**Usability Engineering**

**A Brief History**

The earliest computers were experimental machines.

They were programmed and operated by the same people who designed and built them.

The next generation of computers were large mainframes.

They were programmed and operated by highly-trained staff.

Computers of this era were expensive and relatively rare, so it was important to use them to their maximum potential.

To achieve this, they were usually *batch-programmed*:

* Tasks (programs and data) prepared off-line.
* Batch of tasks loaded and run in sequence.
* Operator intervention limited to preparing and loading batches.
* No intervention/interaction whilst batch is running.

The next generation of computers were terminal-and-server systems or individual desktop systems.

They were programmed and operated by trained staff.

Usability became a significant issue when computers began to be used:

* By people with little or no knowledge of computers.
  + Training non-technical staff to use computers is expensive, so employers demanded computers that could be used with less training.
* By people with special needs.
  + For example, people who are blind/visually-impaired require an alternative to a visual display.
* In safety-critical applications.
  + Even highly-trained staff make mistakes, but in some applications operator-error is unacceptable.

Early computers were difficult to use because they employed Command-Line Interfaces (CLIs).

In order to use a CLI, the operator must know a large number of commands and the correct syntax for use with each.

Thus CLIs rely heavily on *recall*.

However, psychological research suggests that humans are better at *recognition* than recall.

In the 1960s, a number of researchers set out to create computer interfaces that relied on recognition.

In 1968, Douglas Engelbart demonstrated a system that used menus in conjunction with a pointing device.



Other researchers extended the approach to include windows and icons, thus creating the first Graphical User Interfaces (GUIs).

Some key dates in the development of GUIs:

|  |  |
| --- | --- |
| 1981 | Xerox 8010 Star Information System, the first commercial system to use a GUI. |
| 1983 | Apple Lisa. |
| 1984 | Apple Macintosh. |
| 1985 | Microsoft Windows 1.0, a window manager for DOS |
| 1995 | Microsoft Windows 95, a full graphical OS for PCs. |

For many years, the study of computer usability largely focussed on:

* GUIs and associated input devices for general applications.
* Interfaces for specialist applications.
* Accessibility, i.e., the development of adapted or alternative interfaces for those unable to use GUIs.

Accessibility means giving users with special needs the same level of access as other users.

For example, people who are blind/visually-impaired can operate Graphical User Interfaces with the aid of a *screen-reader*.

However, compared to other users, screen-reader users tend to:

* work more slowly
* make more errors
* report higher levels of fatigue.

Thus, while blind/visually-impaired people can use GUIs, it cannot be said that they have the same level of access as other users.

**The Role of HCI Today**

In recent years, new challenges and possibilities have emerged:

* Legislation has raised standards regarding accessibility, etc.
* The web has changed users' expectations regarding software usability.
* Computers have become smaller and more ubiquitous.
* New interaction technologies have been developed.

Current HCI research includes the development of new/improved interfaces:

* To meet more stringent legal standards (particularly with regard to accessibility), e.g.:
  + EC Directive 90/270
  + Americans with Disabilities Act (ADA)
* To reduce errors in safety-critical applications, e.g.:
  + Military/Aircraft systems
  + Medical systems
  + Plant control (nuclear, chemical, etc.)
* To meet the demands of web developers and users:
  + The web supports increasingly sophisticated applications, but...
  + Users have no committment to individual web applications.
* To support/exploit new technologies:
  + devices with small screens (or no screens)
  + devices with small keyboards (or no keyboards)
  + 3D interaction
  + Gestural and tactile interaction
  + Voice interaction

Much research now concentrates on the design of adaptive/proactive systems:

|  |  |  |
| --- | --- | --- |
| **Reactive Systems**   User always initiates actions   Large screens, focus of user attention   Little need for adaptivity |  | **Proactive Systems**   System or User can initiate actions   No-screen/hands-free, user attention elsewhere   Speech/multimodal output   Adaptivity essential for effective operation |

**What Makes a Good Interface?**

Some definitions:

"A user interface should be so simple that a beginner in an emergency can understand it within ten seconds." (Theodor Nelson, 1965)

"...any application designed for people to use should be:

* easy to learn (and remember),
* useful, that is, contain functions people really need in their work, and
* be easy and pleasant to use."

