9. User Interfaces

9.1 How to Do User Interface Design 9-4
9.1.1 Analyze Users 9-4
9.1.2 Groups and Scenarios 9-5
9.1.3 Command Hierarchy 9-6
9.1.4 Interaction Dialog Planning 9-7
9.1.5 Screen Planning 9-9
9.1.6 User Interface Look-and-Feel 9-10
9.2 Reports 9-13
9.3 Other Considerations 9-14
9.3.1 Help 9-14
9.3.2 Prototyping 9-14
9.3.3 Miscellaneous 9-15
9.4 Foreign Countries and Languages 9-17
9.5 More Information on Forms, Screens, and Reports 9-20
9.5.1 Introduction 9-20
9.6 Form design 9-22
9.6.1 Form Design Criteria 9-22
9.6.2 Components of Forms 9-23
9.6.3 Styles of Forms 9-23
9.6.4 Physical Considerations in Form Design 9-24
9.7 Screen Design 9-26
9.7.1 Screen Design Criteria 9-26
9.7.2 Components of Screens 9-29
9.7.3 Styles of Screens 9-31
9.7.4 Standards 9-33
9.7.5 Physical Considerations in Screen Design 9-33
9.8 Report Design 9-35
9.8.1 Report Design Criteria 9-35
9.8.2 Components of Reports 9-36
9.8.3 Styles of Reports 9-38
9.8.4 Physical Considerations in Report Design 9-40
9.9 Design by Prototyping 9-44
9.10 Testing Screen Designs 9-46
9.11 CASE Tools for Input and Output Design 9-47
9.12 Summary 9-48
9.13 References 9-49
9.14 Appendix 9A - Screen and Report Design Forms 9-50
9.15 Appendix 9B - Examples of Reports 9-51

Optional Readings: Chapter 11 of [Montgomery94]

This chapter deals with how an application communicates with users and other objects in the real world. Remember that computers only exist to generate outputs. They have no other purpose! So it is very important for computers to generate organized, usable outputs in a timely fashion. On the other hand, it has also been realized that computers are by nature quite unfriendly, unless the interaction mechanism of the human-computer interface is well tailored to both human nature and to the particular application. As most of a human's interaction with computers will be to enter input into them, and to enter requests for outputs into them, considerable effort should be put into making the user interface easy to use and resistant to errors.

Thus the 'user interface' (UI) as it is called, has become a very important part of design. In fact, this stage is called External Design. As highly interactive user interfaces become more and more prevalent and sophisticated (with hypertext and context sensitive help, or animation), it become very important that external design should be one of the first design steps. It should be done even before architectural design, as often the architecture is somewhat dictated by the nature chosen for the UI. The external design is best documented by either a rapid prototype quickly put together for evaluation, or by a draft user manual.

The typical outputs from an application are:

- printed reports
- user query output screens
- outputs to devices (e.g. audio speakers, or lumber mill control)
- files which can be accessed by (or sent to) other applications
- responses to Remote Procedure Calls (RPCs). RPCs are the **primary vehicle** for the implementation of distributed, client-server applications.

The typical inputs to an application needed to generate the above outputs are:

- data inputs from users
- control inputs from users
- data and control inputs from devices (i.e. input sensors)
- files created by other applications
- information obtained via a network (e.g. RPC calls) or modem.

9.1 How to Do User Interface Design

In this section I will try and outline my views of the issues and steps of External Design.

9.1.1 Analyze Users

One of the first steps is to analyze and classify the users. In fact, you will likely have a number of different classes of user sophistication.

- several different classes of user may use the same application (e.g. secretaries, programmers, and company presidents all may need to use a wordprocessor).
- different types of users may use different parts of an application (e.g. clerks do inventory management, bookkeepers total inventory, accountants use inventory in determining financial status).

Generally you must classify users along two dimensions: novice vs. advanced computer users, and clerical vs. supervisory/management.

	clerical	supervisory/management
novice	new clerk or secretary	president
expert	bookkeeper	office manager or engineer

The thing about supervisory and management personnel is that they usually understand much of the operation of the company and/or the application.

Clerical workers are generally only involved in, and often only understand their small portion of the operation of the company or of a large computer application.

With novice users, it is often necessary to not even show them large, sophisticated menus of options, lest they become overwhelmed. The application must sense user classifications and hide overwhelming feature sets from them. In one company I once worked for, they decided not to change wordprocessors brands for fear 2 of the 12 secretaries would quit. Things are not as bad now as all secretaries realize they have to be wordprocessor confident, but UI's for machinists and automobile service station personnel must be simple.

One last thing to consider is security. Often it is necessary to classify users by which tables they should be allowed to read, and which they are able to add, update, or delete records from. Often some users are not even allowed read some confidential data (e.g. every employee's salary or performance review).

9.1.2 Groups and Scenarios

Generally, users have to do a number of interaction tasks with a computer. Often one is intimately related to another, or follows another sequentially. It is important to form a list of command events (from a user or input sensor) that the system must be able to respond to. Typically there can be a large number of these. In order to handle this complexity, these must be grouped together so that they can be represented as higher level menu entries. Use of a hierarchical menu system is the only way to have relatively short access to choose one of many leaf operations (just like tree data structure access).

Groups can be formed by collecting commands which:

- operate on the same data (e.g. add, query, modify, delete ferries)
- follow one another in sequence (either immediate sequence or lifecycle sequence of some data or association).
- or some other reasonable clustering criteria.

Then you must ask yourself about **use case scenarios**. These are typical **sequential patterns** of use of one command after another. e.g.

- Is it common to re-open the last wordprocessor file you were working on.
- Is is common to enter a product order right after entering a new customer?

In the first case, it means you have to design the application to have a whole extra disk file just to keep the name of the last file edited (and maybe some other preferences).

In the second case, the Enter Order command should be put on the same sub-menu as the Enter Customer. That way, when you return from entering a customer, and the menu system drops you back to the sub-menu from whence last chose a command, Enter Order is right there ready to be chosen. Unfortunately, this is not always possible. There may be good reasons to locate the Enter Order on the Sales menu. In this latter case, at the end of customer entry, the user should be given the option (button to click, or prompt for "do you want to enter an order for this new customer"). Or alternatively, Enter Order should be placed on both the Sales menu, and on the same sub-menu as Enter Customer! In this case, the tree is no longer strictly hierarchical, but instead is a 'directed graph'.

9.1.3 Command Hierarchy

So you must develop a command hierarchy, most likely in tree form but possibly with some repeat entries or graph-like short cuts.

- Most wordprocessors with automatic section numbering such as used in this document can be used to edit a textural representation of a menu tree. Insertion and movement of various commands causes automatic re-numbering of this written prototype of the menu hierarchy.
- Try and keep the number of items in each menu list (main or sub-menus) to less that 7 as psychological tests have shown people have difficulty handling more that 7 things at once.
- Put items of a particular menu list in sequential order of use, if any kind of order exists (this is usually better than alphabetical). Or put items most frequently used near the top of individual menu lists.
- Evaluate which menu items need to have security restrictions in your applications. Maybe all operations requiring security should be grouped in one sub-menu?
- Try to keep the tree balanced so that some branches are not a lot longer than others. The only exception might be to try and migrate extremely high use commands to the top hierarchical level or two. Better yet, add a toolbar of icons across the top of the window,

where the most frequently used operations are represented by directly clickable icons. Or alternatively, annotate across the bottom of the screen what short cuts to operations are available for each function key (e.g. F1, F2, etc.).

