

Interacção Humana com o Computador

Aula 09



Departamento de Informática
UBI 2024/2025

João Cordeiro
jpcc@ubi.pt



HUMAN-COMPUTER INTERACTION

THIRD
EDITION



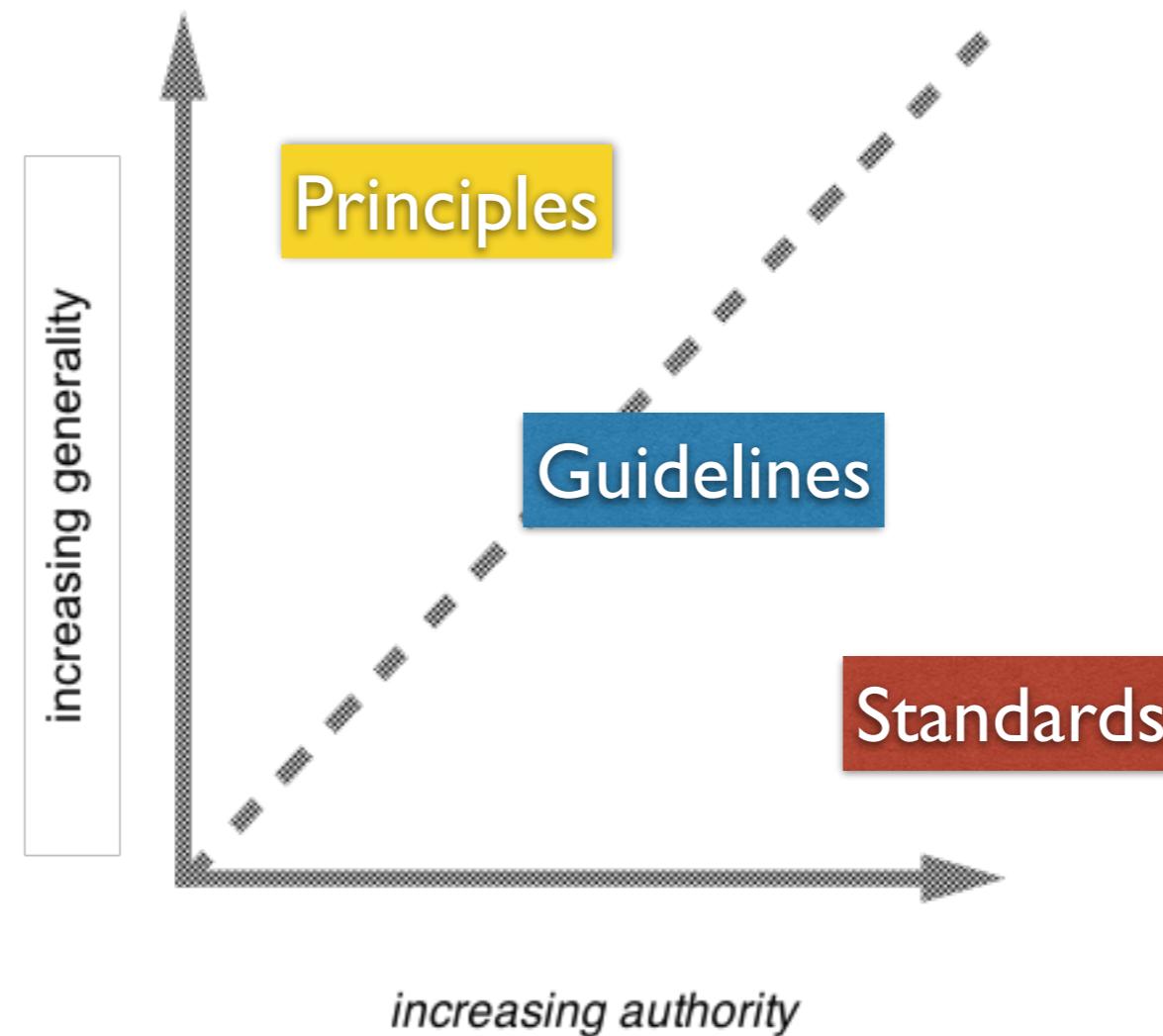
DIX
FINLAY
ABOWD
BEALE

chapter 7

Design Rules



Using design rules



Design rules

- Suggest how to increase usability
- Different kinds, containing different levels of **generality** and **authority**.



