationships is called task analysis.

Decisions about how employees are organized are taken by management. However, work design and management of staff have a great impact on the success of your design. A technically wonderful system that users will not use is a failure.

Criteria for assessing human-computer interface design

You will meet specific criteria for assessing the quality of human-computer interface design later in this unit when you learn about specific form, report, screen and dialogue design. Before that, let’s look at some general quality criteria that your design must meet:

- ease of use
  The characteristics of your user group and your design must match each other.

- meet requirements
  Your design should ensure that each particular document in a system fulfils its purpose.
• be efficient

Your design should achieve its purposes in a way that maximises benefits for minimum costs/efforts. Main benefits to be achieved by your design are user satisfaction and avoidance of errors. Costs are in procurement (pre-printed forms, acquiring software) and operation (staff time to handle a transaction).

• be controllable

A manager must be able to access, monitor and plan operations that your design specifies.

Prototyping a design

The only way of ensuring that your designs are effective is to test them. Engineers build prototypes of cars to test them before going into production. In systems design, prototypes enable users to identify the problems in a design before a system is produced. Previous design approaches relied on drawings of form, report and screen layouts. However, users had a lot of difficulty in seeing exactly how these designs would work in practice. A prototype enables them to actually use a simulation or mock-up of the design. Your prototype should exhibit all the characteristics of a proposed finished product. Fortunately, the advent of ‘visual’ user-interface design tools, as found in software development packages such as MS-Visual Basic, Borland Delphi, and Symantec Visual Cafe, allow very fast prototype designs of user-interface screens.

An advantage to prototyping is that it promotes user involvement in the design. This is a critical success factor. Users are likely to know from experience what practical problems will arise from your design. Also, prototyping allows easy iterations of changes as your design evolves.

Self-test 12.1

1 What are the main dimensions of user characteristics that a systems designer must take into account?

2 List the general human-computer interface quality criteria that your design must meet.
User-interface design

A keyboard, mouse and monitor (or CRT for Cathode Ray Tube) are the most popular user-interface components for a user to communicate with the computer system. There are two aspects to designing a user interface for this situation:

- the overall style of interaction and dialogue between human and computer
- the design of individual screens.

Interaction styles

In information systems, there are two styles of human-computer interaction. These are program-directed and user-directed.

A program-directed style is used in highly structured systems. With this style, programs direct humans through a fixed set of steps to complete a unit of work. Typically, users are presented with a menu of possible transactions that may be executed. They select one of these and the system responds with a screen that looks like a manual form. Users fill in this form using the keyboard. They indicate when they have completed the operation and the system updates files and completes all necessary processing. Screens are fixed in design. The only option open to users is to say which transaction to process next.

Another form of program-directed style is prompting. Here, users are asked one question at a time. After answering a question they are prompted by the next one. This approach is used in systems that are operated by untrained users (e.g., bank automatic telling machines) or when the answer to one question determines which question will be prompted next.

A user-directed style allows users to instigate any functions that are available on the system. A typical form of this style is a command language. Users enter a verb and perhaps some parameters, which the system executes. Consider the following examples in Fig. 12.1.
Figure 12.1  Examples of command prompts and languages

To use a command language effectively, you need to be trained, experienced and a frequent user of the system.

Another form of user-directed style is direct manipulation or the graphical user interface (GUI). This approach allows users to manipulate objects on a screen. For example, the MS-Win95 operating system environment uses an interface in which all programs and files are represented as icons. These may be selected by moving a cursor (on-screen pointer) and clicking a mouse button. Another example of direct manipulation is in using MS-Word (any Windows version) that allows for WYSIWYG or ‘What you see is what you get’ display of the text and images.

Systems addressed by structured systems analysis and design are typically systems where a strict correspondence must be preserved between what is in a computer system and the real information environment of an organization. For these, a program-directed style is the most suitable. A user-directed style is more suitable for office automation, decision support systems and general purpose applications.

In Hong Kong, many large businesses and organizations use large computing systems (e.g., mainframes and mid-range computers), which tend to impose a user-directed style of interface. Applications built on these non-GUI systems tend to be simple, task-oriented and more robust than their GUI counterparts, making it most suitable for mission critical applications.

Reading

‘Graphical user interface styles and considerations’, pp. 577–90 in your text. This reading relates to Objective 2 of this unit.
Dialogue design

In structured systems such as OPIC, dialogue design specifies paths that a user may take through a system. Usually, there will be a main menu screen that controls access to the entire system. For example, a main menu screen for OPIC may look like this (Fig. 12.2):

```
#0 30/01/2001 13:05

Orient Pacific Insurance Corporation

MAIN MENU

Underwriting:
   1 - Record New Policy
   2 - Change Details

Management Information:
   6 - Ad Hoc Enquiries
   7 - Exception Report
   8 - Exposure Report
   9 - Product Report

Premiums:
   3 - Record Premium
   4 - Produce Invoices

Claims:
   5 - Record Claim

Enter Option>_
Select Option and Press <Enter>
F1=Help Esc=Exit OPIC System
```

**Figure 12.2** Sample main menu

This menu allows a user to:

- choose one of the system’s processes or functions
- quit or exit from the system
- get help.

Notice that this menu does not reflect all processes or functions of your system. Different users have different levels of access. For example, an OPIC clerk cannot change product data using MAINTAIN PRODUCT — this is a management function. In fact, this option does not even appear on a menu for this level of user (although it would for a manager). The system identifies users from an identity code and passwords used to gain access to the system.

If a user selects RECORD PREMIUM, another data screen is presented, for example.
A user enters a policy number as indicated and policy details are displayed by the system. These policy data are retrieved from a file on disk. The user then selects a mode of payment and answers ‘Y’ to the ‘transaction complete?’ prompt. The system responds with ‘transaction OK,’ clears data from the screen and leaves the cursor at the ‘enter policy number >’ prompt, ready for the next premium payment. The user may then:

- cancel this transaction
- proceed with the next policy payment
- exit back to the main menu
- quit the system
- get help.

