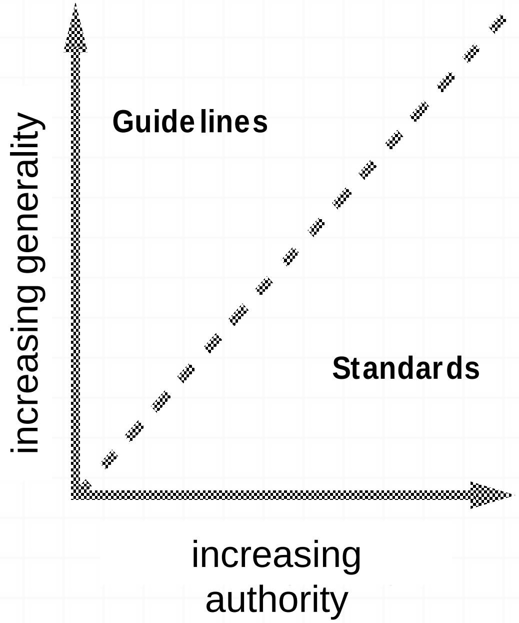
Design Rules

Chapter 7

Design Rules

- Design rules
  - suggest how to increase usability
- Principles
  - abstract design rules
  - an interface should be easy to navigate
- Guidelines
  - advice on how to achieve principle
  - use colour to highlight links
- Standards
  - specific rules, measurable



Principles of Usability

- Learnability:

  the ease with which new users can begin effective interaction and achieve maximum performance

- Flexibility:

  the variety of ways the user and system exchange information

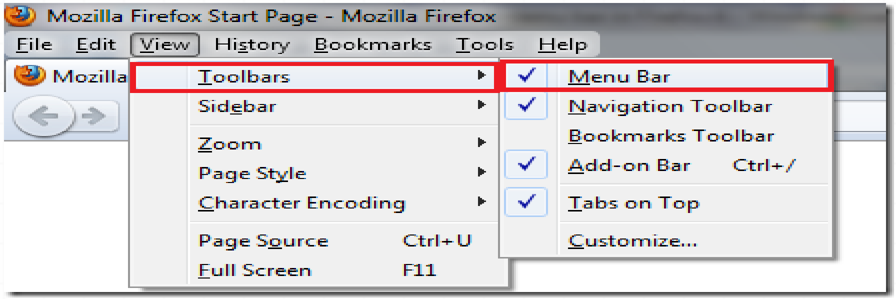
- Robustness:

  the level of support provided to the user in determining successful achievement and assessment of goal

Principles of Learnability

- Predictability
  - Predictability of an interactive system means that the user’s knowledge of the interaction history is sufficient to determine the result of future interaction.(can I ‘tell’ what will happen based on what I have gone

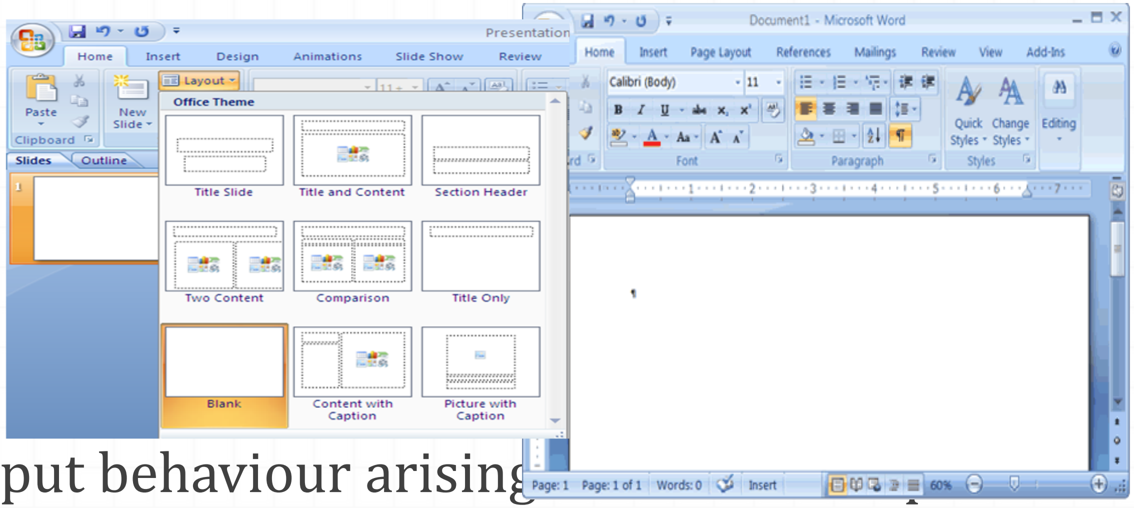
    through in past)
- Familiarity
  - how prior knowledge from real world applies to new system
  - The appearance of the object stimulates a familiarity with its behavior (like a button on door & form)
  - Guessability



Principles of Learnability

Generalizability

- Applying specific interaction knowledge to new situations like MS word & excel etc
- Can be seen as a form of consistency.
  - applications should offer the Cut/Copy/Paste operations whenever possible



Consistency

- Likeness in input - output behaviour arising
- The user relies on a consistent interface

Principles of Flexibility

- Dialogue initiative
  - Who controls dialouge. freedom from system imposed constraints on input dialogue
  - a system-driven interaction hinders flexibility whereas a user-driven interaction favours it.
  - user should be able to Cancel, suspend or resume tasks at any point
- Multithreading
  - ability of system to support user interaction for more than one task at a time
- Task migratability
- Transfer responsibility for task execution between user and system like spell-checking.

