Usability Engineering

Design Rules

Carsten Röcker
inIT - Institut für industrielle Informationstechnik
carsten.roecker@th-owl.de

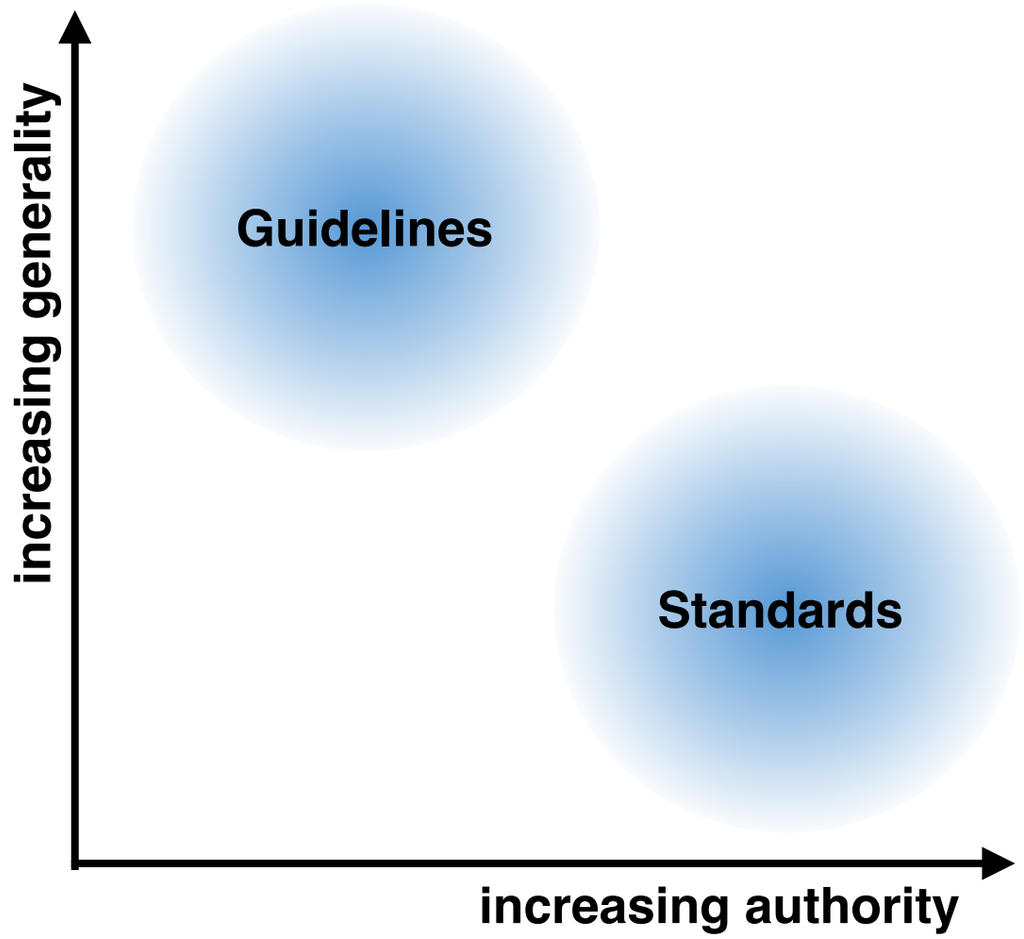
Overview

- Introduction
- Principles to Support Usability
- Standards
- Guidelines
- Golden Rules and Heuristics
- HCI Patterns
- Summary

Introduction

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



Overview

- Introduction
- **Principles to Support Usability**
- Standards
- Guidelines
- Golden Rules and Heuristics
- HCI Patterns
- Summary

Principles to Support Usability

3 Main Categories

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

Overview

- Introduction
- Principles to Support Usability
  - **Learnability**
  - Flexibility
  - Robustness
- Standards
- Guidelines
- Golden Rules and Heuristics
- HCI Patterns
- Summary

Principles of Learnability

- Predictability
- Synthesizability
- Familiarity
- Generalizability
- Consistency

Principles of Learnability

Predictability

- user's knowledge of the interaction history is sufficient to determine the results of his future actions
- different degrees
  - based on currently observable information
  - based on entire interaction history
    (including all key presses and previously displayed screens)

Principles of Learnability

Predictability

- predictability vs. deterministic behavior
  - computer are deterministic machines
    (operation performed at any given point in time results in only one possible follow-up state)
  - but: predictability is a user-centered concept
  - predictability = deterministic behavior from a user's perspective
- **operation visibility** is related concept
  - availability of operations that can be performed next are visible to the user