A method of documenting dialogue design is to create a diagram that shows paths to and from each screen. Each screen is represented by a box with three parts:

- a unique identifier, usually drawn from a process description that uses this screen
- its name
- other screens that can be accessed from this screen.

Lines from the bottom of each box show how a user moves from one screen to the next. Part of the OPIC dialogue chart looks like the example in Figure 12.4 on the next page.
You can see which screens are displayed for each option of main menu. Notice that from screen 2.1, a user can exit either to 2.1.1 below or to 2 above or to the main menu (numbered 0).

Screen dialogue specifies how a user can move around available processes, which are dictated by your systems architecture and task analysis.

**Reading**

‘How to design and prototype a user interface,’ pp. 590–97 in your text. This reading discusses another method of charting a user-interface dialogue — the state transition diagram. It relates to Objective 3.
Self-test 12.2

1 Describe two styles of human-computer interaction.

2 What are the differences between a dialogue chart and a state transition diagram?

Screen design

Each screen is specified in a process description (part of your requirements specification). Data to be designed for each screen are specified in data dictionary entries associated with that process and related data flows. You use these specifications to complete your screen design.

Screens are divided into data zones:

- **Heading zone**: This includes a screen identifier (usually the number of the process description), a screen name and current date and time (supplied by the operating system).

- **Body zone**: This is where specific data fields for this process are displayed.

- **Message zone**: This is where all messages to assist users appear.

- **Action zone**: This specifies all actions available to a user at this time.

You can see examples of these zones in the RECORD PREMIUM screen (Figure 12.3). This particular dialogue design provides a facility for users to enquire about client details (using function key 2 as indicated in the action zone) while processing a PREMIUM. If a client comes in to pay but does not know their POLICY NUMBER, this allows a search by name to find POLICY NUMBER, before proceeding with RECORD PREMIUM. An example of a screen design for this search is given in Figure 12.5 on the next page.
After a user has entered a name or part of a name, the RECORD PREMIUM screen is replaced by a list of client names in alphabetical order. The action zone specifies how a user may ‘scroll’ through this list, looking for the correct client by examining name, address or birth date. When the correct client is found, the list number is selected — the search screen disappears and client details are automatically filled in on the RECORD PREMIUM screen.

The following Figure 12.6 is an example of a direct manipulation screen serving the same purpose as Figure 12.5. In this case, the search for clients appears as a pop-up window, rather than as a full screen.

The list of client names appears in a ‘list box’. Arrows on the side of a scroll box allow a user to move through this list looking for the correct client as before. When the correct client is found, the appropriate line is clicked with the mouse — the search screen disappears and the details will be automatically entered on the RECORD PREMIUM screen.
Increasingly, structured information systems are moving towards use of ‘direct manipulation’ or graphical user interfaces. Rather than using a keyboard for all commands, a mouse is used to move a pointer on the screen.

Operating systems for computers today employ GUIs that are modelled after a desktop populated with folders and icons representing documents and applications. Pull-down or pop-up menus allow quick access to a variety of documents and functions. Such designs help to reduce the user’s need to memorize keywords as in the traditional command prompt interfaces. The GUI operating system has been popularized by the Apple Macintosh and has now evolved from personal computing to many aspects of business computing. Thus, the GUI continues to evolve and now is omnipresent — being included in the operating systems of MS-Win3.1/95/NT, Unix X-Windows, as well as Apple Macintosh, and Internet browsers from Netscape and Microsoft.

Figure 12.7 The MS Windows95 graphical user interface

In Figure 12.7 notice the following:

- A control panel on the bottom left of the screen. When the mouse pointer is above one of these options, further menus appear.

- **Windows** or virtual screen display places serve to display various aspects of the system. This could be for input, output, messages, command-line prompts, etc. They can be moved around the desktop and help to organize information as you wish to see it.

- Icons along the top left of the desktop and within folders are visual representations for the logical elements of documents, applications, Internet browsers, disk drives, printers, etc.
Other considerations

There are several other factors that need to be considered in interface design. These include help facilities, system and error messages, and response time.

• Help facilities

Today, most information systems incorporate online help facilities. If in processing a transaction, a user is unsure what to do, pressing a ‘Help’ button or key (often function key 1 (F1) on the keyboard) presents a window with advice. Help information can be organized in a variety of ways. A popular way is with hyper-text, where a menu of general topics appears, which by asking the user to select the topic (and sub-topics) of choice eventually guides the user to the needed information. A second and preferred way is called context sensitive Help. This method brings up help text specific to what a user is doing at the time help is requested. For example, if a user is at the ‘mode of payment >’ prompt, a help screen giving options for this field will appear (rather than a screen for the entire RECORD PREMIUM process). Another approach is to display all possible values for a field automatically in the messages zone of a screen whenever the cursor is on a particular field.

• System and error messages

Messages to a user need to be clear and unambiguous. They also need to contain an instruction as to what the user should do next. Types of messages are:

— informative: no user action is required (for example, ‘transaction complete’)

— warnings: some unusual though acceptable event has occurred (for example, an unusually low premium for a new policy)

— error: an attempt to enter unacceptable data has been made (for example, a policy type that does not exist in a table of valid policy types)

— disaster: an internal error has been detected that needs the attention of a technical systems person (for example, a database integrity failure).

Context sensitive help facilities should be available to give further advice to a user in these cases.

Reading

‘User interface design concepts and guidelines’, pp. 571–75 in your text. This reading discusses guidelines for good user-interface designs. It relates to Objective 3.

• Response time

Stable and appropriate response times are critical to user acceptance. A sub-second response time is necessary for data entry work in which an operator rarely looks at a screen. However, such fast response may be a
problem if a user waits for a response that has already arrived in the time it took to look up from the keyboard. Response times must be appropriate for the operations with which they are associated.

**Self-test 12.3**

1. In a transaction processing system, what typical options does a data screen present to a user?
2. List ‘data zones’ on a screen.
3. What are the types of messages that a computer sends to its user?