- Try to keep the menu hierarchy short. First, using a menu bar across the top of a window allows direct access to all items on the main menu (i.e. you don't first have to do something to make the main menu visible). And second, try to cut your menu level depth off at three. Often, lower level menu items needn't invoke commands but can instead bring up an interaction screen/window, on which there are a number of fields that can be entered/changed.

9.1.4 Interaction Dialog Planning

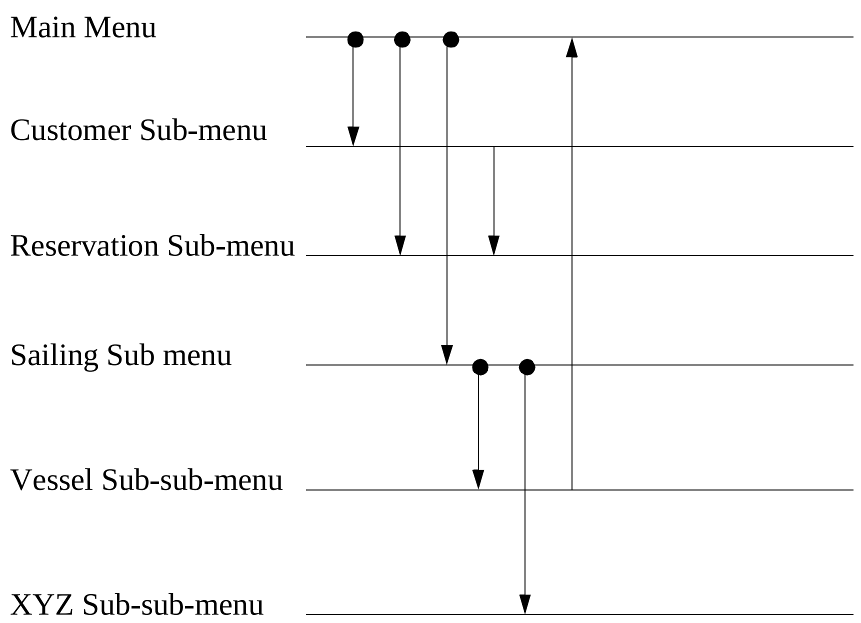
Sometimes it is hard to plan the multi-connected graph that represents the allowable transfers between parts of a menu system. For this, a dialog diagram can be helpful.

Given a menu hierarchy such as:

Main Menu

1. Customer
2. Reservation
3. Sailing
   3.1 Vessel
   3.2 XYZ

a dialog diagram might look like that shown below:

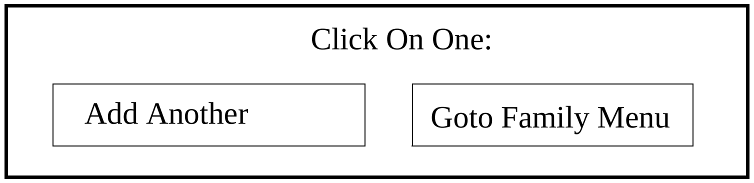


It shows the allowable transfers from menu to menu. The state of the menu system (i.e. what menu is showing) is shown as a horizontal line. Transfers are shown as vertical arrows. Arrows with the black dots are a shorthand for indicating the reverse transfer is also provided, but not shown. There is no sense of ordering when an interaction diagram is used to document the possible menu transfers.

Notice there is a short-cut from the Customer-Sub-menu to the Reservation Sub-Menu. This prevents you from having to go up to the main and then down to get to the Reservation Sub-menu. Also shown is a direct return from the Vessel Sub-sub-menu to the main menu. These extra arrows indicate that the end of a scenario you either:

a) Get a menu where one of the options is to transfer to some other menu not directly adjacent in a hierarchical menu tree. i.e. one sub-menu or command appears on two different other menus.

b) You get a choice:
   - either you are asked, for example, "do you want to add a family?"
   - or you are presented with a screen which allows you to click on a selection which determines your next menu location. e.g.



c) The data you just entered *determines* that you will immediately travel to another sub-menu. e.g. an under-age dental patient which does not have a corresponding, financially-responsible adult family member in the database yet. Another example might be library book selected might bring up a different menu of appropriate options for a book depending on whether it was in stacks, in reserves, or on loan (e.g. recall, or just notify me when returned).

9.1.5 Screen Planning

Once a user has selected an operation, in most information systems they will be presented with a screen.

- The screen may simply take over an entire character terminal display.
- Or in a graphical user interface (GUI), a screen (also called a dialog window) may be a sub-window that pops up.

A '**screen**' is what looks like an electronic form. Various fields are annotated with read-only titles. And the actual cursor is located usually in the first writable field. The user enters data in the first field, then 'tabs' to the next one. Once enough data has been entered either the operation starts, or if necessary another window pops up. Or, if doing an inquiry, once half the data entered (enough for a search on one of the alternate keys), you may hit the 'enter' or 'send' key, and the operation will retrieve the record and fill in the rest of the data. By displaying the

screen representation of a whole record, rather than just the primary key fields, the user was allowed to pick any combination of fields to enter before asking for a search.

Screen-oriented I/O is widely used in data processing (DP) applications. In fact, terminal manufacturers like IBM make terminals that are much smarter than dumb ASCII terminals, but not graphical. In general, they are tailored to presenting screens with field titles which are not editable, and fields where the user can enter data. When the user presses 'enter' or 'send', the terminal only sends the fields the user typed in (to reduce data transmission load/cost/delay), as the receiving computer already know what the fixed fields are.

Anyway, normally any given task, such as entering a new employee into a personnel database, may require the filling in of a number of screens (personal name/address/birthdate, family info, tax and benefits info, department/supervisor info, etc.). This sequence, the exact look of each screen, and type, length/range, unit of measure, precision (decimal places of display), and accuracy (plus or minus 0.03) of every field should be planned. See Appendix 9A.

If data must be entered from printed forms, the data entry order should be the same as on the forms. Either that, or re-design the forms to look like the screens.

In addition to the interaction sequence for each menu leaf command, the errors/exceptions that can occur should be determined at every step along the way. These are best shown in a draft copy of the user manual.

9.1.6 User Interface Look-and-Feel

Note that the above discussion on UI design was basically independent of the look and feel of the UI. By look and feel, I mean whether the menus are accessed by the left, right, or only mouse button, and whether the scroll bars are on the left or right.

The look and feel can be quite varied from one system to another. In fact the whole nature of the interface can be different.

- Most of us are familiar with a noun-verb interface. First we select the object on the screen we want to interact with, then we select the operation we want to do to the object. A good example is in a work processor, where we highlight a piece of text, then ask the system to either cut, copy, underline, or change the font size of it.
- Alternately, many drawing programs need a more flexible mechanism. They often allow you to choose the tool (draw, open, move, erase) you need, then use that tool to click on any (or importantly, many different) objects on the screen. The drawing tool within Framemaker which I am using to create these lecture notes is verb-noun oriented, but quickly reverts to noun-verb after most uses. On the other hand, the Objectbench CASE tool used in Cmpt 370 in (94-2) is fully verb-noun oriented, and takes considerable getting used to. Verb-noun interfaces are not bad, but Objectbench's is poorly planned as to simply exit, you have to select the 'draw' tool, then click on 'sub-system', then choose exit. This doesn't make a lot of sense, and indicates a poorly designed UI. Also, clicking on 'sub-system' does different things depending on which tool you are using. This makes the system's response terribly context or mode-dependent.

You should probably attempt to use a user interface standard that already exists. Your choice may depend on your computer (Sun/Mac/NeXT/IBM-PC), operating system (Unix, OS/2, Windows), and your specific needs. One advantage of NeXTStep and Windows NT, is that they maintain separate input event queues for each application. This allows that if one application is stuck, you can change to another and continue to use it (possibly kill the first).

