

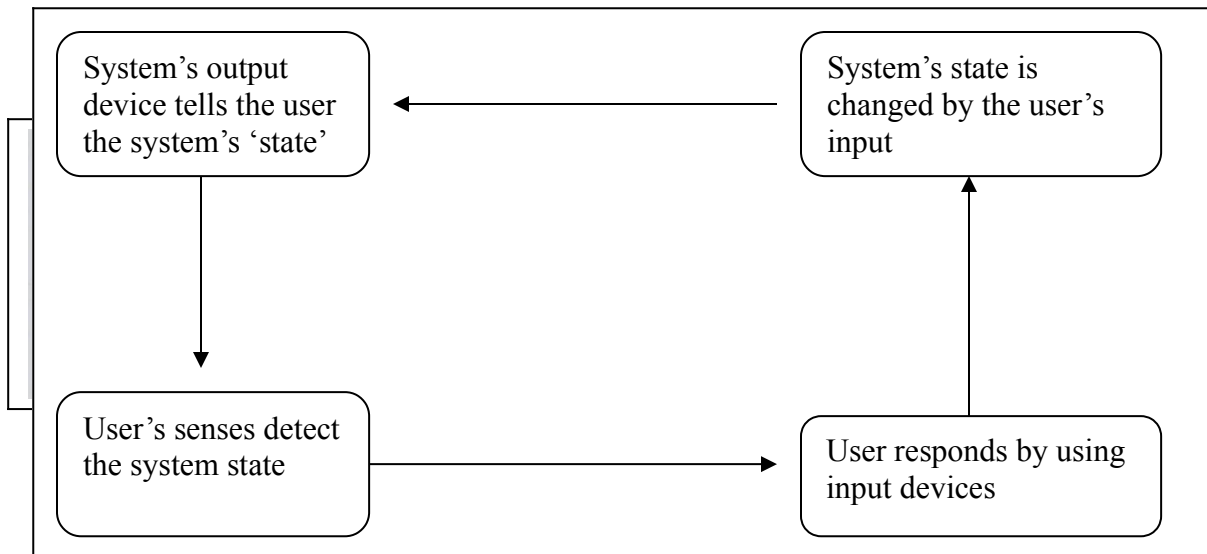
HCI-6- INTERACTION STYLES 1

Aims and Objectives: this lecture follows one from the previous one, which discussed I/O devices. Whereas the previous lecture had hardware focus, this one looks at software and specifically what are termed as ‘Interaction Styles’. It explains what is meant by the term ‘Interaction Styles’ and describes how interaction styles have evolved. It then discusses the relative merits of a number of common interaction styles in terms of how they support the general principals of good interface design. One particular important interaction style, direct manipulation, is covered separately in the following lecture.

Lecture Notes.

Introduction to Interaction Styles

The diagram below gives a very abstract view and generalised view of what is sometimes termed the Human Computer Interaction Loop.



In order to describe the loop (sometimes called a dialogue) we can break into it at any point. For the sake of starting somewhere we will start with the top left-hand box. Let's say for example that you have just switched on your PC and loaded up Microsoft Word. The system will probably indicate its state via the screen which will show Word loaded and ready to do something. Moving to the next box. Your sense (probably eyes in this case) will detect the state of the system i.e. you will see that Word is loaded and ready for action.

Moving to the bottom left-hand box. You may respond to the system's state by using an input device. For example you may use the mouse to click on the "file" menu in Word. As indicated by the next box in sequence, your action will cause the system's state to change. In this example Word will now be in a state of expecting you to select something from the "File" menu.

We would go round the loop again. The system will indicate its state via the screen (i.e. the "File" menu options will display), you may react to this (e.g. pick the "Open" option from the list) the system's state will change again (it now expects you to tell if the file to open) and so on.

This view of interacting with a computer system may seem very abstract but the point is that it is such a generalised view that any sort of interaction with any sort of system (mobile phone, games console, computerised ticket machine etc, can be analysed in terms of this framework.

The different ways that the interaction loop can be executed (e.g. via menu choices, command input, point and click etc) are what are termed interaction styles.

Before looking at the characteristics of different interaction styles in detail we will review the principals of a good interface design

Principles of Good Interface Design

1. Naturalness

The interface should not get in the way of the user carrying out their task (e.g. writing a letter booking a flight etc) It has been said that the interface is "invisible" to the user. This means that the user should not really be aware of performing their task.