**Activity 12.1**

1. Study a PC software package (such as a WP, Spreadsheet, or DBMS). Analyse the human engineering of the user interface and the display screens. If possible, discuss the dialogue and screen with system users. What do they like and dislike about the design?

2. Display properties can be overused and frequently hinder a user’s performance during a terminal session. Cite some examples in which display properties are appropriate and in which display properties may hinder a user’s performance.

3. Match each term to the appropriate definition:
   a. Menu bar
   b. Pull-down menu
   c. Cascading menu
   d. Pop-up menu
   e. Iconic menu

   A. Uses graphic representations for menu options.
   B. Made available once the user selects a choice from a menu bar.
   C. Used to display horizontally across the top of the screen/window a series of choices from which the user can select.
   D. Another type of vertical listing of choices that must be requested by the user.
   E. A menu that must be requested by the user from another higher-level menu.

*Source: Mariga and Morris (1998, p. 94).*
Input design

Data must somehow find its way into the information system. Hence, we will now address the various aspects associated with data entering into our system. We will address the modes and methods of input commonly found in modern information systems.

Data capture, data entry and data input

Data capture is the identification of new data to be input into the system. In a business organization, considerable data flows through its normal operations. These operations, called transactions (sales, reservations, newly manufactured goods, etc.), generate data. Traditionally, we can expect many transactions to use paper-based forms (e.g., typical to many small and medium-sized businesses in Hong Kong). These paper forms are source documents from which data will eventually be entered into a computer system. An example of data capture is when your tutor grades your assignment and pencils in a grade on a student list.

Data entry is the process of translating the above generated source documents into computer-readable format. This is typically done in 'batch' mode where a pile of documents have been generated for the hour/day/week/etc. that will need to be converted to computer-readable form. An example of data entry is when your tutor types the assignment grades into an MS-Excel spreadsheet.

Data input is the actual entry of data into the system (database). Following through with the above example, this would represent a teacher (or Course Coordinator) importing the data from the Excel spreadsheet given by the tutor into the main class grading database.

The above examples demonstrate the three steps by which data are entered into an information system. Modern input methods seek to enter data into the system as soon as they appear, hence, the above three steps are usually combined into one simple step. This can be seen with online registration and ordering systems found on the Internet. In such cases, no paper source document is ever produced. Nor is there any need for human intervention in translating the data to computer-readable format. Note that this objective all but eliminates errors in the step of translating source data into the system.

Reading

‘Input design concepts and guidelines’, pp. 537–46 in your text. This reading discusses data capture and input in greater detail. It relates to Objective 4.
Input forms and screen design

Both paper-based and screen-based input forms should be designed appropriately for all those who will use them. Poorly-designed forms can be costly because they result in people filling them out incorrectly or are simply too awkward to use.

Some guidelines and principles for good practice in form design are given below.

Input form and screen objectives

- ensure form is easy to fill out
- design for accurate completion
- make form as attractive as possible
- ensure that the form meets the purpose for which it was designed.

Detailed design guidelines

- data entry screen should correspond to paper-based form, if one exists
- areas of the screen not needed for data entry should be made inaccessible
- allow users to move freely through the field of the data entry screen
- automatically justify data entries; place alphabetic entries to the left and numeric entries to the right
- allow users to leave the data entry field without filling in all of the data fields
- never require the users to enter information that exists in the database
- display default values in data fields when appropriate.

Automatic data collection technology

Bar-code scanning, magnetic-strip encoding and smart cards have become well-established input technologies for many information systems. We can see these in our everyday life. Renting a video from your video rental store,
the cashier will scan in the bar-code in your member card and the bar-code on the movies you want; the system will automatically enter this information into the database and generate a receipt for the customer. The EZ-Pass tunnel toll collection method is another example of bar-code scanning. If you use an MTR card to get on the subway, the card itself is magnetic (the same technology is used in magnetic-strip encoding) and will read/write to the card as well as its internal database. Smart cards such as the Octopus card that is used for a variety of transportation modes in Hong Kong (MTR, buses, etc.) is one example of a smart card used for data collection.

These automatic data collection technologies serve to replace paper and manual operations associated with data entry, with a very high degree of accuracy and efficiency.

**Internal controls design**

A well-designed input system will include controls (rules for error checking) to see that data entry and input is valid. Common data entry and input errors include typographical mistakes and gaps from incomplete input before submission to the system. The following are established methods for validating data on input:

- **Completeness checks**: These determine whether all required fields have values in them prior to acceptance.

- **Limit and range checks**: These determine that the input value is logically bounded. For example, age should be a positive number from 1 to 125 and the system should prevent any other values from being input.

- **Combination checks**: These verify that some correspondence between two or more fields are logically correct. For example, making sure that an entry for a district in Hong Kong matches the regional area (HK Island, Kowloon and the New Territories).

- **Self-checking digits**: An extra digit (called a ‘check’ digit) on a sequence of digits verifies that the value is valid. This can be seen in the Hong Kong ID sequence, which has a check digit in parenthesis at the end (e.g., C#### # (###)). If any of the seven digits is incorrect, the system will indicate this to the user entering it into the system.

- **Format (picture) checks**: For the input of text (or string) values that take on a particular fixed format, a control can be imposed to enforce this. For example, a Hong Kong ID must start with a letter followed by six digits and have a check digit in parenthesis. Any variations from this would register as an error for re-input.

