Lecture 5

chapter 7

design rules

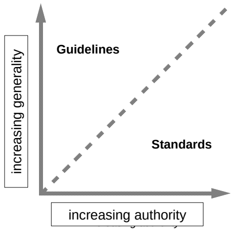
design rules

Designing for maximum usability
– the goal of interaction design

- Principles of usability
  - general understanding
- Standards and guidelines
  - direction for design
- Design patterns
  - capture and reuse design knowledge

types of design rules

- principles
  - abstract design rules
  - low authority
  - high generality
- standards
  - specific design rules
  - high authority
  - limited application
- guidelines
  - lower authority
  - more general application



Principles to support usability

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 the user in determining successful achievement and assessment of goal-directed behaviour

Principles of learnability

Predictability – users don’t like surprises (with exceptions like games)
- determining effect of future actions based on past interaction history
- operation visibility

Synthesizability – requires user to have a mental model (chap 1)
- assessing the effect of past actions
- immediate vs. eventual honesty – changing WYSIWYG doc v updating web pages

Familiarity
- how prior knowledge applies to new system
- guessability; affordance

Generalizability
- extending specific interaction knowledge to new situations

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

Principles of flexibility

Dialogue initiative – system and users in a conversation
- freedom from system imposed constraints on input dialogue
- system vs. user pre-emptiveness
- understanding of main use-cases

Multithreading
- ability of system to support user interaction for more than one task at a time
- concurrent vs. interleaving
- multimodality – button click / alt + / menu item

Task migratability
- passing responsibility for task execution between user and system
- ultimate user control

Substitutivity
- allowing equivalent values of input and output to be substituted for each other
- representation multiplicity (graph/values)
- equal opportunity (define line by drawing or specifying length/position)

Customizability
- modifiability of the user interface by user (adaptability) or system (adaptivity)

Principles of robustness

Observability
- ability of user to evaluate the internal state of the system from its perceivable representation
- browsability; defaults; reachability; persistence; operation visibility

Recoverability
- ability of user to take corrective action once an error has been recognized
- reachability; forward/backward recovery; commensurate effort

Responsiveness
- how the user perceives the rate of communication with the system
- Stability

Task conformance
- degree to which system services support all of the user's tasks
- task completeness; task adequacy

Usability Engineering with Principles

- Principles make good foundations for usability specification components
  - E.g., we could measure Predictability in terms of how often the user has to back-track in performing a task with a prototype
  - We could observe whether the user can successfully Multi-thread two system tasks that appear in a more complex usability scenario
  - We can get subjective impressions of system Responsiveness

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
  - many large organisations have their own standards
- hardware standards more common than software - high authority and low level of detail
- ISO 9241 defines usability as effectiveness, efficiency and satisfaction with which users accomplish tasks

Limitations of Standards

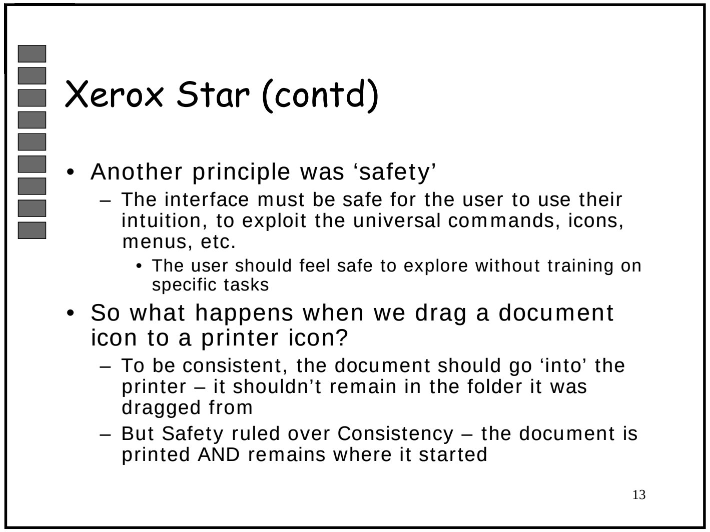
- Difficult for a UI standard to be meaningfully specific and yet actually applicable AND useful for a broad area of application
  - Somewhat more possible to specify *processes* and *principles* than details of what the interface will contain
- However, in a specific context a standard can be very specific
  - E.g., Can have a corporate intranet standard for logos, colours, fonts and other style devices; use (or non-use) of Frames, other layout techniques; etc.
  - Easy for users – they know what to expect
  - Saves lots of work if it's embodies in a nice template or style sheet, too!