If you are designing the cockpit of a new jet aircraft, you may need to design your own UI. But if not, a standard interface 'look and feel' will have the following advantages:

- often is already familiar to users
- cheaper than developing your own
- purchased UI system library will have fewer bugs.

If you have a choice, evaluate the various standard UIs carefully to get the best match between the users and the application. Even for a given UI standard, there are likely two or more different UI libraries from different vendors you could choose to buy. One might have built in functions to easily create tool bars, say, while another might not.

9.2 Reports

Reports are normally generated in response to a command. Reports are often multi-line outputs of data. Generally the user will have to select the data to be output (e.g. sales for Jan thru June, or all sales by a particular salesman). The selection criteria is entered on a screen and then the report is requested.

To create a report often requires selecting tables from a data base, specifying a join, then selecting columns with a projection, selecting rows which fit a search criterion (e.g. sales orders over $50,000 in total value), calculate some derived attributes, sort the resultant table into the presentation order, sums some columns, and print the report. Obviously, specification of such a report requires a considerable amount of screen entry.

Report (and screen) formats are best planned on a planning sheet (which is a matrix of character positions). See Appendix 9A. Each space on a planning sheet represents a possible character location on a report or screen. These planning sheets assume every character is the same width and height. In particular, reports where iiiii's are narrower than mmmmm's, are rarely used, even though modern printers are capable of using proportional-spaced fonts. Many reports need to be arranged in columns anyway, and fixed-space fonts such as `courier` are best for this.

Long reports often have 3 sections:
- a header describing the type, date, and selection criteria of the report
- body pages, comprised of page number and column headers on each page, plus one line (or occasionally several lines) for each object being reported.
- a summary, often consisting of some column totals, the total number of object reported, and possibly some other calculations like column averages.

For a great set of examples of reports, see Appendix 9B of these lecture notes (which is taken from the B.C. Ferry Corporation Request for Proposal that has been put on reserve in the library for Cmpt 370).

9.3 Other Considerations

9.3.1 Help

Users often need on-line help. Here are different types of help designs you can consider:

- Command help, where if you know the command you want help with, you can ask for it. Even unix gurus need the 'man' pages help feature. PC-DOS has a similar feature.
- A help index that subjects can be looked up in is good (you don't have to know the exact name of the command to look it up in the man pages. e.g. passwd).
- Hypertext help, where the help text has markers which can be selected to take you directly to information about the marked subject.
- Context-sensitive help, where when the user simply asks for help, and gets an message explaining the situation he is in, and his options as to how to proceed, even if he doesn't know what situation he is in.
- Prompt help, where at each step along the way, the user is prompted for exactly what to type or do next.
- User Manual.
- Undo feature.

Usually a system will use at least three of the above methods to provide help.

9.3.2 Prototyping

User interfaces provide a grand opportunity for prototyping in order to get customer response to the proposed UI design, before it is properly implemented. In particular, consider prototyping if:

- the proposed interface is highly interactive or manipulative (e.g. rubber-banding),

- has many classes of users,
- has vague UI or application requirements,
- the developer (i.e. YOU) are unfamiliar with the application,
- or the project is otherwise risky.

9.3.3 Miscellaneous

a) Choose the right user devices for the job. You shouldn't necessarily use a laser printer for huge inventory reports. And there are lots of great input devices you could specify for your system:
   - magnetic strip
   - optical card reading (like course evaluations)
   - bar codes readers (you can print them too!)
   - image scanners, fax receivers, TV frame grabbers
   - audio data, or voice commands
   - Electronic Document Interchange (instead of paper) over net/modem.
   - from files you can get other applications to create.

b) Strive for consistency. The screens you enter data on should look similar to each other (e.g. date always in top right corner), and the different reports should have some consistency. An inconsistency that you may encountered with the ObjectBench CASE tool is to end entry of a text line, you can use a carriage return, but to end a text box, you must click outside of the text box window. In the latter case, the box disappears and the user is not sure whether his text was captured or discarded. First there is inconsistency, and second there is no feedback.

c) Have the user do as few steps as possible

d) There should be no 'dead air'. Always provide feedback so the user doesn't wonder if the computer has frozen.

e) Don't require the user to remember any thing/data from one screen to the next.

f) Make the screen legible. Users will need to use it for hours. Make the fonts large enough, and the icons obvious. If your application will have to run on various GUI screen resolutions, provide install features that allows the setting of the font size.

g) Use blinking and highlighting sparingly. Use blinking only for urgent things, as it is very annoying. Use highlighting to 'focus' the user on the important next stuff.

h) Don't use audio unless really needed, as it really annoys neighbouring users.

i) Provide escapes. It drives users crazy if they can't back out (for instance, quit if they can't remember their password and CNTL-C doesn't cause a quit)

j) Provide defaults for frequently-entered values.

k) If possible use scrolling lists rather than user-keyed entry of data. This is easier and faster for users, and tends to keep referential integrity as users can only select from among existing values for foreign keys. Unfortunately, this really puts a burden on the computer and network, and may slow things down a lot. It is impractical if the lists are very long (e.g. Air Canada's reservations).

9.4 Foreign Countries and Languages

Most of the material in the rest of Section 9 are derived from Cmpt 370 lecture notes by Paul Franklin based on Chapter 8 of a previous 370 textbook [Fertuck 92]. Another good reference, with chapters called "Designing Effective Output", "Designing Effective Input", "Designing the User Interface", and "Designing Accurate Data Entry Procedures", is [Kendall 92].

Some applications have to be available to users of more than one language. Various factors should be taken into consideration:

Formatting Rules — formats of common data items like dates, numbers and money vary from country to country.

1) **Dates and times**

- Common to use MM-DD-YY in North America.
- Common to use DD-MM-YY in Europe.
- So what does 12-1-92 mean — Jan 12th or Dec. 1st?
- One solution: use YY-MM-DD.
- Unwise to use alphabetic forms like Dec. 12th.
- Local configuration can specify how stored date is to be displayed.
- Time may be specified with 24-hour clock in some countries.

2) **Numbers**

- In U.S. and Canada, a period is used to separate integers from fractions.
- In U.S. and Canada, a comma is used to separate each group of three digits in long numbers like 1,234,567.89.
- In Europe, this may be 1.234.567,89.
- Configuration options can allow different displays.
- In the U.S., France and Canada, a billion is 1,000,000,000.
- In Britain and Germany, a billion is 1,000,000,000,000.

3) **Currency**

- Symbols are different in each country.
- They may consist of more than one character.
- Local configuration options needed.

4) **Units of Measure**

- Metric units more common, but U.S. still avoids metric.
- Imperial gallon is 160 ounces, American gallon is 128 ounces.
- British long ton is 1,016 kg.
- U.S. short ton is 907 kg. = 2000 pounds
- Metric ton is 1,000 kg.

Languages — place different requirements on the designer.

1) **Word length**

- *Text may expand in other languages*!
- Tables and screens may need extra space for other language versions.

2) **Word order**

- Command languages may seem unnatural when translated.
- Subject-verb-object order different in other languages.
- Arabic and Hebrew are keyed from right to left, except for numbers.
- Cursor must move in correct direction for type of input.
- Tables may then be filled from right to left — key on right, data to left.

3) **Sorting**

- If software uses internal EBCDIC or ASCII codes, sorting may not work correctly.
- Characters like a, â, ä, à, å, A, Ä, Å may not be sorted before b and B.
- Most DBMS use ASCII or EBCDIC internal codes for sorting.
- May need to develop some sorting algorithms.