Types of design rules

● Principles

- **abstract** design rules
- **high** generality
- **low** authority

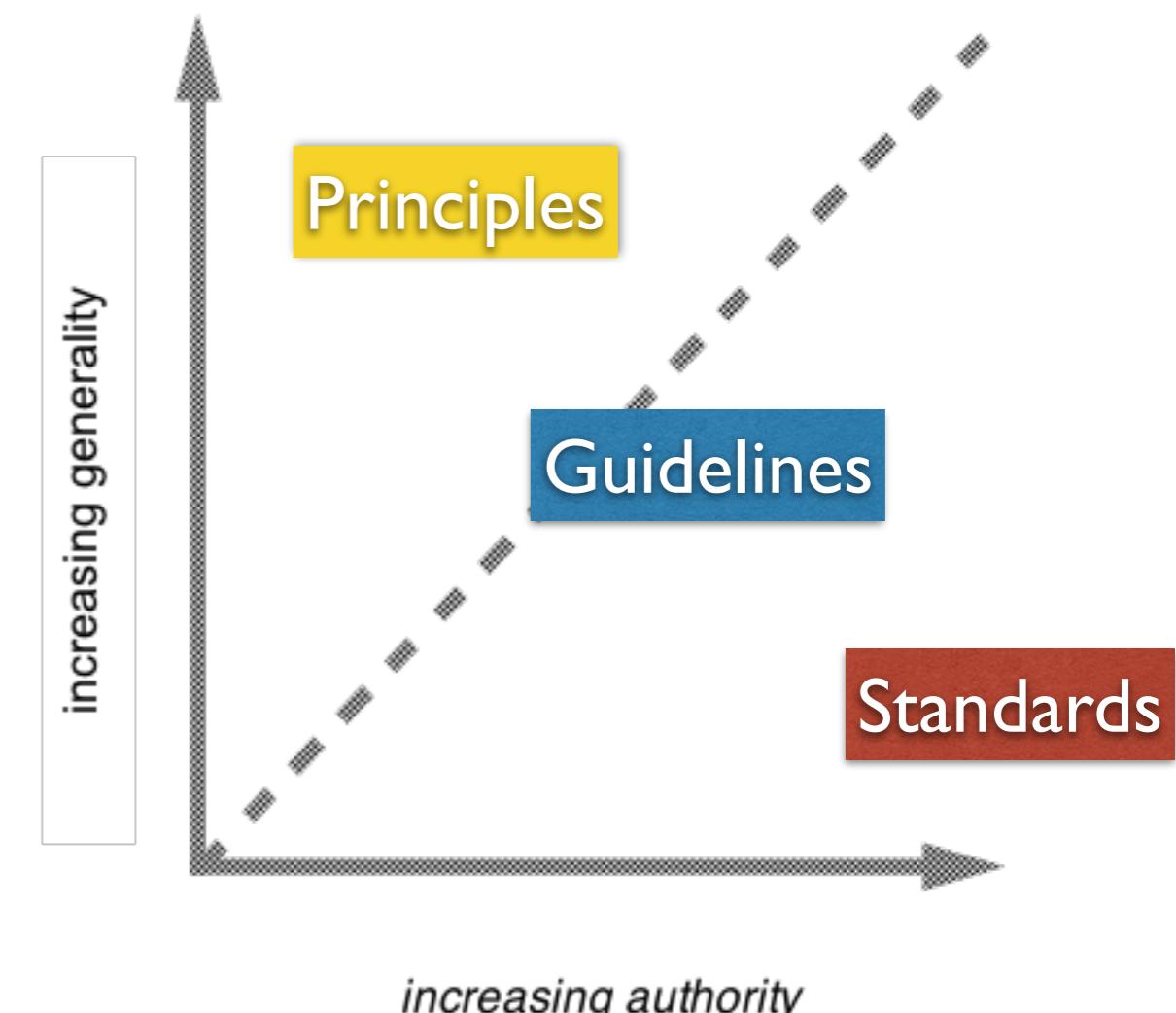
- Psychology
- Sociology
- Comp. Science

● Guidelines

- claim **more** authority
- **less** general / more tech.

