



Human Computer Interaction

1. Introduction

- HCI covers a broad subject area which is concerned with the **interaction between people and computers**. HCI is a **multidisciplinary** subject encompassing Computer Science, Psychology, Linguistics, Ergonomics and Sociology.
- The emphasis in this course concerns those aspects of HCI which are the **concern of the software designer**.

2 The importance of the User Interface

- The user interface is the main contact between the user and the computer system, it is the part of the system that the user sees, hears, touches and communicates with.
- The user interacts with the computer in order to carry out some specific task which is of importance, often the task will be a fundamental or critical part of the user's job.

- The types of problems caused by poor interface design include **reduced user productivity**, **unacceptable learning times** and **unacceptable error levels**, all of these factors will **lead to user frustration** and potentially to **rejection of the system by the user**.

3. User Interface Design:

- What makes a 'good' interface?
- Well-designed interfaces provide a **good match between the user's task, skill level, and learning ability** and will lead to **satisfied and productive** users
- A good interface will be **easy to learn and easy to use**; it will **encourage the user to experiment and try out new features** within the system without getting frustrated.

- In order to understand what is meant by a ‘good’ or ‘well designed’ user interface, we need to have some understanding of the **classes of user interface** commonly available and of their **appropriateness for given situations**.

3.1 Classes of User Interface

■ Command Language:

- ◆ these are **dialogues** in which the user types instructions to the computer in a **formally defined command language**, e.g. `mv file file2`, in UNIX for copying file1 into file2.
- ◆ This type of interface is **very flexible**, allowing users to **create their own commands**.
- ◆ The interface requires **significant level of training** and a **high degree of memorization**.

■ Natural Language:

- ◆ these are interfaces in which the user's **command language** is a significant, well-defined subset of some natural language such as English
- ◆ They are typically **easy to learn**; however they often require considerable **typing skills on the part of the user**

Menu System

- these interfaces allow the user to issue commands by selecting choices from a menu of displayed alternatives.
- They are popular since they reduce learning time, reduce the number of key strokes necessary and help to structure decision making.

Form Filling Dialogues:

- ◆: the user enters data by filling in fields in one or more forms displayed on the screen.
- ◆ The use of forms on the screens considerably simplifies data entry and requires very little training to use

■ Direct manipulation Interfaces

- ◆ the user manipulates, through button pushes and movements of a pointing device such as a mouse, graphic or iconic representation of the underlying data.
- ◆ Direct Manipulation Interfaces represent task concepts visually, are easy to learn and use, they encourage exploration and or experimentation with the system features, and generally result in a high level of user satisfaction.

- Choosing the most appropriate class of user interface to **match the needs and expectations of the users** is an important aspect of good user interface design
- To **assist in making the right decisions** and hence **achieving a good user interface design**, a number of design guidelines are available

3.2 *Design Guidelines*

- The major guidelines common to many of the existing texts can be summarized into five categories:
- **Naturalness:**
 - ◆ Dialogue which does not cause the user to significantly alter his or her approach to the task in order to interact with the system
 - ◆ The ordering of the user input is important; it should be geared towards the normal order of working of the user rather than whatever is easier for the programmer

- ◆ **Phrasing should be self explanatory**, e.g. print, copy end have **obvious meanings** whereas 'pip' (CP/M – control program for microcomputers- keyword for copy) or 'mv'(UNIX keyword for rename) do not.
- ◆ Use of **non-standard abbreviations** should be avoided since they **slow down word recognition** and introduce unnecessary stress.

Involves:

- : Consistency

- ◆ : a consistent dialogue ensures that expectations which the user builds up through using one part of the system are not frustrated by idiosyncratic changes in the conventions used in another part
- ◆ Consistent layout for screens which fulfill a similar function ensures that the user knows where to look for instructions, error messages etc

- ◆ The dialogue should also be **consistent with established norms**.
- ◆ Diversions from norms will cause confusion

■ **Non-Redundancy:**

- ◆ a non redundant dialogue requires the user to **enter only the minimum information** for the system's operation

- ◆ Too much information on one screen is detrimental to the clarity of the screen and will lead to unnecessarily delay when the user tries to 'spot' a particular field or item.

- **Supportiveness:**

- ◆ Refers to the amount of assistance which the dialogue provides to the user in running the system.

- ◆ It has three major aspects: the quantity and quality of instructions provided; the nature of the error messages produced and the confirmation of what the system is doing.
- ◆ Inputs should be confirmed: if their acceptance will result in an irreversible action, for example, delete file; if a code has been entered and the user has to check the associated description or when confirmation of completion of particular actions is desirable.

•Flexibility:

- This depends largely on the skill and expertise of the user in relation to a given task. Different types of dialogue may be used in different situations e.g. a hierarchical menu structure for use by a first time user may be navigated using commands and parameters once the user becomes more experienced.

3.3 WIMP User Interfaces

- A WIMPS interface is one which: **W** – presents information to users via multiple **W**indows on the display screen. **I** represents data objects as **I**cons, **M** – uses a **M**ouse as a selection device, **P** – has menus which **P**op-up automatically on the screen or which **P**ull-down from a menu bar at the top of the screen.