Principles of Learnability

Synthesizability

- necessary for building up mental models
- synthesis = ability of the user to assess the effect of past operations on the current state
- when an operation changes some aspect of the internal state, it is important that the change is seen by the user
- **principle of honesty** = ability of an UI to provide an observable and informative account of internal changes
  - notification can be *immediately* or *eventually*

Principles of Learnability

Example: Principle of Honesty

- task: movement of file from one directory to another
- principle of honesty implies that the file location is visible to the user after it has been moved
  - command line interface (*eventual honesty*)
    - necessary to remember destination directory
    - verification requires that directory is shown
  - visual desktop interface (*immediate honesty*)
    - file is immediately visible after it has been dragged to the new folder

Principles of Learnability

Familiarity

- new users have wealth of experience across application domains, which is obtained through
  - interactions in the real world
  - interactions with other computer systems
- familiarity = correlation between the user's existing knowledge and the knowledge required for effective interaction
- related concept: **Affordances**
  - e.g., soft button on GUI "suggests" that it should be pressed

Principles of Learnability

Generalizability

- users often try to extend their knowledge of specific interaction behaviors to similar but previously unencountered situations
- generalization can occur within or across applications
  - example for single application: graphical drawing package
    - *circle can be drawn as a constraint form of an ellipse*
    - *generalization: square can be drawn as a constraint form of an rectangle*
  - example of multiple applications: copy&paste command
    - *same operation in all applications*

Principles of Learnability

Consistency

- consistency = likeliness in behavior arising from similar situations or similar tasks
- probably the most widely mentioned principle
- but: not a single property of an interactive system
- instead: consistency must be applied relative to something
- because of relative nature it can be dangerous to follow
  - e.g., QUERTY keyboard
  - “consistent“ keyboard layout hampered innovation

Summary

Example for Consistency: Airplane Warning Systems

- warnings are classified into 3 categories, depending whether they require
  - immediate recovery action
  - eventual but no immediate action
  - no action (advisory)
- consistent color code
  - red => immediate action
  - yellow => eventual action
  - green => no action

Principles of Learnability

Summary of Principles

Principle	Definition
Predictability	determining effect of future actions based on past interaction history
Synthesizability	assessing the effect of past actions
Familiarity	how prior knowledge applies to new system
Generalizability	extending specific interaction knowledge to new situations
Consistency	likeness in input/output behavior arising from similar situations or task objectives

Overview

- Introduction
- Principles to Support Usability
  - Learnability
  - **Flexibility**
  - Robustness
- Standards
- Guidelines
- Golden Rules and Heuristics
- HCI Patterns
- Summary

Principles of Flexibility

- Dialogue Initiative
- Multithreading
- Task Migratability
- Substitutivity
- Customizability

Principles of Flexibility

Dialogue Initiative

- dialog = interaction between user and system
- both partners can initiate the dialog
- **system pre-emptive** dialog: system initiates dialog and user responds to requests for information
- **user pre-emptive** dialog: user is free to initiate any action towards the system
- general goal: maximize the user's ability to pre-empt the system

Principles of Flexibility

Multithreading

- dialog thread = coherent subset of a dialog
- multi-threading = interaction with more than one task at a time
- windowing systems naturally support multi-threaded dialogs
  - each window can represent a different task

Principles of Flexibility

Task Migratability

- transfer of control between system and user
- should be possible for either side to pass over control
- example: spell-checker
  - computer can check words against internal dictionary
  - but: not desirable to leave task completely to computer
    - e.g., handling of proper words, intentional duplication of words, etc.
  - spell-checking best performed in cooperative way
- example: autopilot
  - control of aircraft's position in flight envelope greatly automated
  - but: in case of emergency control is handed over to the pilot

Principles of Flexibility

Substitutivity

- equivalent values can be substituted by each other
- example: scaling of document in printer menue
  - either by providing absolut size or zoom factor
- user can choose form which best suits his needs
- substitutivity can minimize user error and cognitive load by avoiding unnecessary calculations
- related concepts: *representation multiplicity* and *equal opportunity*

Principles of Flexibility

Representation Multiplicity (Substitutivity)

- flexibility for state rendering
- refers to information output
- example: visualization of object temperature
  - presentation as digital thermometer with actual numerical values
  - presentation as graph to visualize temperature changes over time
- each representation provides a perspective to the internal state of the system
- user is free to choose the representation which is most suitable for the current task