Guidelines

- more suggestive and general – less emphasis on 'authority' (more 'should', less 'must')
- many textbooks and reports full of guidelines
- abstract guidelines (principles) applicable during early life cycle activities
- detailed guidelines (style guides) applicable during later life cycle activities
- understanding justification for guidelines aids in resolving conflicts

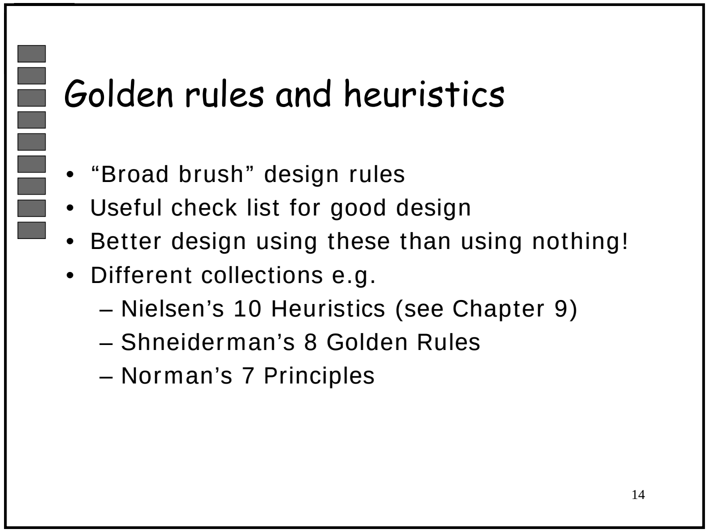
Xerox Star – a guideline conflict story

- At Xerox PARC the Star prototype demonstrated many features of today's Microsoft Office / Windows environment (way back in 1979)
- Design was explicitly (and rather uncompromisingly) guided by principles/guidelines/golden rules
- One of the best was Universal Commands – consistent meaning for options such as double-clicking to 'open', dragging and dropping to move, and dragging to a trash can to delete…



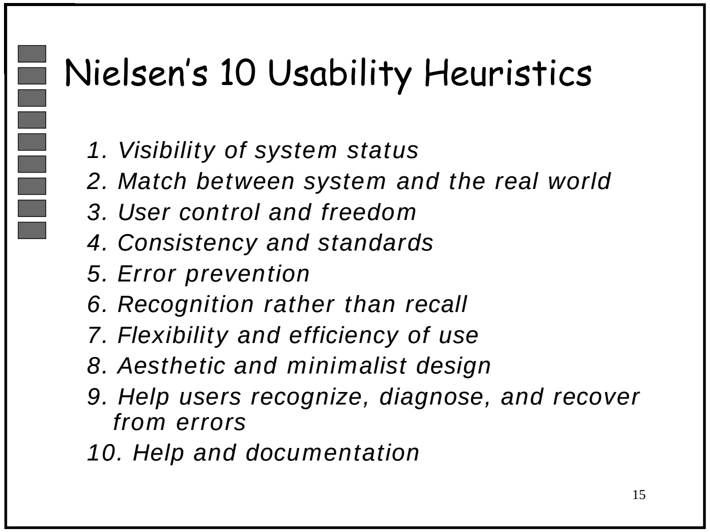
Xerox Star (contd)

- Another principle was 'safety'
  - The interface must be safe for the user to use their intuition, to exploit the universal commands, icons, menus, etc.
    - The user should feel safe to explore without training on specific tasks
- So what happens when we drag a document icon to a printer icon?
  - To be consistent, the document should go 'into' the printer – it shouldn't remain in the folder it was dragged from
  - But Safety ruled over Consistency – the document is printed AND remains where it started