**Reading**

‘How to design and prototype inputs’, pp. 552–63 in your text. This reading provides additional examples of internal controls. It is related to Objectives 6 and 7.
Self-test 12.5

1. Why are automatic data collection technologies (ADC) becoming popular nowadays?

2. Why are input controls important?

GUI controls for input design

The graphical user interface associated with the direct manipulation approach to user-interface design includes a number of ‘controls’ or design components that allow for the input of data into the system. Controls include ‘text-box’, ‘radio buttons’, ‘check box’, ‘list box’, ‘drop-down list’, ‘combo-box’ and ‘spin-box’, to name a few. Understanding how each of these controls function serves to help the user-interface designer to build better interfaces. We will now list the most popular GUI controls for input in Figure 12.8, over the page.
<table>
<thead>
<tr>
<th>Appearance</th>
<th>Name</th>
<th>Description of use</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Text-box</td>
<td>Used to provide free-form input of text line(s) of information.</td>
</tr>
<tr>
<td>Radio-buttons</td>
<td>Used to allow selection of one of a possible set of choices. Examples are a single choice of a few colours and Boolean value (e.g., Yes/No).</td>
<td></td>
</tr>
<tr>
<td>Check-box</td>
<td>Used to allow any combination of Boolean choices. Particularly useful for configuration purposes. The example shown can be for the options to add to a hamburger.</td>
<td></td>
</tr>
<tr>
<td>List-box</td>
<td>Used to allow for the ability to select (one or possibly more) items from a list. Its role can be similar to radio buttons or check-boxes, but it is particularly useful when the list of displayed items can change in size and value. A scroll bar to the right side may appear if the list is longer than the display box.</td>
<td></td>
</tr>
<tr>
<td>Drop-down list</td>
<td>Initially shown like a text box, if it is clicked on, a pull-down control appears and a boxed list appears below it for selection. It is particularly useful when screen space is limited and the possible values for a field are also limited. Once you have selected the value you want, the appearance returns to what appears as a normal text-box.</td>
<td></td>
</tr>
<tr>
<td>Combo-box</td>
<td>Very similar to the drop-down list, but allows for the creation of new values to the list. That is, whereas a drop-down list is a bounded set, the combo-box allows the user to add a new value or override choices in the list. Again, helpful for saving on screen space.</td>
<td></td>
</tr>
<tr>
<td>Spin-box</td>
<td>This control is used for adjusting values, typically well-suited to numeric (integer) values. Minimizes the amount of space used and allows for mouse-only input of numbers.</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 12.8** Table of GUI input controls
To create a suitable GUI for a system, you will need to match the user-requirements of each data attribute/field with the appropriate GUI controls for each. That is, a good user-interface designer needs to understand the design objectives of the GUI controls, their advantages/disadvantages and to use them appropriately. Please review Figure 12.6 in your text which describes how the above GUI controls are used in developing an effective input screen for a video title.

Some basic guidelines for developing GUIs include:

1. Set popular default values for controls so that the user does not need to input the same value repeatedly.

2. Think in terms of minimizing the number of mouse clicks and/or keyboard typing needed when deciding on different GUI controls/designs.

3. Allow keyboard short-cuts (e.g., use of control keys such as ALT, CTRL, TAB, ESC and F1 to F12) to allow for fast input and/or navigation of the screen. It is known that keyboard entries tend to be faster when performed by experienced users than entries made by mouse input alone.

4. Allow for the ability to undo a previous operation. Such facilities make for a user-friendly interface.

It is expected that each successive generation of GUI for systems development will introduce new controls that will provide greater flexibility and convenience.

Reading

‘GUI controls for input design,’ pp. 546–52 in your text. It discusses in detail when and how to use the variety of GUI controls in an interface. This reading relates to Objective 8.

Activity 12.2

1. Locate a copy of MS-Visual Basic, JAVA, Borland Delphi, or Symantec Visual Cafe. Note that the above vendors may have free demos, trial or beta versions of their software that can be downloaded from the Internet. Identify the above input controls for GUI described here and observe what new input controls are available in the above applications development packages for the user-interface designer.

2. Match the term with the correct definition:

   a. Text-box
   b. Check-box
   c. List-box
d Drop-down list

e Spin-box

f Combo-box

A This control is used when screen space is limited and there are many predefined values.

B A control that requires the user to select data item values from a list with a larger number of possibilities.

C This control requires the user to type the data inside the box.

D A screen-based control that consists of a single-line text-box followed immediately by two small buttons.

E This control combines the capabilities of a text-box and a list-box.

F This control consists of two parts and the user has to provide a yes/no value.


3 A moving company maintains data concerning fuel-tax liability for its fleet of trucks. When truck drivers return from a trip, they submit a journal describing mileage, fuel purchases and fuel consumption for each state travelled through. This data needs to be input daily to maintain records on trucks and fuel stations. For each TRIP JOURNAL data attribute, indicate the GUI control to be used on an input screen.

A TRIP JOURNAL consists of the following entries:

   TRUCK NUMBER - 4 characters
   DRIVER NUMBER - 9 characters
   CO-DRIVER NUMBER - 9 characters
   TRIP NUMBER - 3 characters
   DATE DEPARTED
   DATE RETURNED

   Plus 1 to 20 of the following:
   STATE CODE - 2 characters
   MILES DRIVEN - 5 digits
   FUEL RECEIPT NUMBER - 9 characters
   GALLONS PURCHASED - 3 digits (1 decimal)
   TAXES PAID - 4 digits (2 decimals)
   STATION NAME - 10 characters
   STATION LOCATION - 15 characters
Output design

We have just discussed how to design screens and forms for convenient data input. Now, we need to address the corresponding data output aspects. Specifically, we will cover the types of output used in information systems and the choice of formats suitable for users’ needs. Note that the output represents the most visible result of an information system.

Types of output

Output can be classified under a number of categories. They are listed and defined here:

- **External outputs**: These represent results that are directly generated by the system for a user’s consumption. Examples include bills, checks, membership lists, etc.
- **Turnaround outputs**: These represent results that are generated that will later re-enter the system as an input. Examples include invoices and confirmation statements.
- **Internal outputs**: These are results that are used within the organization that is supported by the information system. Examples include management reports that show detail, summary, and exception data.

The internal output report classifications are as follows:

- **Detailed reports**: These are outputs that show all the details of the data collected. This can include listings of daily transaction data from a retail store.
- **Summary reports**: These are outputs that show summarized information based on detailed data. Such reports are suitable for managers and may include charts and graphs.
- **Exception reports**: These are outputs that show ‘special’ or non-routine transactions or events that occur in the system processes. Typically, information such as a list of customers with overdue payments are representative of an exception report.