4) **Multilingual documents**:

- Several countries have more than one national language.
- Applications may need to support more than one language on a terminal.

9.5 More Information on Forms, Screens, and Reports

In information systems, most of this communication has traditionally occurred through forms, screens and reports.

- Forms are used to help collect and record data for the system from people who do not have access to the particular CPU(e.g. the enterprise's customers).
- Screens ar used to enter data interactively at the keyboard.
- Reports and screens are used to deliver output to the users.

This chapter will show you how to

- Specify the characteristics of good form, screen and report design.
- Identify the different components of forms, screens and reports.
- Identify the different styles of forms, screens and reports.
- Assess the impact of physical considerations on the design of forms, screens and reports.
- Use prototyping to design screens.
- Adjust the design of screens for different countries and languages.

9.5.1 Introduction

- Computer is useful only if it is able to communicate with people and machines that use it.
- **Communication** is the **exchange** of messages with some **permissible content** within some **context** to produce a **conversation**.
- **Exchanges** are usually commands or data sent to the computer, followed by a reply. Everything sent to the computer should be acknowledged somehow.

- **Context** is the environment that gives meaning to the exchange. The same action or data entry may in a different context cause a different system response.
- **Permissible content** is the set of messages that are allowed in a particular context.
- **Conversation** is a series of exchanges linked together to assist in performing some process.
- Most common input method is to enter data on forms, then key data into an input screen.
- Most common output is a printed report or report on a screen.
- Three new trends are changing importance of forms, screens and reports:
  - **Interactive data entry** allows users to key data directly (i.e. as it is acquired).
  - **Interactive data inquiry** allows users to produce ad hoc reports whenever they need them.
  - Also a strong move to eliminate remaining paper communications by **Electronic Data Interchange (EDI)**.
- Note that if users are going to have to enter long part numbers or other identifying designators, the IDs are often designed to include checksums. For instance, if entry clerks will have to enter thousands of 5 digit numbers per month, it would pay to add one extra 'checksum' digit to each number. e.g. 12312 would be known externally to all humans as 123129, where the 9 = 1+2+3+1+2. Thus the computer can subsequently catch any typing mistakes by the data entry clerks! See the chapter titled "Designing Accurate Data Entry Procedures" in [Kendall 92]
- Many reports are only needed temporarily, or "just-in-case", and thus long term paper storage is not necessary.

9.6 Form design

- Forms are not always needed.
  - Data may be captured directly on a screen or by some device such as a bar code reader. Federal express used forms and captures some stuff electronically, and uses bar codes to connect the two.
- Forms are needed for input from people who do not have access to a workstation.
- They are needed for legally important documents.
- They are needed for turnaround documents. For example, bills/ invoices are used to prompt other people into action. Often they are returned with previously empty fields filled in with added information.
- Ideally, forms and screens should be designed together, since the contents of the forms must be typed into the screen in screen order.
- If existing forms are not well-designed, now is the time to redesign them.

9.6.1 Form Design Criteria

A well designed form must satisfy the following criteria:

- Must be definitive.
- Must be consistent with related forms and screens.
- Consistent forms make it easy for users and data clerks to learn.
- **Completeness**: should gather all data at the source so there is no need to transcribe data to other forms.
- Cost of handling usually much more than cost of design and printing.
- **Organization**: parts of the form should be arranged in an efficient order with visual separation between the major sub-sections.
- May be a compromise between sequence of entry, frequency of use, function and importance of data.

- First data available, most frequently used data, and/or most important data should be at the beginning.
- All data about a single function should be grouped together.
- Irrelevant information should be minimized.

9.6.2 Components of Forms

Basic elements of forms are

- **Descriptive information —** provides help on how to identify, fill in or use the form. In particular, should direct writer which portions should be filled out for each situation. For check boxes, should indicate whether either or both of two boxes can be ticked.
- **Intentional information —** is the labels and headings to indicate what goes in the blanks.
- **Extensional information —** which is the spaces to be filled in.
- Spaces can be blanks in which numbers or text are entered.
- Spaces can be check boxes to be selected.
- Different styles and sizes of type may be used to distinguish titles, headings and captions.
- Areas reserved for data may be a different color.

In forms analysis, forms are divided into data areas.

9.6.3 Styles of Forms

- There are three common styles of forms.
- They differ in the relationship between the intentional captions and the extensional blanks.
- Captions can be above, preceding or within the blanks.

1) Captions Within Blanks

Last Name, First Name:

2) Captions Above Blanks

Fill in Last Name, First Name:

3) Captions Precede Blanks

Fill in Last Name, First Name: ______________________________

- Captions before blanks often used when original form is created by a typewriter or word processor.
- In this case, all text may be in same size and style of font, which makes it not as clear.

9.6.4 Physical Considerations in Form Design

Several physical characteristics of a form have to be specified:

- **Size —** standard sizes are 8 1/2 by 11 and 8 1/2 by 14 inches (legal). Smaller forms may be half a standard size.
- **Color —** should be light to make written material readable. Type should be in black ink.
- **Format —** 8 1/2 inch dimension should be horizontal (portrait format). This fits computer screen width, printers, file folders and document feeders.
- **Type —** should be a clear simple font. Captions and headings should be large enough to read clearly. Boldface can be used to emphasize.

- **Spacing** — vertical spacing between lines is 1/6th of an inch so that a typewriter can be used to fill in the form.
- **Position** — titles should be centered, and section headings positioned sensibly.
- **Preprinted sequence numbers** — are used when forms have to be accounted for. Sometimes sequence numbers are bar coded.

Not all input comes from forms keyed in at screens — other devices may be used:

- **Optical scanners** — read printed or typed characters from a page. Some can even read hand-printed characters.
- **Bar code scanners** — laser scanners in supermarkets read the Universal Product Code (UPC) on products. Can be used in other applications.
- **MICR** — magnetic ink character recognition is a system that reads specially shaped alphabetic characters printed with magnetic ink. Most commonly used for sorting bank cheques.
- **Magnetic stripes** — as on the back of a credit card, can be read by a reader somewhat like a tape recorder.
- **Telephone keypads** — touchtone keypads can be used to send numeric data. Customers can transmit credit card numbers, authorization codes, etc.
- **Voice** — voice recognition devices are improving. Can be used to collect data from people whose hands are busy.
- **Electronic instruments** — data may be sent automatically from sensors.
- **Operators** — telephone operators may receive data from customers and key them without using intermediate forms.
- **Files** — data suppliers may create files and transmit them directly.

All of these methods can reduce the cost and number of errors.

9.7 Screen Design

Screen communication is in the form of active dialog.

- Dialog has semantic and syntactic rules.
- Semantic rules determine the meaning of elements in the interface.
- Syntactic rules determine the sequence and order of elements.
- Data input is often done by transcribing information from a paper form to a form on the screen.
- Updating is like correcting an entry on a form.
- Deletion is like destroying a form.
- Inquiry consists of asking the computer to retrieve one or more forms for inspection.
- Not all screens correspond to forms.

9.7.1 Screen Design Criteria

- Good screen design reduces interface complexity **as perceived by the user**.
- Several features make screens easier to use.
- These features are generally found in graphical user interfaces (GUI) such as
- Macintosh.
- Windows.
- Presentation Manager (OS/2).
- X-Windows Motif.

We’ll look at these features individually:

1) **Concreteness**:

- People find it easier to work with a concrete object, rather than an abstract concept.
- Easier to learn icons than abstract commands.