Principles of Flexibility

inIT

Equal Opportunity (Substitutivity)

- blurs the distinction between input and output at the interface
- user can decide what is input and what is output
- example: drawing a line
  - possible to draw line by direct manipulation and software calculates length
  - alternative: specification of line coordinates and system draws line
  - both approaches are equally important and must be made equally available

Principles of Flexibility

Customizability

- modifiability of the UI by the user or the system
- user-initiated modification (*adaptability*)
  - user can adjust the form of input and output
  - e.g., adjustment of UI elements, definition of macros
- system-initiated modification (*adaptivity*)
  - automatic customization of the UI by the system
  - e.g., system recognizes experience of user (novice, expert) and adapts dialog to match the needs of the user

Principles of Flexibility

Summary of Principles

Principle	Definition
Dialogue Initiative	freedom from system imposed constraints on input dialogue
Multithreading	ability of system to support user interaction for more than one task at a time
Task Migratability	passing responsibility for task execution between user and system
Substitutivity	allowing equivalent values of input and output to be substituted for each other
Customizability	modifiability of the user interface by user (adaptability) or system (adaptivity)

Overview

- Introduction
- Principles to Support Usability
  - Learnability
  - Flexibility
  - **Robustness**
- Standards
- Guidelines
- Golden Rules and Heuristics
- HCI Patterns
- Summary

Principles of Robustness

- Observability
- Recoverability
- Responsiveness
- Task Conformance

Principles of Robustness

Observability

- ability of user to evaluate the internal state of the system from its perceivable representation
- can be divided into 5 sub-principles
  - operation visibility
  - browsability
  - defaults
  - reachability
  - persistance

Principles of Robustness

Operation Visibility (Observability)

- visibility of operations that can be performed
  - if operation can be performed, there should be visible indication for the user
  - if operation can not be performed, it should be indicated as well (e.g., greying of button)
- general goal: recognition over recall
  - avoid that the user has to remember what he can do and what he can’t do

Principles of Robustness

Browsability (Observability)

- possibility to explore the current internal state of the system
- usually complexity of the domain does not allow to show all relevant information at the same time
- example: word processor
  - only part of the document is shown
  - but: outline view can give overview of entire document
- important: browsing should not have any side effects on the internal state

Principles of Robustness

Defaults (Observability)

- assist the user by passive recall
- suggested response to a question can be recognized as corrected instead of recalled
- example: printer dialog

Principles of Robustness

Reachability (Observability)

- possibility of navigating through the observable system states
- important that users can navigate from any given state to any other state

Principles of Robustness

Persistance (Observability)

- duration of the effect of a communication act and the ability of the user to make use of that effect
- effect of vocal communication not persistent (except in the mind of the receiver)
- effect of visual communication usually persistent
- example: e-mail notification
  - beep vs. pop-up window

Principles of Robustness

Recoverability

- ability of a user to take corrective actions once an error has been recognized
- *forward error recovery*: acceptance of the current state and negotiation from that state towards the desired state
  - not always possible (e.g., card house)
- *backward error recovery*: undo of the effects of the previous interaction in order to return to a prior state
  - e.g., undo after an unintentional deletion of a paragraph in a word processor

Principles of Robustness

Responsiveness

- how the user perceives the rate of communication with the system
- *response time* = duration needed by the system to express state changes
  - generally, short durations and instantaneous response times are desirable
  - if instantaneous response is not possible: system should indicate that request has been received and is being processed

Principles of Robustness

Responsiveness

- *response time stability* = invariance of the duration for identical or similar operations
- example: pull-down menus
  - users expect menu to pop up instantaneously
  - variations in response time might cause interaction problems

Principles of Robustness

Task Conformance

- degree to which the system supports all of the user's tasks
  - *task completeness*: coverage of tasks
  - *task adequacy*: addresses user's understanding of the task

Principles of Robustness

Summary of Principles

Principle	Definition
Observability	ability of user to evaluate the internal state of the system from its perceivable representation
Recoverability	ability of user to take corrective action once an error has been recognized
Responsiveness	how the user perceives the rate of communication with the system
Task Conformance	degree to which system services support all of the user's tasks

Overview

- Introduction
- Principles to Support Usability
  - Learnability
  - Flexibility
  - Robustness
- **Standards**
- Guidelines
- Golden Rules and Heuristics
- HCI Patterns
- Summary