**Reading**

‘Output design concepts and guidelines’, pp. 505–13. This reading relates to Objectives 9, 10, 11 and 12 of this unit.
Graphical output formats

Traditional paper-based tabled reports are not the only format for data presentation. More frequently, users ask for more graphical presentations of output. Graphical output formats such as bar charts, pie charts, line charts, scatter charts, and many other variants of these, are included in standard output designs.

Output report and screen objectives

- serve the intended purpose
- suit the user
- deliver the proper quantity of output
- use the most appropriate output media
- provide timely output
- avoid bias in the output presentation.

Design guidelines

- screen title to indicate purpose/function
- screen should display how to exit from it
- consistent alignment of text in display
- use abbreviations only when they are easily understood by the user and significantly shorter than the full text
- arrange items so they are recognisable by the user
- break up long strings of alphanumeric characters into smaller groups of three or four characters
- use colours to highlight important information.

Today’s information systems are increasingly online. As a result, there has been a decrease in the use of paper media (forms and reports). For example, automatic teller machines in banks are replacing deposit, withdrawal and transfer forms; electronic funds transfer is replacing cheques; electronic data interchange (networks) is replacing orders and invoices. At OPIC, notebook and laptop computers are replacing policy proposal forms and premium tables maintained on paper by agents.

Your requirements specification and systems architecture drive your design of forms, reports and manual operations. General design guidelines and those for screen design also apply to forms and report design.
Self-test 12.6

1 Differentiate between medium and format.

2 You are asked to design computer outputs for a small system of an insurance company. What principles should you consider before starting on the output prototype?

3 Describe four alternative formats for presenting information.

4 What are guidelines for good practice in form design?

Activity 12.3

1 For a sampling of the various types of graphical output that are possible, take a trial run through your favourite spreadsheet program (e.g., MS-Excel or Lotus 1-2-3) and/or presentation package (e.g., MS-Powerpoint).

2 The sales manager for Sound Stage Entertainment Club has requested a daily report. This report should describe the nearly 1,000 customer order responses received on a given day. A response is a member decision on whether to accept the record-of-the-month selection, request an alternate selection, request both or request that no selection be sent that month. The report is to be sequenced by MEMBERSHIP NUMBER and CATALOGUE NUMBER. The data repository for the report follows:

The ORDER RESPONSE REPORT consists of the following:

DATE * of the report
PAGE NUMBER

Plus 1 to 1,000 of the following:

MEMBERSHIP NUMBER * 5 digits
MEMBER NAME * which consists of the following:
MEMBER LAST NAME * 15 characters
MEMBER FIRST NAME * 15 characters
MEMBER MIDDLE INITIAL * 1 character
MUSICAL PREFERENCE * possible values include:
Easy Listening, Teen Hits, Classical, Country, Jazz
SELECTION OF MONTH DECISION * possible values include:
Yes, No, None

Plus 1 to 15 of the following:

CATALOGUE NUMBER * 5 digits
MEDIA * possible values include:
Record, Cassette, Compact Disk, Audiophile, 8 Track, Reel

NUMBER OF PURCHASE CREDITS needed * 2 digits
PERIOD AGREEMENT EXPIRES * date membership expires

What type of output was being requested by the sales manager? Prepare an expanded data repository for the output. Prototype the requested output. Verify the output with your instructor (serving as the sales manager or system user). Be sure to include appropriate report headings, edit masks and timing entries.

3 Match the chart with the correct definition:

a  Bar charts
b  Column charts
c  Pie charts
d  Line charts
e  Scatter charts

A  A simple variation of the bar chart that is used when there is a need to show the variation over a period of time or to depict comparisons among items.
B  Used to show trends over a period of time, at even intervals.
C  Used to show individual figures or values at a specific time or to depict comparisons among items.
D  Used to show the relationship or proportion of parts to the whole at a specific point in time.
E  Used to plot the data values for two items to show uneven intervals or clusters of data.

Documenting a human-computer interface design

Your design must be both acceptable to the people who will use your system and the programmers who will construct your system. To achieve these objectives, you need to generate two documents: a user manual describing systems behaviour (for users) and a technical manual specifying internal aspects of your design (for programmers).

In the past, the user manual was often one of the last pieces of documentation produced in a system development. This was because it was not seen as a significant part of a development process. It was often hastily produced and thus of poor quality. Modern structured methods place more importance on the user manual. This is where a system’s external behaviour is specified. The user manual is now treated as a specification of what must be built rather than as documentation of what has been built. You write a user manual from the user’s perspective. You have to put yourself in their place and see a system from their point of view.

Production of a user manual is made much easier by the use of prototyping tools. You can electronically ‘cut and paste’ screen formats from a prototype into your manual. Online help facilities are also a significant source of text and instructions. Directly cutting and pasting from a prototype ensures that your manual and online help are consistent. Consistency is an important characteristic of good documentation.

The technical manual, which specifies a system’s internal design, is written from a programmer’s point of view.

In preparing both these manuals, remember the golden rule — as far as possible, avoid redundancy in your documentation.

User manual

Like all user-interface designs, your user manual depends on the users’ characteristics. For casual, infrequent users, your manual must be very ‘step-by-step’. It must guide users through each task. For skilled, frequent users, your manual can be a reference document that is used when things go wrong.

Your user manual will contain three parts:

- An online user manual that specifies screen dialogue, screen designs, error messages and procedures for using a system.

- An operations manual that describes manual processes and batch processes not covered in the online user manual. This contains form design, report design and job descriptions for users.

- A systems support manual that outlines the assistance available from information systems (operations assistance and maintenance procedures).
Technical manual

A technical manual specifies the internal systems design. It documents aspects such as database design, software design and control design. These are the topics covered in Units 9, 10 and 11. System behaviour is designed from a user point of view — it is primarily an external design. Your technical manual should amplify and explain any internal implications of your external design. At this time, your technical manual is not very extensive as most information is contained in your user manual. Typical contents include the general approach you want programmers to take, detailed dialogue design and an explanation of sources of fields on each screen design (if this is unclear from your user manual). You redraw data flow diagrams if business processes are reorganized. You annotate data flows to show physical details (for example, form numbers). You enhance data dictionary entries with data constraints (used by programmers to check the validity of data as it enters a system).