- Icon is the picture of an object on the screen.
- Icon is mentally associated with a tool that performs a specific function.

2) **Visibility**:

- Users find it easier to operate an interface where they can see the commands, rather than have to memorize them.
- People have relatively poor memories.
- Commands, function keys used, options and helps should be available on the screen.
- Easier to select from a list of options.

3) **Modifiability**:

- Easier to copy a template or partial example, rather than start something new.
- Template is a reminder of what has to be done.
- Blanks can contain default values.

This provides an example of the kind of information needed, or can provide current directory path for path queries, etc.

4) **Interaction**:

- Interactive interfaces are best.
- They provide continual feedback, and make limited demands on short-term memory.
- User can be notified of errors immediately.
- With batch input, user may continue to make errors unknowingly for a long time.

5) **Familiar conceptual model**:

- Easier to use a familiar conceptual model.
- E.g. the “desktop” with papers on it.
- E.g. a form for filling in data.

6) **Consistency**:

- Same command should perform the same function everywhere in the system.
- Same function should always have the same command to invoke it.
- Function keys should have the same meaning on every screen.
- Users get confused when a different command is used in programs to do the same thing.
- E.g. to exit, could use stop, quit, exit, bye, end, logout, etc.

7) **Simplicity**:

- Guideline: KISS — keep it simple, simon.
- Don’t clutter up the screen.
- Blank space is usually the best separator.
- Use sound only in emergencies. Why?
- OK to use a beep to warn touch-typers of errors.
- Users should have a way of shutting sound off.

8) **Tailorability**:

- Users should be able to tailor the interface to meet their needs and abilities, though this adds implementation expense.
- Should be able to specify
- Default starting points.
- Level of expertise.
- Terminal capabilities.
- Preferences for output format.
- Private mailing lists.
- Personal procedures, etc.

9) **Pointing**:

- Easier to use a finger, mouse or trackball to point at a place on the screen, rather than having to type in a command.
- Skilled typists may find function keys are faster for some things.

10) **Context sensitivity**:

- Options and commands available should depend on the current state of the screen.
- Help key should provide information about current screen.
- Inappropriate commands should be made unavailable.
- They might be marked grey to show they are not available.
- Reduces the need to learn and remember rules.

9.7.2 Components of Screens

- A screen contains many components.
- Components are the semantic elements of the dialog.
- Sequence in which they are used determines the syntax.
- Figures 8-4 and 8-5 of [Fertuck 92] illustrate.

Components are:

- **Panel title** (also called Title Bar) — tells the user what the panel is about.
- May contain a filename.
- May indicate position in a sequence of screens.
- Could contain a form number, or page number in a word processor.
- **Menu bar** — lists the main menu horizontally.
- Menu actions may be selected by pointing, cursoring, or typing.
- Pointing may be done with a mouse or touch-sensitive screen.
- A highlighted/underlined/upper-cased letter may indicate that typing a letter will select the action. Or Alt-typing the letter.
- **Pull-down menu**— lists the specific actions available in a submenu.
- Actions that generate another panel are shown by three dots.
- Users selects actions that are desired.

- **Tool Bar** - a (usually) horizontal row of action icons use for invoking frequently used commands
- **Check box** — another way to list specific options or actions available.
- User checks box or boxes needed by pointing or cursoring.
- **Icon** — an area with a picture that reminds the user of the mode or action that is selected.
- Another way of specifying options.
- Pictures must be chosen well.
- **Scrolling list** — another way to select options.
- Used where there is a potentially list of options know to the computer to select from (thus relieving the user of remembering and typing the one he wants).
- E.g. a list of files to select.
- A few items are shown at a time in a window.
- List can be scrolled to see other parts of it.
- **Scroll bar** — allows scrolling to another part of the list.
- A bar at the side of a scrolling window.
- Highlighted part of the bar rides up and down like an elevator to show current position in the window. On some systems, also indicates the fraction of the file showing in the window.
- **Window** — allows user to view part of a larger area.
- Generalization of a scrolling list to allow vertical and horizontal scrolling around a large area.
- Borders of window can be moved by using scroll bars.
- **Availability emphasis** — shows which options are available or unavailable.
- Some options shown in a different color. e.g. dangerous or urgent in red.
- Color indicates availability. e.g. ‘grayed-out’ means unavailable.

- **Pop-up window —** portion of the screen that pops up to extend the dialog.
- Used for error messages, confirmations or parameter specifications.
- Used when a specific task must be done before processing continues.
- **Heading —** used to identify columns or groups of related items.
- E.g. *Address* identifies the fields that make up an address.
- **Entry field —** space where users type data.
- Data may be entry into a database, a command, text in a word processor or any other kind of textual entry.
- **Prompt —** a label that identifies the kind of data that must be entered.
- Must be unambiguous.
- E.g. Does DATE mean shipping date, today’s date or expected date?

9.7.3 Styles of Screens

Screens have many purposes. Each purpose may require a different style of screen:

- **Logo panels —** displays ownership, copyright and login information.
- Usually occupies the whole screen.
- Usually the first screen.
- May disappear after a time, or provide a command to continue.
- **Menu panels —** contain one or more choices that a user can select.
- Use variants of action bars, pull-down menus, and icons.
- May take up a whole screen or only one line.
- **List panels —** usually a pop-up window that allows the selection of files, customers, accounts etc.

- List usually pops up when needed and disappears when selection has been made.
- **Entry panels —** used to enter data, parameters, text or graphics. These are the panels where the work of the application occurs.
- **Parameter entry panels —** used to enter required parameters
- Usually implemented as pop-up windows.
- May appear when an action has been selected.
- Usually has headings, prompts, and entry fields.
- Usually has some way of indicating parameters have been entered.
- **Form entry panels —** used to match the structure of a paper form.
- May be no corresponding paper form.
- Usually used for repeated entry, and take up whole screen.
- **Tabular entry panels —** accept repeated lines of data, like lines of an order form.
- Same characteristics as form entry panels.
- Heading at top of columns and several lines for data entry.
- **Information panels —** display help information, messages and reports.
- Content cannot be changed by user.
- Used for communication to the user.
- Usually implemented as pop-up panels.
- May inform the user about progress.
- **Combined panels —** often convenient to combine several panels on a single screen.
- E.g. entering kernel entity and characteristic entities.
- May also include information panels.
- Panels are sometimes called “panes” within a window.

9.7.4 Standards

- System needs standard rules for format, messages, and function keys.
- Standard keys for basic functions reduce learning time.
- Can also simplify programming, as utility routines can be written to perform the same task for each screen.
- Standards should be set for placement of major screen elements.
- This reduces the operator's search time.
- Experienced users should be provided with shortcuts.
- E.g. function keys, type-ahead buffers, etc.
- Modern graphical user interfaces (GUI) like Windows, Macintosh, OS/2, etc., have brought a set of widely recognized standards to screen design.

9.7.5 Physical Considerations in Screen Design

- Two main kinds of screens:
- Programmable — can process the logic needed to implement GUI.
- Unprogrammable — can only do this by communicating with a remote computer.
- This communication is long and costly when computer is far from screen.
- Terminal operator may have to wait.
- Screens can also be divided into graphics-oriented and text-oriented.
- Graphics increase cost, but increase the amount of communication.
- Most commercially implemented graphic interfaces have been implemented on personal microcomputers.
- Screen is attached directly to the computer.
- One CPU per screen.

