design rules

Designing for maximum usability
– the goal of interaction design

1. Principles of usability: General Understanding
2. Standards: Direction for design
3. Guidelines: Direction for design
4. Functional fixedness and design patterns:
   Capture and reuse of design knowledge
5. Case study: Guidelines for touch-screen design

1 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

1.1 Principles of learnability

Predictability
– determining effect of future actions based on past interaction history
– operation visibility

Synthesizability
– assessing the effect of past actions
– immediate vs. eventual honesty
## Principles of learnability (ctd)

**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

## 1.2 Principles of flexibility

**Dialogue initiative**
- freedom from system imposed constraints on input dialogue
- system vs. user pre-emptiveness

**Multithreading**
- ability of system to support user interaction for more than one task at a time
- concurrent vs. interleaving; multimodality

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

## Principles of flexibility (ctd)

**Substitutivity**
- allowing equivalent values of input and output to be substituted for each other
- representation multiplicity; equal opportunity

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

## 1.3 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

## Principles of robustness (ctd)

**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

## 2 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 defines usability as effectiveness, efficiency and satisfaction with which users accomplish tasks
Standards ISO 9241

System usability comprises the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use, where:

- **Effectiveness** measures the accuracy and completeness with which users achieve specified goals;
- **Efficiency** measures the resources expended in relation to the accuracy and completeness with which users achieve goals;
- **Satisfaction** measures the freedom from discomfort, and positive attitudes towards the use of the product.

http://www.usabilitysa.co.za/usability.htm

3 Guidelines

- more suggestive and general
- 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

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

3.1 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

3.2 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.

Apply Norman’s 7 Principles: NOW

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.
4.1 Functional Fixedness ("idea blocker")

Functional fixedness in problem solving means that the functions or uses assigned to objects tend to remain stable and fixed. One explanation of this phenomenon is that top-down processing is overactive: People retrieve a concept and unfold its functions and uses, without looking for alternatives. There are several examples of functional fixedness, one is the two-ropes problem.

The Two-Ropes Problem

The study of Maier (1931)
The author led his participants into a room with two ropes hanging from the ceiling. Their were other tools in the room, like a hammer, pliers, and extension cords. This task often involves insight and is a demonstration of functional fixedness.
The task was to hold both ropes together:

Solution:
Bind the hammer or another heavy tool to one of the ropes and let them swing so that you can get the other rope.

Overcoming Functional Fixedness - An example

Overcoming functional fixedness may save lives:

Drs. Wallace and Wong, two physicians, learned that a woman suffered from a collapsed lung just after they left Hongkong. The only surgical equipment they had were a rubber tubing and a scalpel. Despite lacking important surgical tools, they operated on the woman and saved her life, with objects that normally have other functions, such as a coathanger, a knife, a fork, and a bottle of Evian water.

(Source: Matlin, 2003)

Patterns: One way to overcome Functional Fixedness

Overcoming functional fixedness may save lives!

Drs. Wallace and Wong, two physicians, learned that a woman suffered from a collapsed lung just after they left Hongkong. The only surgical equipment they had were a rubber tubing and a scalpel. Despite lacking important surgical tools, they operated on the woman and saved her life, with objects that normally have other functions, such as a coathanger, a knife, a fork, and a bottle of Evian water.

(Source: Matlin, 2003)
### 4.2 Functional Fixedness in Software Design

- Rule of thumb for reusable SW: "make components generic".
- SW engineers tend to design in a way that solves the particular problem at hand, but do not easily see opportunities for reuse.
- Such solutions are "functionally fixed".
- Domain analysis prior to system design can help.
- Example: a more reusable Quicksort algorithm.


### 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.

### 5 Case Study

**Towards Guidelines for Touch Screen Design: Perception of Button Size and Height**

Fjeld, Zuberbühler, Guttormsen, Voorhorst, Krueger
ETH Zurich, Switzerland

### Overview

- Motivations and previous reseach
- Experimental set-up
- Observations
- Results
- Design insights
- Future research issues
Motivation

• Touch screen designers generally have few or no guidelines to choose standard solutions.

• Hence, there is a need for guidelines to support designers of touch screen buttons; that is, button form, size, and positioning.

• Related issues are texture and colours used for the background and for buttons.

Previous research by Fred A. Voorhorst

Exp. 1: Classifying surfaces

Exp. 2: Textured buttons

Exp. 3: Contrasting textures

Main conclusion of previous research

When it comes to **perceiving difference** between touch screen buttons

perceived BUTTON SIZE and perceived BUTTON HEIGHT

are of greatest importance.

We decided to explore perceptual thresholds for these two factors.

Experimental set-up 1/2

• Evaluation of design-relevant visual perceptual cues for the design of touch screens

• Button form (round and rectangular) and button extensions (size and height) were examined

• For each of the two forms; with size and height as visual perceptual cues, we estimated perceptual threshold in terms of pixels.

Experimental set-up 2/2

• Within-group design with four conditions

• 12 subjects were presented with 16 varieties of a find-and-select task; 8 tasks with lower time limit (2 seconds) and 8 tasks with higher time (8 seconds) limit

• Estimation of the perceptual threshold by dynamic adaptation using best-Pest method; 50% level

• Test environment was coded in Director/Lingo

Task at typical start-up

Condition (rectangular, height)
Condition (row) and stimulus (column): unmodified; #pixels threshold; task start-up.

Observations 1/3
Perceptual thresholds, box plots

Observations 2/3
Perceptual thresholds, box plots

Observations 3/3
Perceptual thresholds in # pixels

Results
Main conclusion
• Button height should be preferred over button size as perceptual cue, which is equally valid for round and rectangular buttons

Secondary conclusions
• If height is chosen as cue, round buttons should be preferred
• If size is chosen as cue, rectangular buttons should be preferred

Results re-formulated into design insights
1. With equal choice among height and size as cue, height should be the preferred cue.
2. Given height as cue and equal choice among button forms, round buttons should be preferred.
3. Given size as cue and equal choice among button forms, rectangular buttons should be preferred.
Future research issues

1. Which buttons for time-critical contexts of use.
2. Which buttons for safety-critical contexts of use.
3. Textures and colors for the same contexts.

High Precision Touch Screen Interaction
Pär-Anders Albinsson and Shumin Zhai, CHI 2003
Swedish Defence Research, Linköping