● Standards

- **specific** design rules
- **high** authority
- **limited** application



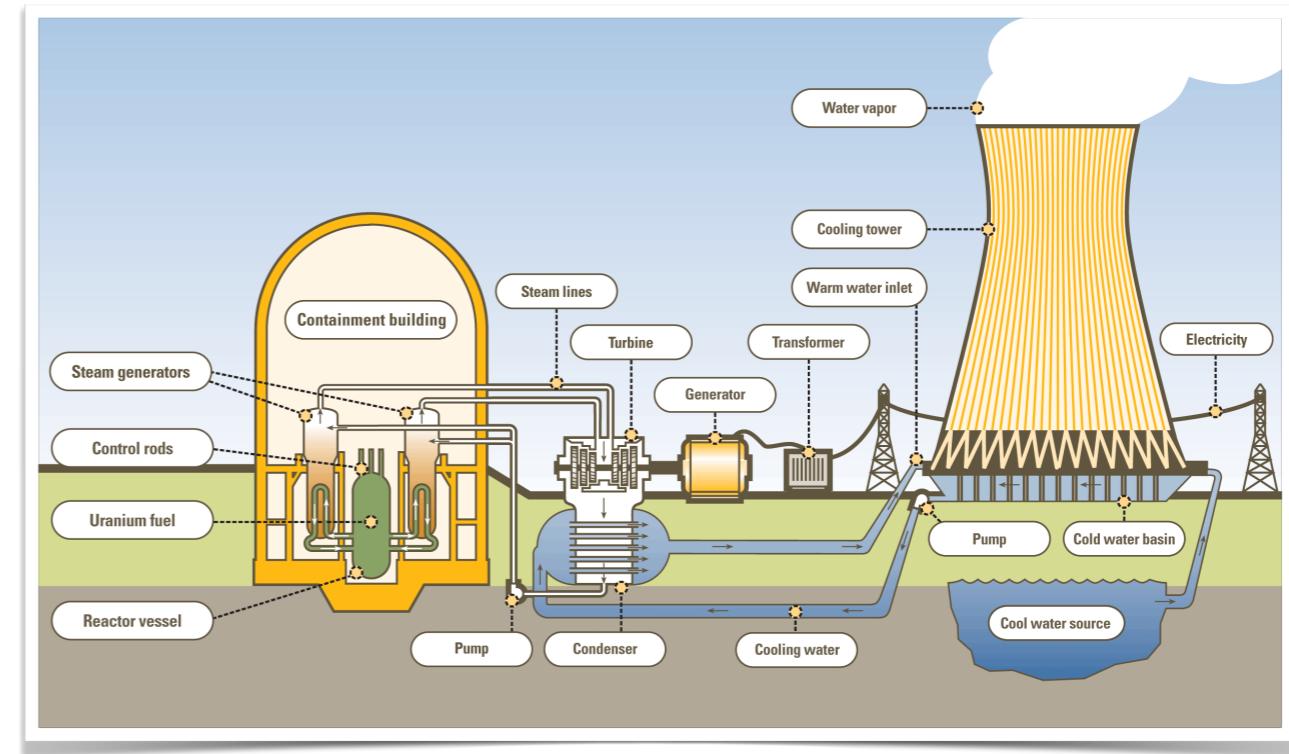
Standards



Standards (ISO ...)

- Set by **national or international bodies** to ensure compliance by a large community of designers. Standards require sound underlying theory and slowly changing technology.

- Longer history on **safety-critical** domains



- **Hardware** — More common
Ergonomics and physiology
- **Software** — With authority but low level of detail
Cognitive Sciences



Standards

- **Interim Defense Standard 00-25 on Human Factors for Designers of Equipment. (UK)**

Standard for the design of military equipment

1. Introduction
2. Body Size
3. Body Strength and Stamina
4. Workplace Design
5. Stress and Hazards
6. Vision and Lighting
7. Visual Displays
8. Auditory Information
9. Voice Communication
10. Controls
11. Design Maintainability
12. Systems (software)