- Graphics interfaces may not be necessary, and may be time-consuming.
- Trade-off between communication to user and processing time to paint screen.
- Infrequent users may need more user friendly, graphics-oriented communication.
- Some applications used by different types of users.
- Can provide several ways to use the screen, e.g.
- Pointer for poor typists.
- Function keys for experienced users with good typing skills.

9.8 Report Design

Design of reports is a cross between form and screen design.

- Reports are usually printed on paper.
- They may be shown on screens or stored in files for future use.

Printed reports are used when

- Output goes to an outside organization,
- Is too voluminous to be browsed on-line, **or**
- Is needed for control or audit purposes.

Screen reports are used for

- Database inquiries about a single occurrence,
- Low volume outputs, **or**
- When the inquiry is one step in an interactive process.

Many elements of a report correspond to elements of forms and screens. However, reports are frequently used to summarize data or identify subsets for further examination.

9.8.1 Report Design Criteria

- Reports convey information from the computer to the user.
- Must do it clearly and economically in a timely and accurate manner.
- Clarity requires that all intended information is presented in an understandable form.
- Economy requires that this be done without wasting the computer's resources or the user's time.

There are several criteria we can use to judge the quality of a report:

- **Timeliness —** must be prepared in time to be used.
- Most reports provide information used in making decisions.
- Information is of very little use after the decision deadline.

- **Accuracy —** must be correct.
- Data must be accurately recorded, transmitted and transformed into summary data.
- Data must be captured on time.
- We consider incomplete data to be inaccurate.
- **Relevance —** must be relevant to purpose of report.
- All of the relevant information must be included.
- Irrelevant data must be excluded.
- Result should include all that is needed, and nothing that is not.
- **Clarity —** should be clear and understandable.
- Report must be designed to present data in a logical order.
- Desired information should be easy to locate.
- Comparisons, ratios, exception flags and graphs should be used where appropriate.
- **Cost —** each report has two costs.
- Cost of preparation consists of analysis, design, computation and distribution.
- Cost of reading the report and locating desired parts is often ignored.
- Reading cost can be reduced by careful design.
- Should select only appropriate information and present it clearly.
- If cost of report exceeds expected benefits, don't prepare it.

9.8.2 Components of Reports

Reports have many of the same components as screens. They also have several unique characteristics:

- **Title Page —** should be one if there is more than one page.
- **Table of Contents —** should exist if report consists of several parts.

- **Footnotes —** explains special coding, etc.
- Reports may contain complex coding structures that need explaining.
- Could also be explained on a separate page.
- **Address —** If a customer's address is included, it should be placed so it will show through a window envelope.
- **Sorted Lists —** lists should be sorted in some sensible order.
- **Control Breaks —** lists of sorted items are often broken into groups.
- E.g. purchase sorted by customer name, so start new page for each customer.
- Such a report is said to break on customer-name.
- Headings and sub-totals may be inserted at a break.
- Figure 8-6 of [Fertuck 92] illustrates.
- **Summary Data —** users often do not want to see detailed data.
- Users may want to see an aggregated summary, like total purchases for each month or each customer, etc.
- Summary data may be included in a control break.
- Typical summaries are count, total, average, minimum, maximum.
- Summary data may also be summarized into a grand total.
- **Exception Highlights —** some reports are designed to identify special cases.
- E.g. excessive costs, low inventory.
- Report should highlight the exceptions automatically, using bold type, underlining, or asterisks.
- If only exceptional items are of interest, then others should not be listed.
- **Confidentiality —** multipart forms can be designed with black zones so that some information is invisible on some copies.

- Printing black ink on red paper makes reading possible, but photocopying extremely difficult.
- **End of Report —** should be identified so that user knows that all of a multi-part report has been received.
- May be identified by summaries, page count or a message.

9.8.3 Styles of Reports

Information can be communicated in three general ways:

- **Files —** are used when computer is communicating with another computer program.
- Data must be stored in some standard format.
- Program reads data and does further processing.
- Commonly used to send information to another department.
- Statistical packages — that then analyze the data and produce summaries, correlations, forecasts, etc.
- Graphic packages — that can produce a variety of graphs to help visualize the data.
- Spreadsheets — that can insert the data into a mathematical model to create special reports and projections.
- Word processors — to format data, insert it into reports, or merge information into form letters.
- **Text —** use words and numbers to provide information in the form of lists, tables, possibly with summary data and page breaks.
- Such reports may be concise, complete and difficult to understand.
- People have to compare individual numbers to make comparisons.
- **Graphs —** convert data into scaled lines, areas or positions that are easy to compare visually.
- Graphs are not as complete as tables.
- Precision is limited by resolution of the display medium, and the ability of the viewer.

- Graphs are very good at revealing relative size, proportion, trends and variation.
- These are often important concepts in business.
- Different kinds of graphs may be used, depending on number of variables, and whether they are continuous or categorical.
- Continuous: variable can take on any value within a range.
- Categorical: variable can take on any of a set of values. Values do not have to be numeric or have any order.

We'll look at the most common kinds of graphs, illustrated in Figure 8-7 of [Fertuck 92]:

- **Bar graphs** — used when a continuous variable can be divided into portions for each category of one categorical variable.
- E.g. sales in various regions.
- Provide a clearer indication than tables of numbers.
- **Pie charts** — emphasize proportion rather than amount.
- Sometimes one sector may be exploded.
- Actual amounts or proportions are shown with category labels.
- **Column Graphs** — show proportion of a continuous variable in each category of one categorical variable.
- Serve the same function as pie charts.
- **Stacked Bar Graphs** — are an extension of column graphs, using a second categorical variable.
- Shows the amount of the continuous variable that is each of the cells created by the two categorical variables.
- Same as a two-way cross tabulation chart.
- E.g. share of each kind of expense in each region.
- Shading indicates the different expense categories.
- **Area Graphs** — are similar to stacked bar graphs.

- The continuous variable is categorized by a single categorical variable and is plotted against another continuous variable (usually time) on the horizontal axis.
- Shows how share of each category changes over time.
- **Line Plots** — used when two continuous variables are being related.
- Plot shows the relationship.
- Horizontal and vertical axes are used to plot the two variables.
- When the horizontal axis is time, is called a Time-Series Plot.

A good graph follows some guidelines designed to make sure the impression corresponds to the reality.

- The zero point of the dependent axis must be shown clearly.
- Otherwise, trivial variations may look more important than they are.
- The visual space represented by a variable should be proportional to the data being graphed.
- Representing labor costs by pictures of people of different heights is misleading.
- Be aware that as costs double, area covered by a person wrongly gives the impression of quadrupling.
- Monetary time series should be deflated to remove the effects of inflation, and should be converted to common units.
- Multi-dimensional graphs are good to portray functions of several variables.

9.8.4 Physical Considerations in Report Design

Some characteristics are important in choosing an output device:

- **Resolution** — refers to the amount of detail being printed.
- Usually measured in dots per inch (dpi).
- Visual clarity improves as dpi increases.
- 75 dpi is minimal for readable output.

- 300 dpi is adequate for office correspondence.
- 1000 dpi is used in high quality printing.
- **Type fonts —** are complete sets of print characters with the same face, size and style.
- Face refers to the character shape.
- Style refers to thickness and slanting (italics).
- **Graphics —** is the ability to draw lines and pictures.
- Many devices can only print characters.
- **Volume —** is the speed of output.
- May be measured in characters/minute (cpm), **or**
- Lines/minute (lpm), **or**
- Pages/minute (ppm).
- Can range from 1800 cpm for slow dot matrix printers to 150 ppm for large laser printers.
- **Cost —** is the production expense per page for paper, ink, maintenance, operators and equipment depreciation.
- Should also count cost of users time getting prints.
- Cost increases with increases in resolution and choices of type.
- Cost decreases as volume increases.
- Low-volume devices often used because of convenience.