Principles of flexibility (ctd)

Substitutivity:

- allowing equivalent values of input and output that be substituted for each other
  - Example: Margin settings provide measurement in inches & centimetres

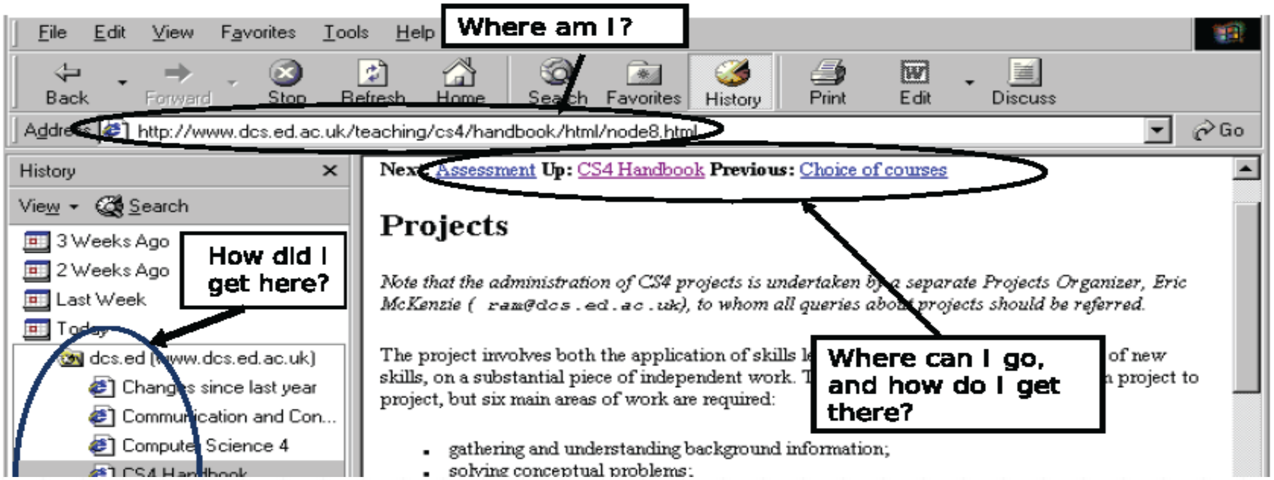
Customizability

- Interface can be modified to different needs
  - Colours & layout change
  - provide choice of methods; allow short-cuts; permit users to change features: deferred design

Principles of Robustness

Observability:

- ability of user to evaluate the internal state of the system from its apparent representation
  - E.g., Where What of navigation:
  - Where am I? — immediate honesty wrt system state
  - Where am I going? — operation predictability
  - Where have I been? — synthesizability
  - What can I do now? — predictability



Where am I?

How did I get here?

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

http://www.dcs.ed.ac.uk/teaching/cs4/handbook/html/node8.html

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 a project to project, but six main areas of work are required:

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

Principles of Robustness

- Recoverability
  - Support for Undoing errors
  - ability of user to take corrective action once an error has been recognized
  - Error messages should be concise, informative, Specific & constructive

Principles of Robustness (ctd)

Responsiveness

- Measures the rate of communication between the system and user
- Response time is generally defined as the duration of time needed by the system to express state changes to the user
- short durations and immediate response times are desirable

Task conformance/Completeness:

- degree to which system services support user's tasks
- System should perform all the tasks that user needs or wants

Standards

- set by national or international bodies to ensure compliance by a large community of designers
- Standards can apply specifically to either the hardware or the software used to build the interactive system
  - software standards are based on theories from human psychology or cognitive science, which are less well formed & vary
  - hardware standards are relatively stable and requirements changes for hardware do not occur as frequently as for software
  - Standards are more suitable for hardware than software.

Golden rules and heuristics

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

Golden rules and heuristics

Norman's 7 Principles:

1. Use both knowledge in the world and knowledge in the head.

2. Simplify the structure of tasks.

3. Make things visible: bridge the gulfs of Execution and Evaluation.

4. Get the mappings right.

5. Exploit the power of constraints, both natural and artificial.

6. Design for error.

7. When all else fails, standardize.

HCI design patterns

- An approach to reusing knowledge about successful design solutions
- A pattern is an invariant solution to a recurrent problem within a specific context.
- Examples
  - Non-reversible actions like paying or destroying something), need to show the user a Warning Message
  - Users do actions they later want reverse because they realized they made a mistake or because they changed their mind.
    - Maintain a list of user actions and allow users to reverse selected actions that is “history of user actions”

HCI design patterns (cont.)

- Characteristics of patterns
  - capture design practice not theory
  - capture the essential common properties of good examples of design
  - represent design knowledge at varying levels: social, organisational, conceptual, detailed
  - embody values and can express what is humane in interface design
  - are intuitive and readable and can therefore be used for communication between all stakeholders
  - a pattern language should be generative and assist in the development of complete designs.