Interface Design Rules

HCI Lecture 10

David Aspinall

Informatics, University of Edinburgh

23rd October 2007

Outline

Principles and Guidelines

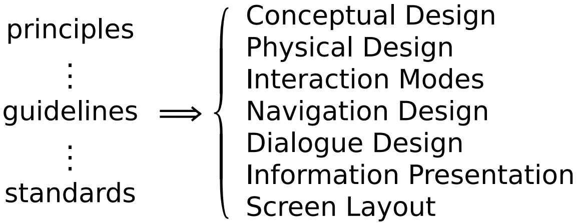
- Learnability
- Flexibility
- Robustness

Other Guidelines

- Golden rules and heuristics
- HCI patterns

Standards

Interface Design Roadmap



principles

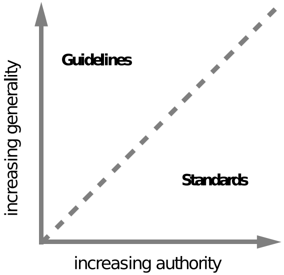
guidelines

standards

Conceptual Design
Physical Design
Interaction Modes
Navigation Design
Dialogue Design
Information Presentation
Screen Layout

- **design rules** have differing generality and operate at various levels. They:
  - complement modelling and evaluation;
  - encapsulate understanding and best practice;
  - help us to design for maximum usability.

Types of Design Rules



- principles
  - abstract design rules
  - “an interface should be easy to navigate”
- guidelines
  - advice on how to achieve principle
  - may conflict; understanding theory helps resolve
  - “use colour to highlight links”
- standards
  - specific rules, measurable
  - “MondoDesktop links are RGB #1010D0”

Outline

Principles and Guidelines

- Learnability
- Flexibility
- Robustness

Other Guidelines

- Golden rules and heuristics
- HCI patterns

Standards

Usability Principles

- **Learnability**
  the ease with which new users can begin effective interaction and achieve maximal performance

Usability Principles

- **Learnability**
  the ease with which new users can begin effective interaction and achieve maximal performance

- **Flexibility**
  the multiplicity of ways the user and system exchange information

Usability Principles

- **Learnability**
  the ease with which new users can begin effective interaction and achieve maximal performance

- **Flexibility**
  the multiplicity of ways the user and system exchange information

- **Robustness**
  the level of support provided to the user in determining successful achievement and assessment of goal-directed behaviour

Learnability (1): Predictability

Predictability — determinism and operation visibility

Learnability (1): Predictability

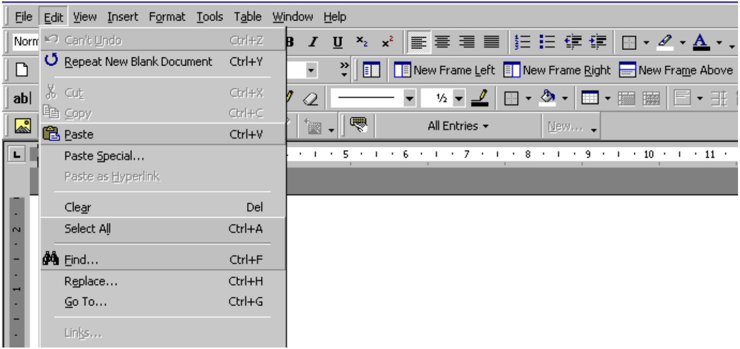
Predictability — determinism and operation visibility

- System behaviour is observably deterministic:
  - *Non-deterministic delays should be avoided*
  - *Operation effect determinable by interaction history*

Learnability (1): Predictability

Predictability — determinism and operation visibility

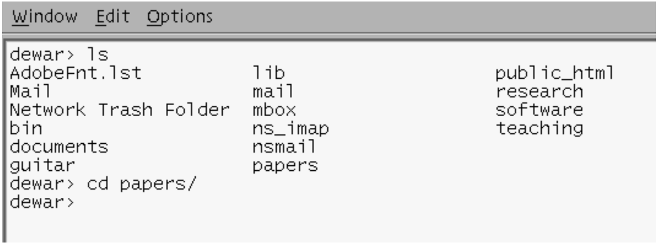
- System behaviour is observably deterministic:
  - *Non-deterministic delays should be avoided*
  - *Operation effect determinable by interaction history*
- operation visibility:
  - *user actions should be matched by a response*
  - *affordance/logical constraints should be used to indicate available actions*:



Learnability (2): Synthesisability

Synthesisability — can assess effect of past actions

- Direct Manipulation interfaces promise *immediate honesty*
- Others have *eventual honesty*
- Command line interfaces are never honest:



```
dewar> ls
AdobeFnt.lst           lib                    public_html
Mail                   mail                   research
Network Trash Folder   mbox                   software
bin                    ns_imap                teaching
documents              nsmail
guitar                 papers
dewar> cd papers/
dewar>
```

Learnability (3): Familiarity

Familiarity — matching users’ expectations

- how prior knowledge applies to new system
  - *guessability* of the system
- knowledge of task and of other systems
- use of metaphor (e.g. tab-stops in word-processor)
- use of natural language syntax, affordances
  - *regions on the screen which denote buttons should be shaded to give a three-dimensional appearance*

Learnability (4): Consistency

Consistency — likeness in input/output behaviour arising from similar situations or task objectives

Learnability (4): Consistency

Consistency — likeness in input/output behaviour arising from similar situations or task objectives

- probably the most widely mentioned principle: “Be consistent!”

Learnability (4): Consistency

Consistency — likeness in input/output behaviour arising from similar situations or task objectives

- probably the most widely mentioned principle: “Be consistent!”
- challenge (and danger): consistency not self-contained
  - consistency within screens
  - consistency within applications
  - consistency within desktop
  - . . .

Learnability (4): Consistency

Consistency — likeness in input/output behaviour arising from similar situations or task objectives

- probably the most widely mentioned principle: “Be consistent!”
- challenge (and danger): consistency not self-contained
  - consistency within screens
  - consistency within applications
  - consistency within desktop
  - . . .
- Examples: consistent patterns in layout; same short-cut keys for similar action; same placement for recurrent menu options
  - *Always place the Quit command as the last item in the leftmost menu*

Learnability (5): Generalizability

Generalizability — extending specific interaction knowledge to new situations

Learnability (5): Generalizability

Generalizability — extending specific interaction knowledge to new situations

- helps give a predictive model of system for user
- a form of consistency

Learnability (5): Generalizability

Generalizability — extending specific interaction knowledge to new situations

- helps give a predictive model of system for user
- a form of consistency
- examples:
  - drawing circles → drawing ellipses

Learnability (5): Generalizability

Generalizability — extending specific interaction knowledge to new situations

- helps give a predictive model of system for user
- a form of consistency
- examples:
  - drawing circles → drawing ellipses
- UI standards and guidelines assist/enforce generalizability
  - *applications should offer the Cut/Copy/Paste operations whenever possible*

Flexibility (1) : Dialogue initiative

Dialogue initiative — who controls dialogue flow

Flexibility (1) : Dialogue initiative

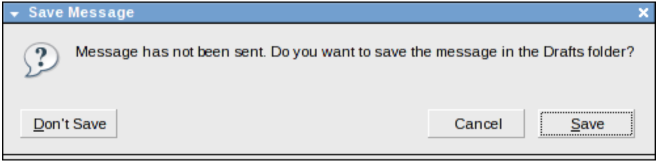
Dialogue initiative — who controls dialogue flow

- freedom from system imposed constraints on input dialogue
- *user should be able to abandon, suspend or resume tasks at any point*

Flexibility (1) : Dialogue initiative

Dialogue initiative — who controls dialogue flow

- freedom from system imposed constraints on input dialogue
- *user should be able to abandon, suspend or resume tasks at any point*
- modal dialog boxes are *system pre-emptive*
- direct manipulation is *user pre-emptive*
- *minimise system pre-emptive dialogue and maximise user pre-emptive dialogue*



Message has not been sent. Do you want to save the message in the Drafts folder?

