

What is a User Interface?

The **user interface**, or the **human/computer interface** is what the user sees, and includes:

- The physical controls – buttons, etc.
- What the system looks like how the system accepts input from the user
- How the system responds to user input
- How the system outputs the results of processing



Specialised User Interfaces

There are many examples of computer applications with specialised interfaces:

- Tills in McDonalds, are often just ordinary PCs with specialised keyboards
- Games consoles – Xbox Kinect, PlayStation V.R., Alexa
- Computers are often adapted for people with disabilities – e.g. computers operated by blow-pipes, and Stephen Hawking's speaking computer



What Makes a Good User Interface?

Good User-Interfaces are...

- Safe – not ambiguous or confusing (especially in safety critical applications)
- Effective – they do what they're supposed to do, quickly
- Efficient – they are clear and easy to use
- User-friendly – intuitive and easy to learn
- Enjoyable – HCIs can be irritating, e.g. "Are you sure?" messages with no Yes or No buttons



Designing the User Interface

When designing the HCI, what will you consider?

- Who will be using the system:
 - What is their level of skill and knowledge?
 - Are there any special physical requirements?
 - Will there be a range of users?
- What does the system actually do?
 - How many options are there?
 - How accessible do they need to be?
- The environment in which it will be used – e.g. military use, temperature, noise, motion, moisture
- The technology available – iris recognition, voice, etc.



Types of User Interface

Main types of U.I.

- Command line – e.g. DOS or SQL
- Menu-driven – e.g. old DOS applications, camera menu
- Forms and dialogue boxes – used in Windows applications – e.g. Print or Browse
- Graphical User Interface – also known as GUIs, or WIMP (window, icon, menu, pointer, or window, icon, mouse, pull-down menu)



Common User Interfaces

Many applications, especially those that run in an operating system with a GUI have a similar appearance – a common user interface.



Windows applications, for example, all support:

- Ctrl C for copy, X for cut and V for paste
- Use of the Tab key to move between controls
- Use of the Alt key to operate menus
- Plus many more!
- But all the above I wouldn't be able to do in linux or iOS applications.

Advantages of a Common Interface:

- Having a common user interface brings benefits:
- It's quicker to learn new applications
- Familiar interfaces make applications easier to use
- All applications looking the same makes inexperienced users more confident
- Once an ICT "expert" is familiar with Windows, they should be able to operate almost any application

Command Line Interfaces

- Often very flexible
- Good for expert users who know the commands – not hunting through menus for the option you require
- Usually require fewest computer resources
- Requires users to learn commands
- Things you don't know about are invisible
- Not very good for novices or people who can't type

Menu-Driven Interfaces

- All of the options are visible – no hidden features
- No obscure commands and syntax to learn – therefore the it will be easier for new users and less training will be required
- Can be frustrating for experienced users who want to do something buried behind 10 menus!
- The user interface may be limited – e.g. by the size of the screen and the number of options you can fit on

Natural Language Interfaces

- No training required – you just tell it what you want!
- They could potentially be as powerful and flexible as CLIs
- No menus to negotiate
- Makes the computer look intelligent to naïve users
- Some sentences are ambiguous – "Let him have it!"
- It would be difficult to recognise all the different ways of saying things (and regional dialects)
- Artificial languages are often more precise
- Not yet popular

Forms and Dialogue Boxes

Issues when designing forms and dialogue boxes, all of these could be seen as a pro or con:

- Good and easy to use simple Layout – not too cluttered
- Navigation – order of fields, tabbing, etc.
- Field sizes - appropriate to data expected which helps with data validation
- Whether forms and dialogue boxes are modal – i.e. can the user access anything else while they are displayed?
- Use of appropriate controls and validation – combo-boxes, radio buttons, checkboxes, etc.

Use of Sound

- Feedback using sound is easily noticed – e.g.
 - Telephones
 - Alarm clocks
 - E-mails arriving in Outlook!
- Users can find too much noise distressing, so it needs to be used carefully
- Sounds won't necessarily be heard in a noisy environment such as a factory
- Sound output could be in the form of speech – either synthesised or digitised



Speech Recognition

Speech can also be used for input:



- **Command and Control** – small vocabulary systems can be used for controlling devices or systems. They are usually better at recognising different voices as there are fewer words to differentiate – e.g. just Yes or No
- **Dictation systems** – e.g. Office or ViaVoice, these are large vocabulary systems used to enter text. They require training for your voice, and tend to be less reliable and more resource hungry than voice control systems.

5 Core Principles of User Interface Design based on Apple's UI design

1) Clarity & Simplicity

- Ask yourself "is everything deadily obvious?"



You don't need names to know what these buttons are!



2) Flexibility

- Design something which can be good in all situations.
- Think through all possible content situations



3) Familiarity

Keep it simple – our eyes love seeing simple and familiar things. Spend less time re-inventing the wheel and use common solutions well.



4) Efficiency

Make sure the user can complete their task in the most efficient way. A lot of shortcuts provide more efficient strategies for users.



5) Consistency

A way to start feeling familiar about certain things, even if they were unknown before. Consistency and structure help us feel at home and at ease.