2. Consistency

It was noted in the lecture on human memory that ideally a good interface should be consistent (e/g/ in data format, use of icons, terminology etc). The consistency can be between different parts of the same system, between different releases of the same system and with other similar systems (e.g. using standard icons).

3. Relevance

This means that the user should have to make the least possible effort to achieve their aim. They shouldn't be required to enter information that is not relevant to their task nor should they have to process irrelevant output. Think how annoying it can be when using the web to have to enter lots of details into a form before allowed to subscribe to some free service. Of course in this example the only reason the service may be free is because the provider wants to get hold of those details for advertising purposes!

4. Supportiveness

Users (especially novices) shouldn't be left wondering things like:

- what can I do next?
- did that last action work?
- where am I in the system?
- what inputs are valid here?
- why didn't that work?
- Etc.

5. Flexibility

Ideally the interface should support different user's needs and performances. A simple example would be that the user should perhaps be able to choose the colours used by the system

Quick Quiz Question 1

- What are the four stages of the "interaction loop"
- One of the principles of good interface design is that the interface should be flexible. Can you think of any disadvantages of an extremely flexible system that allows you to customise all aspects?

Interaction Styles – A Very Brief Historical Perspective

Below is a very quick review of how interaction styles have evolved. At each stage the dominant interaction styles are noted (*in italics*) and the main type of users identified. Of course this is a very much simplified picture.

- **Early Systems**

Early interactive system largely used a *command-driven* interface. Most users could be considered as experts i.e. academics, scientists or other highly trained users.

- **Data Processing Systems**

With the rise of mass data processing systems *form filling* interfaces became common. Often the layout of the form would match that of some paper document from which information was being entered. Users would be trained for the task (e.g. data entry clerks).

- **Mass computer use and the rise on the PC**

With the mass use of computers and particularly the dominance of the PC more supportive and flexible styles have emerged. Common styles are *menus* and *questions* and *answer dialogue*. Most significant has been the emergence of what are termed *direct manipulation* interfaces. These are discussed in some detail later in the lecture

- **The era of mobile computing**

It seems that now we are moving into an era of even more wild spread computing perhaps most significant characterised by the increasing of mobile computing devices. Will new interactions styles become common (e.g. *voice recognition*) or may we switch back to using some older styles (e.g. many mobile phones make heavy use of menu style interface.

It should be noted that the emergence of newer interaction styles do not necessarily totally replace old styles. For example command driven interfaces are still in use. It is also important to realise that different interaction styles are not mutually exclusive and commonly co-exist within the same system.

Each interaction style is described below:

Below is an example of interacting with a command driven interface, namely MS-Dos

```

C:\Gill\temp>dir
Volume in drive C has no label
Volume Serial Number is 3045-1505
Directory of C:\Gill\temp

.<DIR>      02/02/98  12:08 .
..<DIR>      02/02/98  12:08 ..
BOOKING  CPP      2,820  18/12/97  13:30 booking.cpp
BCWDEF   CSM     102,970  02/02/98  13:00 BCWDEF.CSM
2 file(s)      105,790 bytes
2 dir(s)    1,084,850,176 bytes free

C:\Gill\temp>renam booking.cpp booking2.cpp
Bad command or file name

C:\Gill\temp>ren booking.cpp booking2.cpp

C:\Gill\temp>cd ..
  
```

The screenshot shows a MS-Dos command prompt window. The first command is 'dir', which lists the contents of the directory C:\Gill\temp. The output shows two files: 'BOOKING.CPP' and 'BCWDEF.CSM'. The second command is 'renam booking.cpp booking2.cpp', which results in an error message 'Bad command or file name'. The third command is 'ren booking.cpp booking2.cpp', which also results in an error message 'Bad command or file name'. The fourth command is 'cd ..', which changes the directory to the parent directory. There are two callout boxes: one labeled 'unforgiving' pointing to the error messages, and another labeled 'no feedback' pointing to the lack of confirmation for the 'ren' command.

Such interfaces tend not to be very supportive of the user for example:

- The user often has to explicitly request information about system state e.g. by entering a "dir" command.