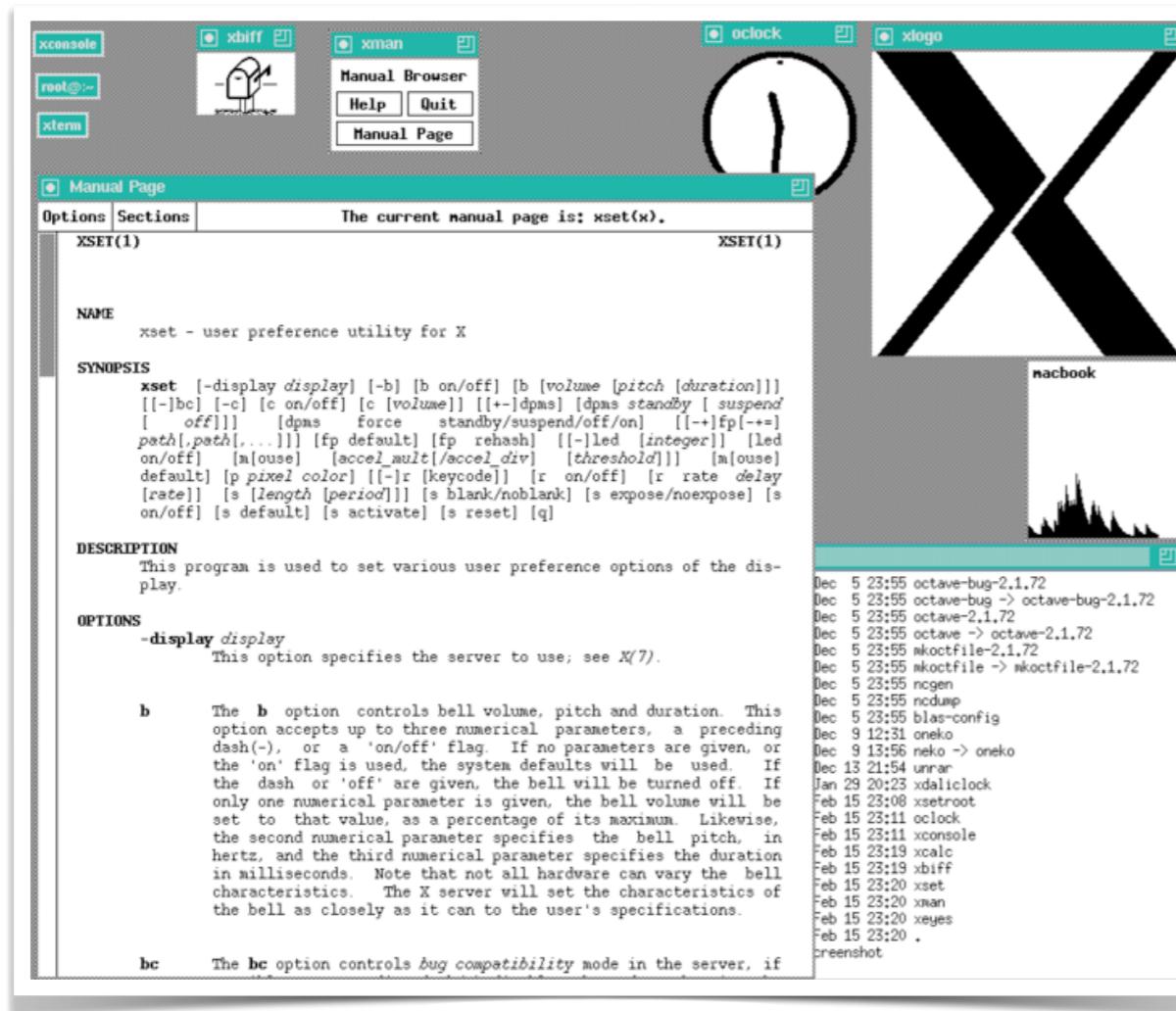


BSI: British Standard Institution.



Standards in Software

- Most of them are more suggestive than mandatory! Usually become standards way before any formal standardization was set.



X-Windows



Massachusetts
Institute of
Technology

1984

*The strength of a norm lays
in its fellow community.*



Standards

• Examples:

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.

(a) A typical example of a military standard

5.1 Subdivision of the 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)

generality

(b) From German standard DIN 66 234 Part 3 (1984), adapted from Smith [324]

5.15.3.2.1 Standardization

The content of displays within a system shall be presented in a consistent manner.

(c) From US military standard MIL-STD-1472C, revised (1983), adapted from Smith [324]

Figure 7.1 Sample design standards for displays. Adapted from Smith [324].