Most common output devices are:

- **Dot matrix printers —** print individual characters by impact of a vertical row of wires through an inked ribbon.
- Character is constructed from an array of dots.
- Resolution depends on number of dots used.
- Resolution varies from 75 to 150 dpi.
- Speeds vary from 1,500 to 15,000 cpm.
- Cost as low as a few hundred dollars.

- Can print graphic images.
- **Ink jet printers —** produce an image by droplets of ink directed at the page by electrostatic charges.
- Somewhat more costly than dot-matrix.
- Can produce graphic images.
- Better looking output.
- **Line printers —** print a whole line at a time by the impact of hammers through an inked ribbon.
- Can achieve volumes from 200 to 3,600 lpm.
- Resolution can be good with carbon ribbons.
- Only one font available, no graphics images.
- Can use carbon paper for multiple copies.
- Often used for high volume applications like billing.
- **Laser printers —** are similar to a photocopier.
- Time to print a page is independent of number of characters.
- Graphics or many fonts can increase time.
- Volumes from 4 to 150 ppm.
- Costs range from below $1,000 to hundreds of thousands.
- Very high quality output.
- **Pen Plotters —** have one or more pens moved over a flat sheet to draw continuous lines.
- Resolution depends on the pens.
- Volume depends on number of lines to be drawn.
- Typically used in scientific and engineering applications.
- **Microfilm —**used when economy, storage space or archival permanence are important.
- Costs 15 to 20% as much as paper.
- Takes 2% of the space.

- Lasts almost forever.
- Magnifying viewer is needed to view the result.
- Relatively poor resolution.
- Often used for catalogs, bank insurance or hospital records.
- **Screens** — are used when output does not have to be stored.
- Common with random inquiries, like airline reservations.
- Volume limited by transmission speed from computer and rate at which user can read the screen.

9.9 Design by Prototyping

- Prototyping is a technique where analyst rapidly develops a proposed screen or report, then asks user to critique it.
- Initial design is called a prototype.
- Design is modified on the basis of the critique.
- Process is repeated until design is satisfactory.
- Can be a very effective process as
- Users may have trouble specifying what they want.
- Users can usually recognize what they want.
- Prototype is a good way of triggering the user's memory and initiating a dialog about the design.
- Prototyping now possible with modern screen generators.
- Prototyping should not be done before database design.

Correct order for prototyping steps is:

- Prepare a data model.
- Identify knowledgeable users who have an understanding of data integrity requirements.
- Start producing screens for these users.

Prototyping can take place in front of a user if a data model is in place and a screen generator is available. This will make sure that

- Data are provided where necessary.
- Format is easy to use and understand.
- Variations of records or transactions are handled.
- Input errors are detected.
- Help is provided to make screen self-documenting.

- Should be done with a representative sample of users.

- Prototyping is an alternative method of systems analysis and design.
- Not effective for batch programs or modules with complex logic.
- Prototypes can also be used for exploration and experimentation.
- Prototyping speeds development of a good system.
- Traditional SDLC delivers a first prototype only.
- Prototyping process needs clear stages — otherwise it goes on forever.
- The design process has three stages:

1) Standard setting.
2) Design — within this stage there are three phases:

- Identification.
- Formatting.
- Enhancing.

3) Packaging.

- Final packaging integrates all screens into menus.
- Users must agree to freeze specifications at end of each stage.
- These stages are illustrated with an example from STARS.

We'll now design a screen to

- Enter student enrollments in courses.
- Update enrollments.
- Retrieve information to answer inquiries from students.

9.10 Testing Screen Designs

- Prototyping means users are continually critiquing the design.
- Function of each screen element should be clear without reading the words.
- Each entry field should be tested with valid and invalid data.
- All commands and options should be tested.
- Testing should be done by a test expert not familiar with the application.
- Careful design and implementation does not necessarily mean system will be error-free.
- Independent tester makes no assumptions and has no biases.

9.11 CASE Tools for Input and Output Design

Tools to aid in designing forms, screens and reports are found in form, screen and report generators and CASE tools.

- **Form generators** are usually specialized word processors or desktop publishing systems.
- Make it easy to produce ruled lines, boxes, and labels.
- Can usually produce a high-quality copy.
- **Screen generators** use the data dictionary of the DBMS.
- Retrieve names and attribute specifications from the DBMS.
- Use them to generate a prototype of the screen.
- **Report generators** are like screen generators.
- Generate an automatic prototype of the report.
- Usually have complicated formatting facilities.
- Produce code that is used in the final implementation.
- **CASE tools** usually have screen prototyping facilities.
- Usually have screen prototyping facilities.
- Obtain information from the CASE data dictionary.
- Usually cannot produce a finished screen.
- Better to use a screen generator coupled to the DBMS.

Most end-users can use report generators to get their own specialized reports.

- Reports can be changed easily.
- End-user activity reduces maintenance work.
- Reports are often left for the user to develop.

9.12 Summary

- Forms, screens and reports are a means of communication between people and the computer.
- Not all input comes from forms.
- Screen painters can be used to simplify the job of screen implementation.
- Prototyping is an effective way of communicating between designers and users.
- Users stay involved until their needs are met.
- **Data navigation diagram** identifies the tables used in a screen.
- **Action diagram** specifies the actions performed by a screen.
- Screen design can get complicated if different countries or languages involved.
- When screens have been implemented, data can be entered into the database and reports can be generated.
- The next step is to implement batch programs that manipulate the data that has been entered into the tables.

9.13 References

[Fertuck 92] “Systems Analysis and Design with CASE Tools” by Len Fertuck, Wm.C. Brown Publishers, 1992.

[Kendall 92] “Systems Analysis and Design”, 2nd ed., by Kenneth and Julie Kendall, Prentice-Hall, 1992.

9.14 Appendix 9A - Screen and Report Design Forms

9.15 Appendix 9B - Examples of Reports

9.16 Abandoned Material from Fertuck

9.17 Design Example Stage 1: Setting Standards

- This phase sets standards about screen and report formats.
- May also set standards about the tools or languages used.
- Main objective: produce consistent products.
- Main elements of a standard deal with:
- Capabilities of target equipment which will display screens.
- Use of function keys.
- Capitalization.
- Fonts.
- Ruling lines.
- Color.
- Position of screen elements.
- Use of GUI components (titles, icons, window sizes, etc.).
- Most GUIs have their own standards for many of these issues.
- Standards may also cover invisible components:
- Internal documentation.
- Error checking.
- Testing.

9.18 Design Example Stage 2(a): Identify Data and Actions

- First stage of prototyping is
- Identification of data that has to appear on the screen.
- Actions that have to be performed on the data.
- Data are organized into:
  - Screens.
  - Panels within screens.
  - Fields within panels.
- Basic actions are:
  - Reading fields from tables.
  - Entering data into fields on the screen.
  - Checking the validity of entries.
  - Writing fields to tables.

9.18.1 Identifying Screens, Panels and Fields

- Most screen painters make it easy to design screens that can do:
  - Inquiries.
  - Updates.
  - Insertions.
  - Deletes.

  on the same screen.
- If a screen contains entries from more than one table, designer must identify these tables.
- Easy to do by checking the E-R diagram.
- Figure 8-9 of [Fertuck 92] shows the portion of [Fertuck 92]'s Figure 5-6 used in this design.
- Operations to be performed are written above each entity.
- When the E-R diagram is annotated to specify the order in which data tables are accessed, it is called a **data navigation diagram**.
- Numbering of annotations gives ordering.