(Gould & Lewis, 1985)

Four Key Concepts (Shakel, 1990):

* **Learnability** - the time and effort required to reach a specified level of user performance
* **Throughput** - tasks accomplished, speed of execution, errors made, etc.
* **Flexibility** - the extent to which the system can accommodate changes to the tasks and environments beyond those first specified
* **Attitude** - the attitude engendered in users by the application.

It is often said that we should aim to design systems that are 'user friendly'.

But how do we:

* identify the users?
* determine what will make the interface appropriate for them?
* test to see if we have succeeded in our aims?

People vary in their requirements depending upon age, education, level of computer-literacy, etc.

* The requirements of a young gaming-enthusiast may differ considerably from those of a retired person who uses a computer in connection with their hobby.
* An interface that uses graphics to present complex data may be impossible for a blind person to use.
* An individual's requirements may change as they learn more about computers generally or gain experience with a particular system.

The conditions under which tasks are carried out may affect the choice of hardware, interaction-style, etc.

* An online ticketing system may use standard interaction techniques, while a ticket-machine may need to be weather and vandal-resistant.
* A gaming-enthusiast may prefer a mouse with multiple buttons, scroll-wheels, etc., but such a mouse would be highly unsuitable for the casual/occasional users of an information kiosk.
* An individual may interact differently with an application depending upon whether they are using it for a primary or secondary task.
* A user may give a work-related task their full concentration, but when using an ATM may also have to think about security, controlling children, avoiding pedestrians, etc..

A system may have to embody a considerable amount of specialist knowledge.

For example, the developers of a computerised hospital-records system may have to take into account a great deal of specialist medical knowledge in order to design an effective system.

In many projects, particularly larger ones, there is a potential gulf between:

* system developers, who have knowledge concerning computer systems
* end-users of the system, who have knowledge of the *domain* for which the system is to be used.

In some cases, there may be an additional potential gulf between:

* primary end-users, who operate the system
* secondary end-users, on whose behalf the system is used.

For example:

* a travel-agent (*primary user*) may use a system to search for flights on behalf of...
* a customer (*secondary user*)

One of the primary goals of HCI is to reduce or elminate the gulf(s) that may exist between developers and end-users.

**Key HCI skills/knowledge**

* Interaction styles
* Interaction technology
* Human psychology/perception/etc.
* Design/Evaluation techniques
  + Guidelines
  + Metrics
  + Modelling
  + Iterative design and evaluation

Interaction Styles:

* Command-language
* Form-filling
* Menu-selection
* Function keys
* Question and Answer
* Direct-manipulation
* Anthropomorphic/Natural Language

Interaction Technologies:

* Input
  + Pointing/selection devices (mouse, joystick, touch-screen, eye-gaze, etc.)
  + Text-entry technology (keyboard, keypad, speech, etc.)
* Output
  + Visual (Text & Graphics)
  + Audio
  + Tactile devices

Human Psychology and Perception:

* Visual perception
* Auditory perception
* Tactile perception
* Memory
* Etc.

Guidelines:

* Web Content Accessibility Guidelines
* Eight Golden Rules (Shneiderman)
* Guidelines for UI software (Smith & Mosier)
* OSF Style Guide
* GUI Bloopers (Johnson)
* Human Interface Guidelines (Apple)
* Speech Interface Guidelines (Larson)
* etc..

Metrics:

* SUMI, WAMMI, etc
* Accessibility validators, e.g., WebXact

Modelling Techniques:

* Task Analysis
  + Hierarchical Task Analysis
  + Knowledge-Based Analysis
  + Entity-Relationship Techniques
* Cognitive modelling
  + Task-Goal Hierarchies
  + Linguistic/Grammatical Models
  + Device-Level Models

Design and Evaluation Techniques:

* Questionnaire design
* Experimental design
* Statistical analysis
* Iterative design and evaluation