Flexibility (2): Multi-threading

Multi-threading — support simultaneous tasks

Flexibility (2): Multi-threading

Multi-threading — support simultaneous tasks

- concurrent vs. interleaving; multimodality

Flexibility (2): Multi-threading

Multi-threading — support simultaneous tasks

- concurrent vs. interleaving; multimodality
- *provide multiple task contexts*

Flexibility (3): Task migratability

Task migratability — how easily functions can be moved between user and system

Flexibility (3): Task migratability

Task migratability — how easily functions can be moved between user and system

- People get bored doing routine tasks and stop concentrating (Yerkes-Dodson Law)
- *automate routine tasks, but don't fix function allocation*

People are better at	**Machines are better at**
Detecting small sensory inputs	Responding quickly to signals
Improvising and using flexible procedures	Following procedures repeatedly and precisely
Reasoning inductively	Reasoning deductively
Selective information recall	Total information recall
Exercising judgement	Following orders

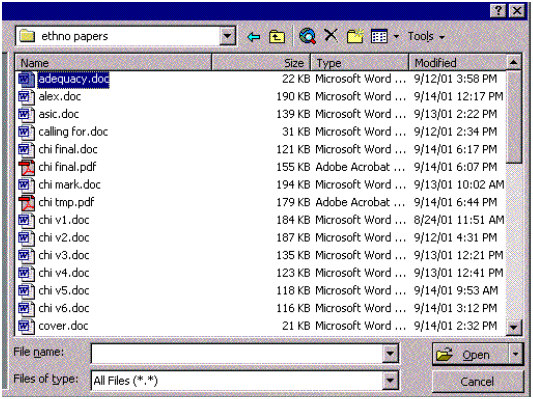
Flexibility (4): Substitutivity

Substitutivity — allowing equivalent values of input and output to be substituted for each other

Flexibility (4): Substitutivity

Substitutivity — allowing equivalent values of input and output to be substituted for each other

- representation multiplicity; equal opportunity
- *don't force users to refer to objects by name if they can point to them*



Name	Size	Type	Modified
adequacy.doc	22 KB	Microsoft Word ...	9/12/01 3:58 PM
alex.doc	190 KB	Microsoft Word ...	9/14/01 12:17 PM
asic.doc	139 KB	Microsoft Word ...	9/13/01 2:22 PM
calling for.doc	31 KB	Microsoft Word ...	9/12/01 2:34 PM
chi final.doc	121 KB	Microsoft Word ...	9/14/01 6:17 PM
chi final.pdf	155 KB	Adobe Acrobat ...	9/14/01 6:07 PM
chi mark.doc	194 KB	Microsoft Word ...	9/13/01 10:02 AM
chi tmp.pdf	179 KB	Adobe Acrobat ...	9/14/01 6:44 PM
chi v1.doc	184 KB	Microsoft Word ...	8/24/01 11:51 AM
chi v2.doc	187 KB	Microsoft Word ...	9/12/01 4:31 PM
chi v3.doc	135 KB	Microsoft Word ...	9/13/01 12:21 PM
chi v4.doc	123 KB	Microsoft Word ...	9/13/01 12:41 PM
chi v5.doc	118 KB	Microsoft Word ...	9/14/01 9:53 AM
chi v6.doc	116 KB	Microsoft Word ...	9/14/01 3:12 PM
cover.doc	21 KB	Microsoft Word ...	9/14/01 2:32 PM

File name:

Files of type: All Files (*.*)

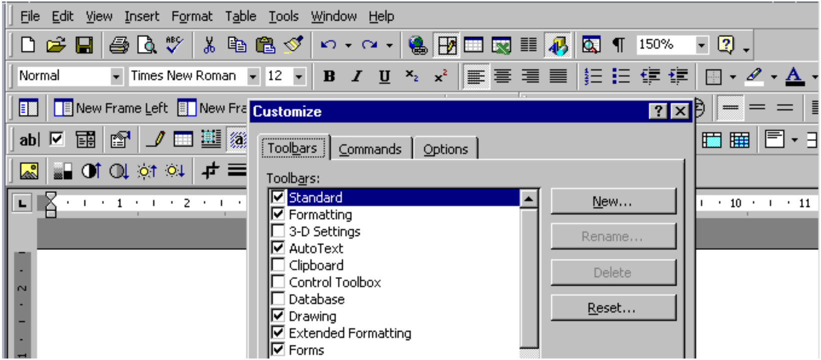
Flexibility (5): Customisability

Customisability — interface can be adapted to suit different needs

Flexibility (5): Customisability

Customisability — interface can be adapted to suit different needs

- modifiability of the user interface by user (adaptability) or system (adaptivity)
- *provide choice of methods*; *allow short-cuts*; *permit users to change features: deferred design*.