Self-test 12.7

1 How does a designer effectively document a user-interface design for users?
Summary

You use your requirements specification and systems architecture as a starting point for design. User-interface design is concerned with the design of forms, reports, screens, online dialogues and the human operations necessary to make a system work well. It is an external view of a system.

Your design is documented in a system specification. The user-interface design is primarily described in the user manual. A technical manual is started. When complete, it will also describe other internal aspects of your design; these are developed in Units 9, 10 and 11 of this course.

Once a systems architecture is adopted and a decision is made to develop a custom, in-house system (rather than purchasing a packaged system), detailed technical specifications of system components are required. The process of preparing these is called systems design. Detailed specifications focus on both external and internal perspectives of a proposed system. These perspectives are categorized as: user-interface design, data design and software design.

In the second section of Unit 12, you learned how to specify the external appearance of an information system and how users can organize themselves to interact effectively with it. You should now understand that a systems designer works with and for users, in creating a physical information system. The ruling principle of design is that a product must match the use to which it will be put. As a designer, you must place yourself in the position of the person who will use your design. It is only from this perspective that you can answer the question ‘will it really work?’. You have learned how to design forms, reports, screens, online dialogues and human operations necessary to make a system work well. You also have learned how to document your design in a system Specification that incorporates a user manual and a technical manual.

In the case study section of Unit 12 (over the page), you are now able to practise user-interface design in a ‘real world’ environment.
References

User-interface design


Glossary

This glossary provides brief definitions of the main technical terms you encountered in Unit 12. The definitions given here are offered in addition to, or in place of, those found in your textbook.

**Action zone:** The area of the screen indicating all the possible functions available to the user.

**Bar charts:** Used to show individual figures or values at a specific time or to depict comparisons among items. The categories to be compared are organized vertically, while the values are organized horizontally.

**Batch input/processing:** The oldest and most traditional input method. Source documents or forms are collected and then periodically forwarded to data entry operators, who key the data using a data entry device that translates the data into a machine-readable format.

**Body zone:** The core of the screen or report where data and information is organized for input or output.

**Combination checks:** Determine whether a known relationship between two fields is valid. For instance, if the vehicle make is a Toyota, then the vehicle model must be one of a limited set of values that comprises cars manufactured by Toyota (e.g. Corolla, Corona, Supra, to name a few).

**Completeness checks:** Determine whether all required fields on the input have actually been entered.

**Context sensitive:** Online help text that is specific to what a user is doing at the time help is requested.

**Data capture:** The identification of new data to be input.

**Data entry:** The process of translating the source document into a machine-readable format. That format may be a magnetic disk, an optical mark form, a magnetic tape, or a floppy diskette, to name a few.

**Data input:** The entry of data in a machine-readable format into the computer.

**Data zones:** Areas or divisions with a screen, format or report layout. Typical zones are heading zone, body zone, message zone and action zone.

**Detailed reports:** Present information with little or no filtering or restrictions.

**Dialogue chart:** A diagram that shows paths to and from screens in a system.

**Direct manipulation:** A style of human-computer interface that uses a mouse or light pen to move a pointer on a screen. Icons or pull-down menus are selected to open documents or run software. An alternative to using a keyboard for all commands.

**Exception reports:** Filter data before they are presented to the manager as information.
**External outputs:** Represent results that are directly generated from the system for user consumption.

**Format (picture) checks:** Compare data entered against the known COBOL picture or other language format defined for that data. For instance, the input field may have a picture clause XX999 AA (where X can be a letter or number, 9 must be a number, and A must be a letter). The field ‘A4898DH’ would pass the picture check, but the field ‘A489ID8’ would not.

**Format:** The way the information is displayed on a medium; e.g., columns of numbers.

**Function keys:** (usually labelled F1, F2, and so on) can be used to implement certain common repetitive operations in a user interface (for example, Start, Help, Page Up, Page Down, Exit). These keys can be programmed to perform common functions.

**Graphic output:** The use of a graph or chart to convey information.

**Graphical user interface (GUI):** A user interface design that typically has a metaphor of a ‘desktop’ and includes components of windows, icons, menus and a pointing device (mouse).

**Heading zone:** Typically, the top of the screen or report identifying what it is or what it does.

**Help facilities:** Online text that explains a screen or data. Requested by a user while processing a screen or transaction.

**Internal outputs:** These stay inside the system to support the system’s users and managers.

**Limit and range checks:** Determine whether the input data for each field falls within the legitimate set or range of values defined for that field. For instance, an upper-limit range may be put or pay rate to ensure that no employee is paid at a high rate.

**Line charts:** Used to show trends over a period of time, at even intervals. It is most common to organize the item being charted on the horizontal axis and the measurement along the vertical axis.

**Medium:** What the output information is recorded on, such as paper or a video display device (e.g., monitor).

**Message zone:** An area (or pop-up window) where system messages can be communicated to the user.

**Mouse:** A small hand-sized device that sits on a flat surface near the terminal. It has a small roller ball on the underside. As you move the mouse on the flat surface, it causes the pointer to move across the screen. Buttons on the mouse allow you to select objects or commands to which the cursor has been moved. Alternatives include track-balls, pens and track-points.

**Online input:** The capture of data at its point of origin in the business and the direct inputting of that data to the computer, preferably as soon as possible after the data originates.
Pie charts: Used to show the relationship or proportions of parts to the whole at a specific point in time. As a general rule, a pie chart should be used to show comparisons that involve seven or fewer portions.

Program-directed: A style of human-computer interface in which programs direct humans through a fixed set of steps to complete a unit of work.

Response time: The time a system is able to return control back to the user after an operation has been requested of the computer system.

Scatter charts: Used to plot the data values of two items to show uneven intervals of clusters of data. Various standard statistical techniques can then be applied to determine what degree of correlation exists.