- Users may identify additional attributes at this time.
- Useful at this point to sketch proposed arrangement of fields on the screen.
- Sketch shows user what screen will look like.
- Details should be left until all fields have been identified.
- Analyst should generate first sketch.
- Figure 8-10 of [Fertuck 92] gives a sketch of first proposal.
- Only some fields are necessary to record an enrollment.
- Others are present to provide information for checking and inquiries. e.g. "Are you the John Smith that lives at 235 Main Street?"

9.18.2 Identifying Actions

- Actions to be performed were specified on the data navigation diagram.
- More formal specification needed now.
- Figure 8-11 of [Fertuck 92] shows an accompanying **action diagram**.
- This is closer to code, and is a guide for programmers.
- The IF statements specify the conditions to be satisfied during screen use.
- Brackets show hierarchical structure of instructions.
- Statements are numbered to correspond with data navigation diagram.
- Bracket with double line at top signifies iteration.
- Condition at top of bracket applies to all instructions enclosed.
- User must agree at end of this stage that all screen elements have been identified.

9.19 Design Example Stage 2(b): Formatting

9.19.1 Define Format

- Designer and user can now look at screen format.
- The elements are labels, positions and emphasis.

9.19.2 Position

- Screen is constructed in panels.
- Each panel presents the elements of a single table, and any items related to the table.
- Figure 8-12 of [Fertuck 92] shows how.
- Prompts should be aligned on the left.
- Fields should be aligned on the right.
- Figure 8-13 of [Fertuck 92] shows panels with tentative labels.
- Positions are chosen to make screen easy to use.
- If screen is used to enter data from a form
- Positions of blanks on form should correspond to screen.
- Order of entry should be the same.

9.19.3 Labels

- Labels must be clear and readable.
- Attribute names may be chosen.
- They may not be good for screen users.
- Some DBMS maintain attribute name and printable report name for each attribute.
- Figure 8-14 of [Fertuck 92] shows new labels suggested by users for ease of understanding.

9.19.4 Emphasis

- Screen elements can be emphasized by
- Borders.
- Blank areas.
- Capital letters.
- Brightness.
- Color.
- Reverse or blinking video.
- Sound.
- Colors can be dangerous if some screens are monochrome.
- Remember also that some people (about 5%) are color-blind.
- Blinking characters and sound should be reserved for emergencies.
- Figure 8-14 of [Fertuck 92] illustrates the use of a border and shading.
- At the end of this stage, users should agree to screen format.

9.20 Design Example Stage 2(c): Enhancements

9.20.1 Add Enhancements

- Last stage is to turn prototype into “industrial strength” application screen.
- Finished screen needs
- Validity checks.
- Operator aids.
- Help facilities.
- Documentation.
- These features may take some time to develop.
- User not involved with most of this.
- User may have difficulty understanding why screen isn’t finished yet.
- Must resist pressure to deliver screens before enhancement finished.

9.20.2 Validity Checks

- Ideally, validity checks performed by DBMS, though some identifying fields can contain a checksum digit to help detect typing errors.
- Many DBMS do not do all desired checking.

Following checks need to be done:

- **Mandatory fields**: some fields must have a value.
- **Unique values**: keys must have unique values.
- **Mandatory relationships**: some foreign keys must exist in both related tables.
- **Unchangeable fields**: some fields must not be changed.
- **Hidden fields**: should not be updated by operator (e.g. time of transaction, etc.).

- **Allowable values**: each field must be checked for allowable values. DBMS may do this automatically. Selecting from a scrolling list of allowable values is a good technique.
- **Character set** — may be restricted to numbers, letters, capital letters, etc.
- **Value set** — may be restricted to a particular set of values, e.g. months of the year. Foreign keys must be checked against parent table.
- **Range** — range of numeric entries may be restricted to integers, and to values between an upper and lower bound (e.g. percentages).

Five kinds of events can trigger the various checks and aids:

- **Entry**: when cursor first enters a screen, panel, record or field.
- **Query**: before or after records are retrieved.
- **Change**: after a field is changed, or before or after insertion, deletion or update.
- **Exit**: when cursor leaves a screen, panel, record or field.
- **Keystrokes**: when the operator presses a function key.

Text explains checks required for STUDENTS example.

9.20.3 Operator Aids

Several features can be added to improve operator efficiency:

- **Defaults**: can provide predictable values.
- **Scroll lists**: can provide a list of possible values.
- **Sorting**: useful to sort list and tables.
- **Queries**: often helpful if the screen can be used to search for selected records.
- Can fill in known fields and let computer find the matching records.
- May provide an SQL window for queries.

9.20.4 Helps

Users may not always know or remember what the screen is for or how to use it.

Help needs to deal with the following questions:

1) "*How do I learn about the system?*"

- Interactive tutorials can be useful.
- Very expensive to hold training courses.
- Interactive tutorials are always available.
- Especially good for large systems with scattered users.

2) "*How do I navigate through fields and screens?*"

- User needs help to understand standard commands.
- User needs to know how to delete, insert, navigate, cancel and log off.
- Should be one standard function key, usually F1 or '?'.
- Help might be cross-referenced to the tutorials.

3) "*What does a screen do?*"

- Requires descriptive titles and meaningful description of each choice on screen menu.
- Avoid acronyms.
- Bad example: UNIX's **GREP**, which stands for **G**lobal **R**egular **E**xpression **P**rinter.
- First layer of help should explain the use of the screen.

4) "*What data goes in a particular field?*"

- Need to explain how to enter data in a particular field.
- This layer of help should be context-sensitive to the current cursor position.
- Context-sensitive help systems provide information about the current screen, field, or command.

5) "*What did I do wrong?*"

- Error messages should be clear.
- Avoid messages like "IEC071 DATA ERROR" — this tells the user nothing.
- Better — "Error: this field must contain a date in the form YYMMDD".
- Avoid threatening words like *abort*, *terminate*, *crash* and *fatal*. The NeXT user interface occasionally uses the phrase "system panic" which is more than a little disconcerting even to a computer scientist.
- Words like this may discourage or frighten users.
- Error messages should have a standard format, and appear in the same place on the screen.
- A good error message specifies the nature of the error and what form correct input should take.

When validity checks, operator aids, and help facilities have been added, the screen is ready to be tested.

- Each field must be tested to ensure that it rejects incorrect data.
- Commands must be tested.
- Screen should then be tested with real users.
- This will test the quality of helps and prompts.
- Best to use users who were not part of the specification process.

9.20.5 Documentation

System now needs documentation to record purpose and function of screens and programs for maintenance, user training and user reference.

There is a trend now towards on-line documentation. This brings advantages:

- **Low cost —** documentation does not have to be printed.
- Can be shipped with software or on update diskette.

- Easy to update.
- **Convenience** — always available.
- Can be used where and when needed.
- Doesn’t require a trip to where documentation is kept.
- **Intelligence** — can be connected to the application.
- Help key can bring up on-line documentation.
- Scrolling windows can provide multiple pages.
- Can use search and index functions.
- Can use hypertext, or even hypermedia.
- Can be linked to tutorials.

9.21 Design Example Stage 3: Packaging

Integrating Screens With Menus

- Each main screen performs a specific self-contained activity.
- Different screens are integrated into an application by collecting groups of them in menus.
- Complete set of screens should be designed consistently.
- Screens are arranged in a hierarchy.
- Menu should not contain more than 10 items.
- May need several layers of menus in large systems.
- Screens should be grouped sensibly.
- Selection menus can easily be rearranged.