Toolbars:

☑ Standard
☑ Formatting
□ 3-D Settings
☑ AutoText
□ Clipboard
□ Control Toolbox
□ Database
☑ Drawing
☑ Extended Formatting
☑ Forms

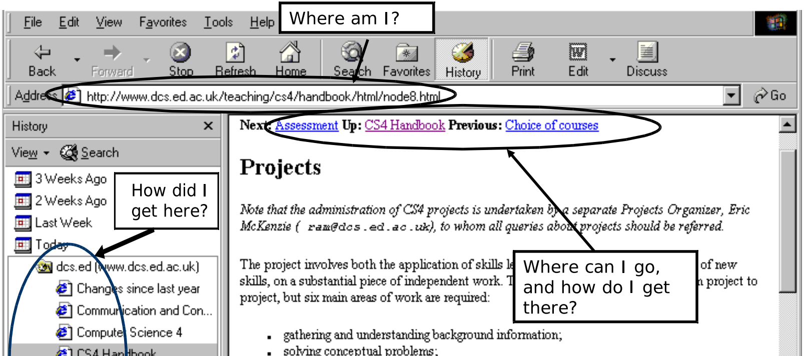
Robustness (1): Observability

Observability — user impression of system state

Robustness (1): Observability

Observability — user impression of system state

- user should be able to evaluate the internal state of the system from its perceivable representation
- E.g., *Where*^{3}*What* of navigation:
  - **W**here am I? — immediate honesty wrt system state
  - **W**here am I going? — operation predictability
  - **W**here have I been? — synthesisability
  - **W**hat can I do now? — predictability



Where am I?

How did I get here?

Where can I go, and how do I get there?

Next: Assessment **Up:** CS4 Handbook **Previous:** Choice of courses

Projects

Note that the administration of CS4 projects is undertaken by a separate Projects Organizer, Eric McKenzie (ram@dcs.ed.ac.uk), to whom all queries about projects should be referred.

The project involves both the application of skills l of new skills, on a substantial piece of independent work. T project to project, but six main areas of work are required:

- gathering and understanding background information;
- solving conceptual problems;

Robustness (2): Recoverability

Recoverability – support for undoing errors

Robustness (2): Recoverability

Recoverability – support for undoing errors

- ability of user to take corrective action once an error has been recognized
- reachability
  - *user should be able to undo back to any point*
- supported by reducing scope for making errors
  - *avoid free-form input where possible*
  - *validate input immediately, allowing correction*
- . . . and ability of user to understand errors
  - *error messages should be concise, informative, specific, constructive*

Poor:

Error 404: document not found

Better:

The requested URL
`http://www.foobar.com/bar.html`
could not be found

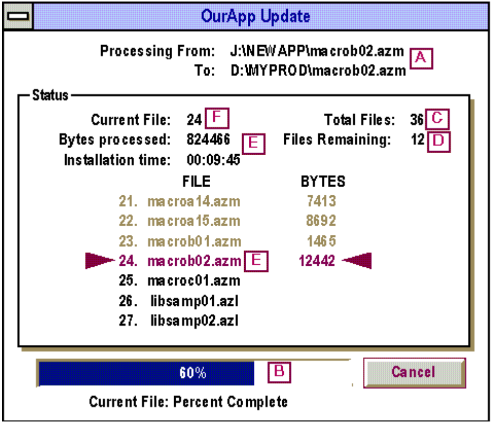
Robustness (3): Responsiveness

Responsiveness — feedback should be commensurate with action

- Sensitivity to delay depends on context
- *Echoing input < 0.1 secs*, *page turning < 0.5 secs*, *string search < 4 secs*

If delay is inevitable, provide reassurance: **time affordances** (Alex Paul Conn (1995))

- A acceptance
- B initiation and heartbeat
- C progress (fine-grained)
- D scope and remainder
- E exception
- F progress and completion