Self-checking digits: Determine data entry errors on primary keys. A check digit is a number or character that is appended to a primary key field. The check digit is calculated by applying a formula, such as Modulus 11, to the actual key (see Figure 12.3). The check digit verifies correct data entry in one of two ways. Some data entry devices can automatically validate data by applying the same formula to the data as it is entered by the data entry clerk. If the check digit entered doesn’t match the check digit calculated, an error is displayed. Alternatively, computer programs can also validate check digits by using readily available subroutines.

Source document: A paper form used to record data that will eventually be input to a computer.

State transition diagram: A method of documenting a dialogue interaction design.

Summary reports: These categorize information for managers who do not want to wade through details.

System specification: Report that documents all details of a system design.

Tabular output: Uses columns of text and numbers. The oldest and most common format for computer outputs. This format presents information as columns.

Task analysis: The study of how people can be organized into satisfactory working relationships.

Technical manual: Part of a system specification that specifies internal aspects of a system design, written from a programmer’s perspective.

Transaction: A unit of work in a process, which typically occurs regularly or repetitively over time.

Turnaround outputs: Those outputs that are typically implemented as a form and eventually re-enter the system as an input.

User-directed: A style of human-computer interface that allows users to instigate any functions that are available on a system.

User manual: Part of a system specification that describes a system’s behaviour, written from a user’s perspective.
**Windows:** A virtual screen or display area found on a graphical user interface where text, graphics and other information can be displayed.

**WYSIWYG:** Acronym for ‘What you see is what you get’ that corresponds to today’s graphical user interfaces of direct manipulation (of text and graphics).

**Zoned output:** Places text and numbers into designated areas of a form or screen.
Answer key for self-tests and activities

Self-test 12.1

1 The dimension of these characteristics are the degree of familiarity and frequency of use that the user has with a system.

2 Quality criteria that your design must meet are ease of use, meeting requirements, efficiency and controllability.

Self-test 12.2

1 A dialogue chart shows the individual screens of a user-interface as it is sub-menued. It also indicates the flow a user will need to go through to locate a particular function. A state transition diagram also indicates a control flow of a user-interface but includes events that direct it to particular screens. Both methods are used for documenting user-interfaces.

2 Two styles of human-computer interaction are:
   • a program-directed style where a system is programmed to process each transaction as a fixed sequence of steps
   • a user-directed style where a user decides what to do next and indicates this with a command or by direct manipulation.

Self-test 12.3

1 Typical options a data screen presents to a user are: continue with the next step of a transaction, cancel this transaction and start another, exit back to main menu, quit this system or get help.

2 Data zones on screens are: heading zone, body zone, message zone and action zone.

3 Types of messages that a computer sends to its user are: informative, warning, error and disaster messages.

4 Context sensitive help explains an error to a user or gives guidance on what to do next in a way that is appropriate to the type of user and the work that the user is trying to do.
**Self-test 12.4**

1. Although technology to support on-line applications costs less than it used to, batch input should not be totally discarded because it still has its value:

   - No matter how cheap and fast on-line processing gets, an online program cannot be nearly as fast as its batch equivalent. Many online programs require some human interaction, and collectively, people are slow, relative to computers.

   - For large-volume transactions, too many working terminals and operators may be needed to meet demand. As the number of online terminals grows, the overall performance of the computer declines.

   - Many inputs naturally occur in batches (e.g., postal delivery) and some may not require immediate attention.

   - Batch processing may be preferable because internal controls are simpler.

**Self-test 12.5**

1. Automatic data collection technologies are becoming popular because they eliminate much of the human intervention associated with input methods. By minimizing or eliminating human interaction, we can decrease the time delay and errors associated with human intervention. This is especially important to businesses operating in a globally competitive environment.

2. Input controls are important because they ensure that the data input to the computer is accurate and that the system is protected against accidental and intentional errors and abuse, including fraud.

**Self-test 12.6**

1. A medium is what the output information is recorded on, such as paper or a video display device.

   Format is the way the information is displayed on a medium; for instance, columns of numbers.

   Therefore, medium and format are essentially two different concepts. Output information needs to be stored on a medium and the way the information appears on the medium depends on the format. The selection of an appropriate medium and format for an output depends on how the output will be used and when it is used.
2 The general principles that should be considered for output design are as follows:

- Computer outputs should be simple to read and interpret.
- The timing of computer outputs is important.
- The distribution of computer outputs must be sufficient to assist all relevant system users.
- The computer outputs must be acceptable to the system users who will receive them.

3 The four alternative formats for presenting information include tabular output, zoned output, graphic output and narrative output.

- **Tabular output**: uses columns of numbers and texts. This format presents information in columns.
- **Zoned output**: places text and numbers in designated areas of a form or screen. It is often used in conjunction with tabular output.
- **Graphic output**: uses graphs or charts to convey information. Some examples include bar charts, column charts, pie charts, line charts and scatter charts.
- **Narrative output**: uses sentences and paragraphs to replace or supplement standard text, numbers and pictures.

4 Guidelines for good practice in form design are: have logical groupings of data, logical flow from one group of data to the next, clear and appropriate instructions and have more than enough space to fill in the required data.

**Self-test 12.7**

1 Effective documentation is user-oriented. A user manual must match the characteristics of users. It may be in a tutorial style for the infrequent or unskilled user; alternatively, for frequent or skilled users, it may be in a reference style. Technical design must be documented in a technical manual suitable for use by systems developers.

**Activity 12.1**

1 Obviously, we can’t provide a key since the question could have too many possible answers. Many available software packages appear to violate most of the common sense principles of dialogue design. One almost gets the impression that new interfaces are developed primarily to distinguish products, not to aid users.

2 Display attributes, such as blinking, highlighting and reverse video can be distracting. They should be used sparingly to call attention to important things, such as the next field to be entered, a message or an instruction.