Standards

- set by national or international bodies
- goal: ensure compliance by a large community
- standards require sound underlying theory and slowly changing technology
- hardware standards more common than software standards
- high authority and low level of detail

Example: UK Interim Defense Standard 00-25 on *Human Factors for Designers of Equipment*

Part 1: *Introduction*
Part 2: *Body Size*
Part 3: *Body Strength and Stamina*
Part 4: *Workplace Design*
Part 5: *Stresses and Hazards*
Part 6: *Vision and Lightning*
Part 7: *Visual Displays*
Part 8: *Auditory Information*
Part 9: *Voice Communication*
Part 10: *Controls*
Part 11: *Design for Maintainability*
Part 12: *Systems*

Example: UK Defense Standard for *Pilot Cockpit Controls and Instrumentation*

11.3 Arrangement of Displays

11.3.1 Vertical Grouping. The engine display parameters shall be arranged so that the primary or most important display for a particular engine and airplane (thrust, torque, RPM, etc.) be located at the top of the display group if a vertical grouping is provided. The next most important display parameter shall be positioned under the primary display progressing down the panel with the least important at the bottom.

Example: DIN 66 234 Part 3

5.1 Subdivision of Display Area

In consideration of a simple, fast and accurate visual acquisition, the display area shall be divided into different sub-areas. Such a division should be:

- input area
- output area
- area for operational indications (such as status and alarms).

Example: US Military Standard MIL-STD-1472C

> **5.15.3.2.1 Standardization**
>
> The content of displays within a system shall be presented in a consistent manner.

Standards

Example: ISO 9241

- defines **usability** as effectiveness, efficiency and satisfaction with which users accomplish tasks
  - *effectiveness* = accuracy and completeness with which specified users can achieve specified goals in a particular environment
  - *efficiency* = resources expended in relation to the accuracy and completeness of goals achieved
  - *satisfaction* = comfort and acceptability of the work system to its users and other people affected by its use
- provides explicit means for measuring usability

Overview

- Introduction
- Principles to Support Usability
  - Learnability
  - Flexibility
  - Robustness
- Standards
- **Guidelines**
- Golden Rules and Heuristics
- HCI Patterns
- Summary

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

Example: Guidelines by Smith and Mosier (1986)

1. Data Entry

1.1 Position Designation

1.1-1 Distinctive Cursor

For position designation on an electronic display, provide a moveable cursor with distinctive visual features (shape, blink, etc.).

Exception. When position designation involves only selection among displayed alternatives, highlighting selected items might be used instead of a separately displayed cursor.

Comment. When choosing a cursor shape, consider the general content of the display. For instance, an underscore cursor would be difficult to see on a display of underscored text, or on a graphical display containing many other lines.

Comment. If the cursor is changed to denote different functions (e.g., to signal deletion rather than entry), then each different cursor should distinguishable from the other.

Comment. If multiple cursors are used on the same display (e.g., one for alphanumeric entry and one for line drawing), then each cursor should be distinguishable from others.

Reference. Whitfield, Ball and Bird, 1983

See also. 1.1-17 Distinctive multiple cursors

4.0-9 Distinctive cursor

Overview

- Introduction
- Principles to Support Usability
  - Learnability
  - Flexibility
  - Robustness
- Standards
- Guidelines
- **Golden Rules and Heuristics**
- HCI Patterns
- Summary

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.,
  - Shneiderman’s 8 Golden Rules
  - Norman’s 7 Principles

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.

Overview

- Introduction
- Principles to Support Usability
  - Learnability
  - Flexibility
  - Robustness
- Standards
- Guidelines
- Golden Rules and Heuristics
- **HCI Patterns**
- Summary

HCI 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, where you can re-orientate (HCI)

HCI Patterns

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, organizational, conceptual, detailed
- are intuitive and readable and can therefore be used for communication between all stakeholders

Overview

- Introduction
- Principles to Support Usability
  - Learnability
  - Flexibility
  - Robustness
- Standards
- Guidelines
- Golden Rules and Heuristics
- HCI Patterns
- **Summary**

Summary

- Designing for **maximum usability** is the goal of interactive system design.
- **Abstract principles** offer a way of understanding usability in a more general sense, especially if we can express them within some coherent catalog.
- **Design rules** in the form of **standards** and **guidelines** provide direction for design, in general and more concrete terms, in order to enhance the interactive properties of the system.

Summary

- The essential characteristics of good design are often summarized through ‘**golden rules**‘ or **heuristics**.
- **Design patterns** provide a potentially generative approach to capturing and reusing design knowledge.