

Dublin City University

School of Computing

Graduate Diploma in Information Technology

CA593: User Interface Development

Part 7: Classes of User Interface

Gareth Jones

March 2008

Introduction

Once task analysis has been completed and the user objects, concepts and actions have been identified, the user interface designer must choose how the user will interact with the system.

Interaction is achieved through a dialogue. There are a number of different classes of interactive dialogue and each of these has advantages and disadvantages depending on the situation in which they are used.

- Command Language
- Natural Language
- Menu Systems
- Form Filling Dialogues
- Direct Manipulation Interfaces

Command Language

The user types instructions to the computer in a formally-defined command language.

Command languages have their origin in computer operating systems.

Users must know (recall or lookup!) the notation required to execute a command.

Command languages may have a few commands or many thousands.

Users may not even know that a command is possible - consider the possibilities within UNIX.

Syntax can required a deep understanding of many concepts. The following UNIX command deletes blank lines from a file

```
grep -v ^$ filea > fileb
```

Command Language

Advantages:

- command interpreter is simple to create.
- alphanumeric display is fine.
- relatively little typing effort.
- user can often create their own commands by combination.

Disadvantages:

- need to learn command language, requires significant level of training.
- errors are frequent in command entry (syntax, semantics).
- possibly wrong system state - all options are available at any time.
- input is through keyboard only.

Command Language

Conclusions:

- NOT for casual and inexperienced users (time and effort for learning).
- GOOD for experts (flexibility, speed). Appeals to “power users”.

Design issues:

- possibility to combine commands.
- mnemonic selection (short, meaningful).
- command redefinition may be possible?

Natural Language

Interfaces where the user's command language is a significant, well designed subset of some natural language such as English.

A long standing dream in computing!

Advantages:

- naturalness.
- typically easy to learn - ideally(!) no training.

Natural Language

Disadvantages:

- vague, ambiguous, incorrect entries - can be slow if the system requires clarification of the command.
- considerable typing skills required - and very large typing effort!
- natural language interpretation is very complex computing task.

Can be useful with voice recognition.

Menu Systems

Users issue commands by selecting from a menu of displayed alternatives.

Advantages:

- reduce learning time, no need to memorize commands, only *recognize*.
- can structure user decision making - no system errors (wrong state) especially if disabling is applied.
- minimal typing effort (occasional users).
- context dependent help is available.

Disadvantages:

- some things are difficult with menus (e.g. A & B).
- structuring large set of choices is difficult.
- for expert users it can be slow (shortcuts).

Menu Systems

Design issues:

- using proper menu elements to save screen space (pull-down, pop-up), cascade.
- ordering of items (alphabetical, category, frequency, conventional).
- how to select (type, point, move cursor).
- how to navigate (hierarchy, ways to exit at wrong sub-menu).
- how to find desired option:
 - logical grouping of options.
 - order menu items: alphabetical, categorical, conventional (e.g. days of week), frequency.

Form Filling Dialogues

The user enters data by filling in fields in one or more forms displayed on the screen. Used when several different categories of data have to be entered through the keyboard.

Screen is usually a “copy” of the paper form.

Advantages:

- simplifies data entry.
- requires very little (no!) training.
- fast – automatic cursor position possible, low error rate (except typing!).

Disadvantage:

- system driven – human is only assisting.

Form Filling Dialogue

Conclusion:

- Good for inexperienced/non-expert users - little/no experience with computers required.
- Good for repetitive tasks.

Increasingly popular on the WWW.

Note poor layout, choice of user concepts/vocabulary and guidance on many web forms.

Direct Manipulation Interfaces

The user manipulates a graphic or iconic representation of underlying data through button pushes and movements of a pointing device such as a mouse.

Main features of direct manipulation systems:

- visibility of objects of interest (icons - difficult to design good ones).
- rapid, reversible, incremental actions.
- command syntax is replaced by direct manipulation of objects.

Direct Manipulation Interfaces

These systems are generally characterised by:

- multiple **w**indows (to display different information.
- **i**conic representation of information.
- **m**enus for command selection.
- **p**ointing device (mouse) to select object, etc.

WIMP – GUI (graphical user interface)

Direct Manipulation Interfaces

Advantages:

- whole system is (usually) visible.
- novices can learn quickly.
- experts can work rapidly. (shortcuts!)
- basic actions are consistent (open, save, . . .).
- errors are rare.

Direct Manipulation Interfaces

More Advantages:

- immediate feedback, users can see the result of their action (is the right/wrong direction of work?).
- less anxiety: reversible actions, possibility of system exploration; generally high level of user satisfaction.
- user initiates, feels in control: confidence, predict responses.

Disadvantages:

- very difficult to develop.

Question and Answer Dialogues

Can be command line or form based.

Questions asked one at a time. Next question may depend on the answer to the previous one.

Advantages:

- System driven - protects user from considerations of navigation.
- Good for novice users.
- Can be very frustrating for experienced users – inflexible or inconvenient.

Conclusions:

- Good where information is elicited from users in a prescribed and limited form.

Hybrids

Many user interfaces are hybrids of these classes.

e.g. menus, and shortcuts.

Hybrids enable the designer to cater for different levels of user within one system.

Support Users' Tasks

Whatever interface type is chosen, users will choose to use a computer if it gives them facilities not otherwise available.

If the facilities are attractive enough, users will use a system despite a poor user interface.

This means that it is crucial to understand the user and their tasks.

Support Users' Tasks

Common mistakes:

- provide excessive number of objects in the interface, which can overwhelm the user. Also these interfaces:
 - are harder to maintain;
 - may contain more bugs;
 - possibly execute more slowly;
 - require more help information, error screens, manuals;
 - take the user longer to learn;
 - increase chance of error;
 - confusion from longer manuals, complex help, less-specific errors and warning information.

Support Users' Tasks

- Insufficient Objects or Actions:
 - user can be frustrated because a desired function is not supported - this makes it vital that the designer really has performed a full task analysis.