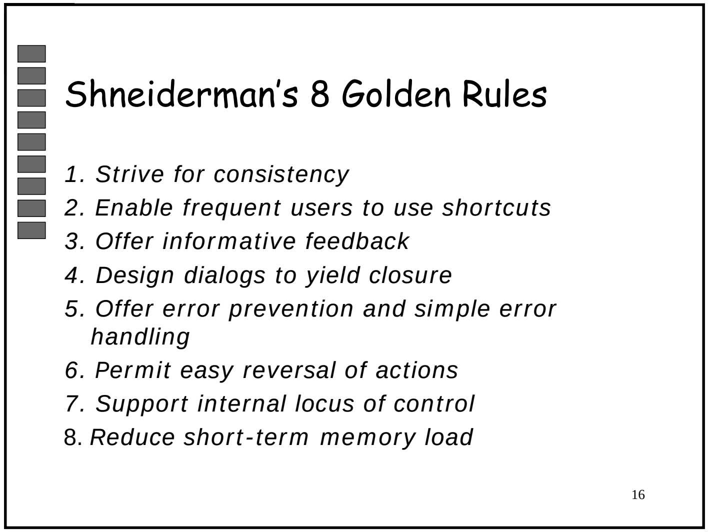
Golden rules and heuristics

- "Broad brush" design rules
- Useful check list for good design
- Better design using these than using nothing!
- Different collections e.g.
  - Nielsen's 10 Heuristics (see Chapter 9)
  - Shneiderman's 8 Golden Rules
  - Norman's 7 Principles



Nielsen's 10 Usability Heuristics

1. *Visibility of system status*
2. *Match between system and the real world*
3. *User control and freedom*
4. *Consistency and standards*
5. *Error prevention*
6. *Recognition rather than recall*
7. *Flexibility and efficiency of use*
8. *Aesthetic and minimalist design*
9. *Help users recognize, diagnose, and recover from errors*
10. *Help and documentation*



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*

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.

Norman (contd.)

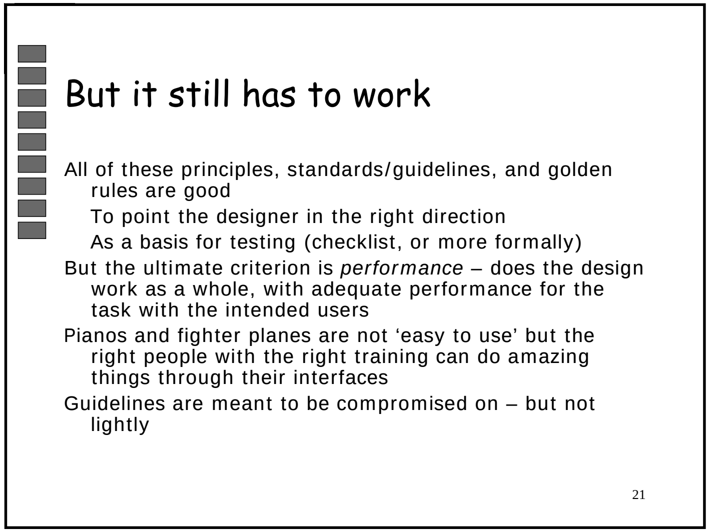
- A random review of Norman's book from http://www.cscs.umich.edu/~crshalizi/reviews/everyday-things/ insightfully says:
  - Norman's principles would almost certainly lead to better design, and they *could* be applied. The one really important question he does not consider is whether they will be. After all, there stand in the way the vanity of designers, their knowledge (which blinds them to the fact that everything is not obvious to J. Random User) and their ignorance (of how people think), the sheep-like quality of users, and (as he notes on the very last page) the imperatives of consumer capitalism. He calls for conscious efforts by designers and consumer revolt, or at least pickiness: how far these will suffice may reasonably be doubted.

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 *languages* which enable complete designs to be generated

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.
- Start point for more info
  - http://www.hcipatterns.org/tiki-index.php



But it still has to work

- All of these principles, standards/guidelines, and golden rules are good
  - To point the designer in the right direction
  - As a basis for testing (checklist, or more formally)
- But the ultimate criterion is *performance* – does the design work as a whole, with adequate performance for the task with the intended users
- Pianos and fighter planes are not ‘easy to use’ but the right people with the right training can do amazing things through their interfaces
- Guidelines are meant to be compromised on – but not lightly