OurApp Update

Processing From: J:\NEWAPP\macrob02.azm
To: D:\MYPROD\macrob02.azm

Status

Current File: 24
Bytes processed: 824466
Installation time: 00:09:45

Total Files: 36
Files Remaining: 12

	FILE	BYTES
21.	macroa14.azm	7413
22.	macroa15.azm	8692
23.	macrob01.azm	1465
24.	macrob02.azm	12442
25.	macroc01.azm	
26.	libsamp01.azl	
27.	libsamp02.azl	

Current File: Percent Complete

Robustness (4): Task conformance

Task conformance — degree to which the system supports the user’s tasks

Robustness (4): Task conformance

Task conformance — degree to which the system supports the user’s tasks

- Few general purpose commands, long methods, simple
- Many highly tuned commands, short methods, complex
- *identify core tasks; provide a command for each*
- But core task set grows over time; language is cluttered as lexicon expands
  - e.g., Unix command language, once a small set now has >700; 10% account for 90% of usage.
  - Microsoft Word command lexicon now includes text formatting, drawing, annotating, WWW related commands, etc.

Outline

Principles and Guidelines

- Learnability
- Flexibility
- Robustness

Other Guidelines

- Golden rules and heuristics
- HCI patterns

Standards

Golden rules and heuristics

- “Broad brush” design rules
- Useful check list for good design
- Better design using these than using nothing!

Golden rules and heuristics

- “Broad brush” design rules
- Useful check list for good design
- Better design using these than using nothing!
- Different collections e.g.:
  - Norman’s 7 Principles (see Lecture 4)

Golden rules and heuristics

- “Broad brush” design rules
- Useful check list for good design
- Better design using these than using nothing!
- Different collections e.g.:
  - Norman’s 7 Principles (see Lecture 4)
  - Nielsen’s 10 Heuristics (used in *Heuristic Evaluation*)

Golden rules and heuristics

- “Broad brush” design rules
- Useful check list for good design
- Better design using these than using nothing!
- Different collections e.g.:
    - Norman’s 7 Principles (see Lecture 4)
    - Nielsen’s 10 Heuristics (used in *Heuristic Evaluation*)
    - Shneiderman’s 8 Golden Rules:
        1. Strive for consistency
        2. Enable frequent users to use shortcuts
        3. Offer informative feedback
        4. Design dialogs to yield closure
        5. Offer error prevention and simple error handling
        6. Permit easy reversal of actions
        7. Support internal locus of control
        8. Reduce short-term memory load

HCI design patterns

- An approach to reusing knowledge about successful design solutions. Originated in architecture (Alexander).
- A pattern is an invariant solution to a recurrent problem within a specific context.
- Examples:
  - Light on Two Sides of Every Room (architecture)
  - Go back to a safe place (HCI)
- Patterns do not exist in isolation but are linked to other patterns in a *pattern language* which enables complete designs to be generated

Outline

Principles and Guidelines
- Learnability
- Flexibility
- Robustness

Other Guidelines
- Golden rules and heuristics
- HCI patterns

Standards

Standards

- Set by national or international bodies to ensure compliance by a large community of designers
- Standards require sound underlying theory and slowly changing technology
- Hardware standards more common than software high authority and low level of detail

Standards

- Set by national or international bodies to ensure compliance by a large community of designers
- Standards require sound underlying theory and slowly changing technology
- Hardware standards more common than software high authority and low level of detail
- **ISO 9241**, *Ergonomics of Human System Interaction*, adopts traditional usability categories:
  - **effectiveness**
    can you achieve what you want to?
  - **efficiency**
    can you do it without wasting effort?
  - **satisfaction**
    do you enjoy the process?

Example metrics from ISO 9241

Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Efficiency relative to expert user	Rating scale for ease of learning
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for ease of learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling

References

Alex Paul Conn.
Time affordances: the time factor in diagnostic usability heuristics.
In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp 186–193, 1995. ACM.

See also:

- Dix et al, Chapter 7.
- More on HCI patterns: `http://www.hcipatterns.org/` (IFIP group), `http://designinginterfaces.com` (O'Reilly book by Tidwell).

Credits: some slides are due to Jon Oberlander, others are from web resources for Dix et al (see `http://www.hcibook.com`).