3.3.1. Advantages of Multiple Windows

- ◆ Allow access to multiple sources of information
- ◆ Allow information to be viewed from different perspectives
- ◆ The user may examine the same information at different levels of detail, for example, overviews in one window with related windows containing further detail

- ◆ Allows the **system** attract the user's **attention** for example by **displaying a new window in the middle of the screen** e.g. a window with a warning message
- ◆ Allow the **user to control multiple concurrent tasks** in an environment where **multitasking** is provided for

3.3.2 Disadvantages of Windows

- ◆ The danger of 'overcrowding' on the screen
- ◆ Distraction from the task in hand by causing the user to manipulate the interface in order to obtain the information required
- ◆ The desk-top metaphor becomes the untidy desk top' metaphor

3.3.3 ICONS

- An icon is a **small window** that provides a **pictorial representation of the contents of the information** which is **accessible via the icon**.
- Most WIMPS interfaces allow designers to **specify their own icons**, they also provide a set of **predefined icons**.

3.3.3.1 Advantages of Icons

- ◆ An icon does not occupy much space on the screen, but, can serve as a reminder of a potentially large amount of information
- ◆ Icons can be designed to match the visual memory of the user
- ◆ They can be used as part of an analogy that closely matches the user model of behavior to real world objects-emoticons???

3.3.3.2 Disadvantages of Icons

- ◆ Potential for ambiguity of the meaning of Icons
- ◆ The meaning of the icon will depend on the user and the user environment (context –dependent, context is important)

3.3.3.3 Design of Icons

- Many methods advocate for **user participation** in the design of the icon
- Some recommend **getting the user to 'doodle'** in an effort to **facilitate visualization of some aspect of the user environment**

Design Steps

- Test the representation of the icons with the users
- Make icons as realistic as possible
- Give the icon a clear outline to help visual discrimination
- When showing commands give a concrete representation of the object being operated upon (instead of an icon)
- Avoid symbols unless their meaning is already known

3.3.3.4 Classifications of Icons

- Rogers(1989) distinguishes between **form** and **function** of icons.
 - ◆ **Four types of according to form:**
 - ☞ **Resemblance icons:** which depict the underlying referent through an analogous image
 - ☞ **Exemplar icons:** which depict a typical example of a general class of objects, for example an icon of a knife and a fork depicts the availability of food.

- 👉 **Symbolic icons:** which convey the underlying referent at a higher level of abstraction than the image itself, e.g. a 'wine glass' with a jagged crack depict fragile
- 👉 **Arbitrary icons:** which bear no resemblance to the referent, for example the representation of a biohazard



Icons types according to functions:

- **Labeling** e.g. a menu item
- **Indicating** e.g. indicating a system state
- **Warnings** e.g. warning of an error message
- **Identifying** e.g. file storage
- **Manipulating** e.g. icon representing a tool for zooming and shrinking
- **Container** e.g. for placing discarded objects

3.3.3.5 Underlying Analogies

- Wimp interfaces were initially targeted at the **office system users**. In an attempt to provide an interface that was natural, consistent and supportive of such users, **designers looked for analogies** in the clerical performance of similar tasks.
- Three examples of analogies within the WIMP philosophy are: the “Concrete Object” Metaphor; the “Desktop” metaphor; the “Travel Holiday’ Metaphor.

- **The “Concrete Object” Metaphor** – the **items here represent the physical (concrete) object**. Thus files are presented as sheaves of paper or folders, directories are presented as drawers in a filing cabinet. **Clerical operations involve physical actions on these objects.**
- **The “Desktop” Metaphor:** suggests that the interface should provide the user with a **similar type of flexibility as the desktop**, e.g. **have access to several information sources**, see a **variety of formats** such as pictures, graphs, etc.

- ◆ The desktop metaphor will **not be appropriate for all application types**. The desire is to provide **an interface that is natural, consistent**, etc, and which is also **appropriate for the class of users /tasks under consideration**. We may well need to **identify different types of icons**, different **analogies** depending on the situation.
- ◆ In addition, the **icons were organized into three classes** with each possessing it's own properties: **container icons** such as folders or baskets; **data icons** such as charts, spreadsheets or documents, and **device icons** such as printers or telephone.

- **The “Travel Holiday” Metaphor:** this was developed by Hammond and Allison in 1987 for a Computer Aided Learning System (CAL). The aim of the system was to provide the student with a large amount of information together with various means to accessing it. The general metaphor of the travel holiday was used plus display frames representing places to visit, and various facilities representing the ways and means of traveling around. Within the general metaphor, a number of more specific metaphors were assumed; navigation e.g. go it alone travel and guided tours. Second, a map facility allowed students to see where they were and where they had been. The third index (notionally the index of a book) provided a mechanism for key-word-based access.

3.3.4 Design Guidelines for WIMPS

- The contents of a window should form a logically related group
- The borders of each window should be clearly delimited
- Avoid filling the screen with a multiplicity of small windows
- Windows should appear initially in a consistent position and have a consistent size
- The default position and size should be adjusted to reflect user preference

- The contents of each window and of each screen should **reflect a logical ordering**, **consistent format** and utilize **minimum highlighting**
- The **spatial position of each window on the screen** should reflect a **logical ordering e.g.** simple, more complex, most complex in that order
- **Use of colors across the whole screen** should be **minimum** and **consistent**
- **Allow 'popping-up' of windows** to attract user attention
- Avoid complicated **coding of mouse buttons**
- Allow the **use of commands** as an **alternative to the mouse** – to give flexibility according to preferences and level of expertise



■ THE END