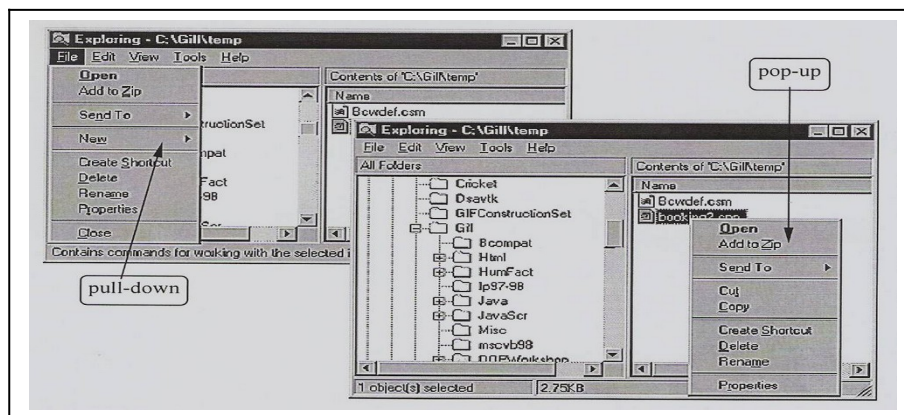
- The user has to remember what commands are available and also remember their correct syntax.

The system may be unforgiving when the user makes a mistake. For example you may think that the system should be able to guess that the user wants to rename their files when they type "renam" but in fact it won't recognise the command unless you get it exactly correct.

In spite of these drawbacks command driven interfaces can be very **quick** and **flexible** for **expert** users. They are still commonly used by people such as UNIX system administrators.

Menus

Menus consist of a set of options – typically only one of which can be selected.



The main advantage of menus over command entry systems is that the user only has to recognise the required option. It is a characteristic of human memory that we find it easier to recognise what we want from a list of options than remember what is available.

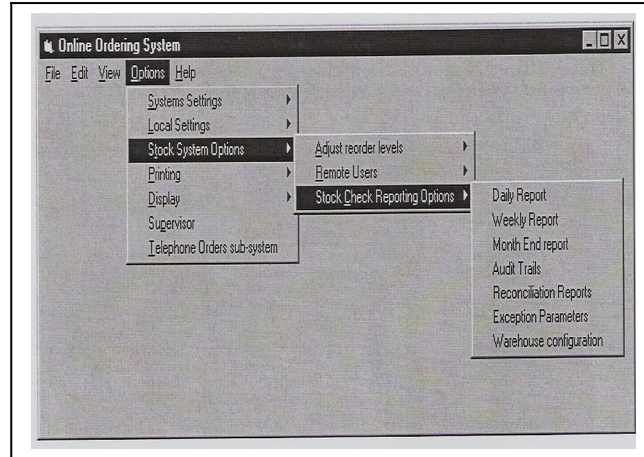
In the early days a menu might fill the whole screen and the only way of progressing would be to choose an option. Nowadays we are used to pull-down and pop-down and pop-up menus whose options only appear when we request them. The disadvantage of pop-up and pull-down menus is that the user needs either to remember which menu the option they want is on or hunt for it.

Menus may be slower to use for experts than command entry systems although this can be alleviated by providing shortcut e.g. via function keys.

When designing menus there are a number of issues that need to be considered two of these are considered below:

Organising multiple menus

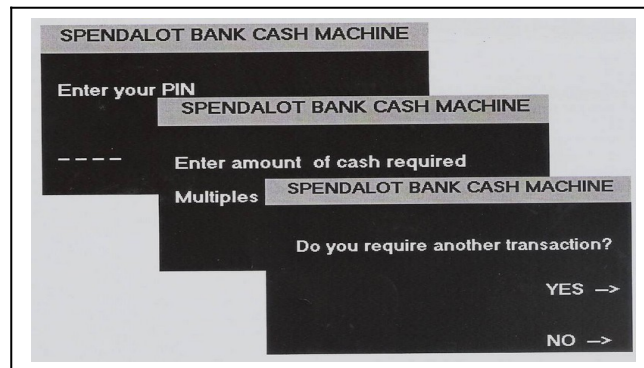
- Typically a hierarchical organisation is used. An issue is how many levels of menu are acceptable. Do you think there are too many levels in the example below?



- Ordering lists of menu items within a menu. Some possibilities are: alphabetically based on logical categories or based on frequency of use with the most frequency used options appearing towards the top of the list.