3 a - C, b - B, c - E, d - D, e - A
Activity 12.2

1. The answer varies with the applications.

2. a - C, b - F, c - B, d - A, e - D, f - E

3. The following answers are only suggestions, student’s answers may vary.
   - Truck Number – Drop-down list
   - Driver Number – Drop-down list
   - Co-driver Number – Drop-down list
   - Trip Number – Drop-down list
   - Date Departed – Spin box
   - Date Returned – Spin box
   - State Code – List box
   - Miles Driven – Text box
   - Fuel Receipt Number – Text box
   - Gallons Purchased – Text box
   - Taxes Paid – Text box
   - Station Name – Text box
   - Station Location – Text box

Activity 12.3

1. The answer will be vary.

2. TYPE: Data Flow     NAME: ORDER RESPONSE REPORT
   EXPLODES TO: Record     NAME: ORDER RESPONSE REPORT
   DESCRIPTION: This report describes one day’s worth of customer responses to automatic order transactions that offer record-of-the-month selections.
   MEDIUM: Paper     OUTPUT TYPE: Internal/Detail
   VOLUME:     NUMBER OF COPIES:
   COPYING METHOD:
   FREQUENCY PREPARED: Daily (reminder: check with the recipient to determine exactly what time of the day the report is needed)
   OUTPUT CHARACTERISTICS: Tabular report on standard stock 11” X 14” paper
   OUTPUT RECIPIENT(S): Sales Manager (reminder: check with the Sales Manager to determine if other users should receive the report)
SPECIAL INSTRUCTIONS:

OUTPUT COMPOSITION:

The ORDER RESPONSE REPORT consists of the following elements:

- DATE * of the report
- PAGE NUMBER
- 1 to 1000 of the following:
  - MEMBERSHIP NUMBER
  - MEMBER NAME, which consists of the following:
    - MEMBER LAST NAME
    - MEMBER FIRST NAME
    - MEMBER MIDDLE INITIAL
  - MUSICAL PREFERENCE
  - SELECTION OF THE MONTH DECISION
- 1 to 15 of the following:
  - CATALOGUE NUMBER
  - MEDIA
- NUMBER OF PURCHASE CREDITS NEEDED
- TIME WHEN AGREEMENT EXPIRES
Notice that our edit masks indicate that data element codes will be decoded on the report. Students frequently fail to accommodate users by translating the codes before making the printout.

3  a - C, b - A, c - D, d - B, e - E
Unit 12

Human-computer interface design
Course Team (Original course)

Project manager: Steve Elliot, OUHK
Course developers: University of Technology, Sydney
Craig McDonald, Consultant
David Wilson, Consultant
Course designer: Shannon Timmers, OUHK

Course Team (Revised course)

Course developers: Ali F. Farhoomand, Consultant
Lester W. Yee, Consultant
Course designer: Linda Chung, OUHK
Course coordinator: John Mak, OUHK
Member: Nigel Leung, OUHK

External course assessor
Professor NV Balasubramanian, City University of Hong Kong

Production
ETPU Publishing Team

Copyright acknowledgements

The following copyright materials are reproduced under the terms of a licence issued by Hong Kong Reprographic Licensing Society. No further copying is permitted.

Information in Activities 12.1–12.3.

Acknowledgement

The OUHK gratefully acknowledges a grant of $100,000,000 from the Hong Kong Government designated for course development.

B329 Systems Analysis and Design was developed in part using resources provided by this generous grant.

Copyright © The Open University of Hong Kong, 1998, 1999.

All rights reserved.
No part of this material may be reproduced in any form by any means without permission in writing from the President, The Open University of Hong Kong.

The Open University of Hong Kong
30 Good Shepherd Street
Ho Man Tin, Kowloon
Hong Kong
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>1</td>
</tr>
<tr>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>Objectives</td>
<td>3</td>
</tr>
<tr>
<td><strong>General human-computer interface design</strong></td>
<td>4</td>
</tr>
<tr>
<td>Using a requirements specification and systems architecture</td>
<td>4</td>
</tr>
<tr>
<td>User characteristics and human factors</td>
<td>5</td>
</tr>
<tr>
<td>Coherent units of work (task analysis)</td>
<td>6</td>
</tr>
<tr>
<td>Criteria for assessing human-computer interface design</td>
<td>6</td>
</tr>
<tr>
<td>Prototyping a design</td>
<td>7</td>
</tr>
<tr>
<td><strong>User-interface design</strong></td>
<td>8</td>
</tr>
<tr>
<td>Interaction styles</td>
<td>8</td>
</tr>
<tr>
<td>Dialogue design</td>
<td>10</td>
</tr>
<tr>
<td>Screen design</td>
<td>13</td>
</tr>
<tr>
<td>Other considerations</td>
<td>16</td>
</tr>
<tr>
<td><strong>Input design</strong></td>
<td>18</td>
</tr>
<tr>
<td>Data capture, data entry and data input</td>
<td>18</td>
</tr>
<tr>
<td>Input forms and screen design</td>
<td>19</td>
</tr>
<tr>
<td>Automatic data collection technology</td>
<td>19</td>
</tr>
<tr>
<td>Internal controls design</td>
<td>20</td>
</tr>
<tr>
<td>GUI controls for input design</td>
<td>21</td>
</tr>
<tr>
<td><strong>Output design</strong></td>
<td>25</td>
</tr>
<tr>
<td>Types of output</td>
<td>25</td>
</tr>
<tr>
<td>Graphical output formats</td>
<td>26</td>
</tr>
<tr>
<td>Output report and screen objectives</td>
<td>26</td>
</tr>
<tr>
<td>Design guidelines</td>
<td>26</td>
</tr>
<tr>
<td><strong>Documenting a human-computer interface design</strong></td>
<td>29</td>
</tr>
<tr>
<td>User manual</td>
<td>29</td>
</tr>
<tr>
<td>Technical manual</td>
<td>30</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>31</td>
</tr>
<tr>
<td><strong>References</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Glossary</strong></td>
<td>33</td>
</tr>
<tr>
<td><strong>Answer key for self-tests and activities</strong></td>
<td>37</td>
</tr>
</tbody>
</table>