Question and Answer Dialogue

With a question and answer dialogue systems present the user with question, the user answers it (e.g. by entering some text or selecting an option) and the system responds by asking another question.



This interaction style can be very supportive. The user isn't left wondering what to do next, except when they don't know how to answer the question of course! This style can be particularly appropriate for novice users; whereas expert users may find the feeling that the system is in control rather annoying and inflexible. A common example of question and answer style dialogue are the "wizards" provided by many software packages to guide users through what otherwise maybe complex. Wizards usually provide a certain amount of flexibility e.g. the user can return to earlier questions to change their answer.

Forms

Forms present the user with a number of entry fields in a fixed format. They are particularly appropriate for data input e.g. the entry of customer orders from printed order forms.

The screenshot shows a web form titled "Saxon Direct - order input". At the top right, there is a field for "Operator Id" with the value "dfw0123". Below this, on the left, is a field for "Cust Account Number". The main part of the form is a table with three columns: "Cat Number", "Qty", and "Size". There are eight rows of input fields for each column, totaling 24 input fields. The form is designed to look like a paper-based order form.

Forms may support the user to some extent in that it is usually clear what data is required to be input. The form may also only allow data of the appropriate type to be input (e.g. numeric data in some fields, textual data in others). A form can feel a very natural style of interface as we are all familiar with filling in paper based forms. It can't be over emphasised how important it is that the user is entering data from a paper-based form that the layout of the online form should match it as closely as possible. For instance, if the user has to transpose fields (e.g. if quantity and size reversed on a paper based form used with the online form shown above) they will almost certainly make more errors than if the order of the fields match.

Quick Quiz Question 2

Identify two issues involved in menu design

Natural Language

Natural language interaction with a computer (either spoken language or typed) may in some ways seem the most natural interaction style imaginable. After all it is the most common way that we interact with each other. Natural language understanding (especially of spoken) is a very difficult and complex problem for software developers to deal with because of individual difference between people the ambiguity of natural language and the sheer size and complexity of the task. However, in the past few years there has been considerable progress in this area leading to the development of systems that can understand a restricted subset of a language for use in a particular applications area.

Even if true natural language understanding becomes possible there may be a number of problems from the user's point of view. These include the number of words required to explain something (especially a problem for typed input) and the need for a way to correct the computer when it understands something wrongly.

Summary

This lecture has discussed the concept of interactions styles. The "interaction loop" is an abstract way of thinking about the way a human communicates with a computer. Interaction styles are the different ways this interaction can take place. Different interaction styles have evolved over time from early command driven interfaces to the common direct manipulation interfaces that are common today. Different interaction styles support the principles of good interface design to different degrees.

The interaction styles discussed in the lecture are: command driven, menus, question and answer dialogues and natural language.

Having completed this lecture you should.

- Be able to explain the concept of interaction styles and their evolution
- Be discussing the characteristic of common interaction styles and the degree to which they support the principles of good interface design.

Tutorial Exercises

- 1 The table below can be used to make comments about the various interactions styles covered in this lecture under headings that refer to the principles of good interface design. As you can see the table been partially completed. Complete the table by including your own comments. There probably won't the room for you to write them on this copy, so you may need to make your own table. Having completed the table try discussing your comments with another student if you are in a position to do so.

Style	Naturalness	Consistency	Relevance	Supportiveness	Flexibility
Command Driven	<i>Not initially - may become natural with use. Depends how well the command language is designed.</i>	<i>A well-designed command language will be consistent e.g. similar options available across a number of commands.</i>			
Menus			<i>Normally menus don't require the user to type much, which is a good feature. The user may be required to search through a lot of menu items not relevant to them which is a less good feature.</i>		
Question and Answer dialogues				<i>Tend to be very supportive in that the user is led through the interaction by the system.</i>	
Forms					
Natural Language					

2. In the lecture natural language interaction was mentioned but not discussed in much detail several web search engines (e.g. Ask leaves) use a sort of natural language interface. Try out one of these search engines and write a brief report comparing it to one of the more traditional SEARCH ENGINES.
3. Choose a word processing system with which you are familiar with and answer the following questions about its menus.
 - What type of menus does it use (pop-up pull-down etc?)
 - What is the maximum level of menus that it has?
 - Within a menu how are the items organised?
 - Are there any sort-cuts available for expert user?