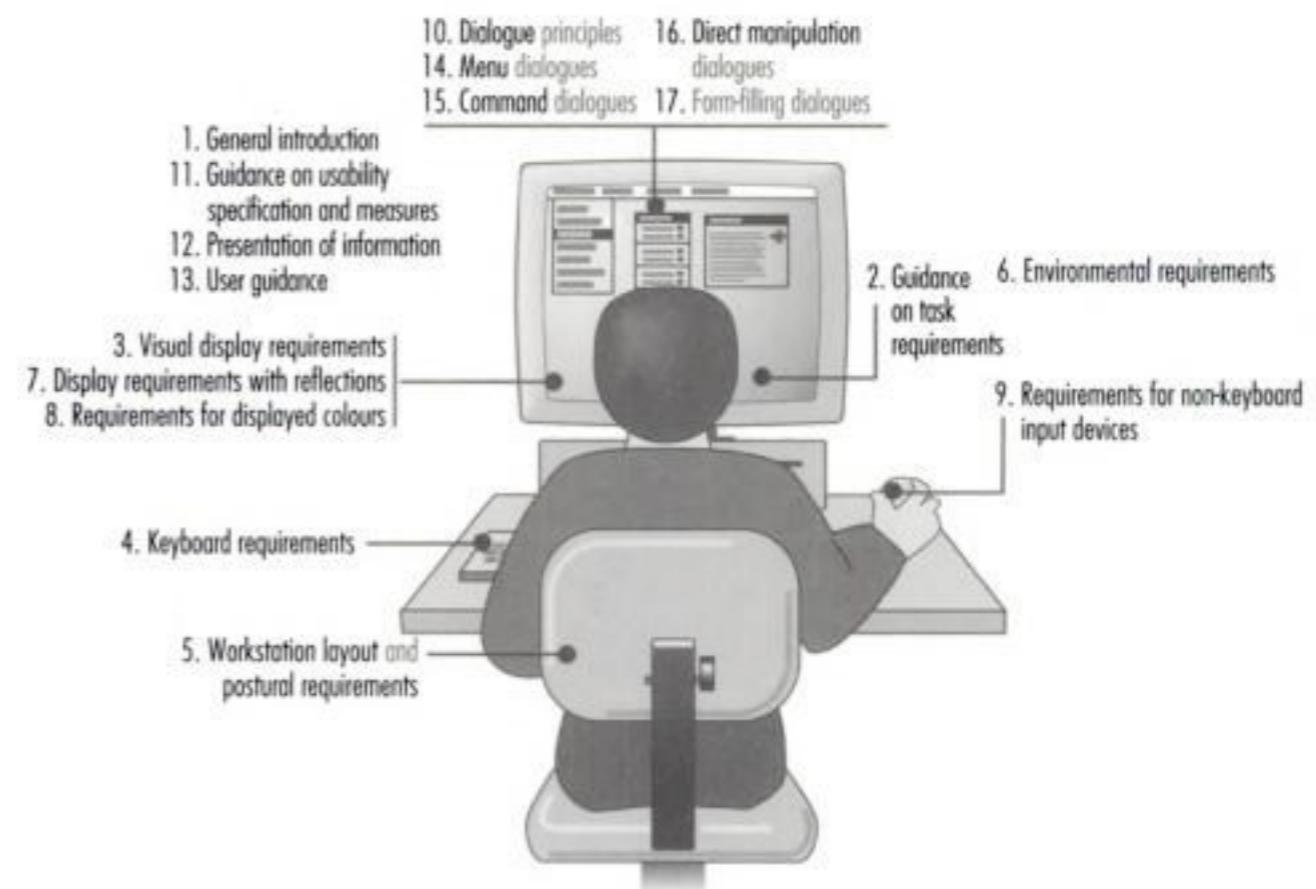
Copyright © 1986 IEEE



Standards | ISO 9241

- **ISO 9241** defines **usability** as **effectiveness**, **efficiency** and **satisfaction** with which users accomplish tasks
- **Effectiveness**
 - precision
 - completeness
- **Efficiency**
 - resources spent
- **Satisfaction**
 - comfort
 - acceptability

Figure 52.11 • Technical Working Groups of the Ergonomics of Human System Interaction Tech (ISO TC 159 SC4). **ISO 9241**: Five working groups broke down the "parts" of below. This illustration shows the correspondence between the parts of the standard and the parts of the workstation with which they are concerned.





Standards | ISO 9241

ISO 9241 is a multi-part standard concerning the ergonomics of human-system interaction. It is aimed at guiding how to design systems that are effective, efficient, and satisfying to use. It covers aspects of product and system design, including:

- **Ergonomics Requirements** for office work with visual display terminals, etc.
- **Guidelines on usability**: It provides specifications and guidelines for designing products that meet ergonomic requirements for office work with visual displays.
- **Physical Ergonomics of work environments**: This addresses ergonomic principles related to physical activity.
- **Visual Ergonomics** in the design of the user interface.
- **Menu Interfaces**, command interfaces, web design, and software accessibility.



Standards | ISO 9241

ISO 9241 is a multi-part standard concerning the ergonomics of human-system interaction. It is aimed at guiding how to design systems that are effective, efficient, and satisfying to use.

- This standard addresses conventional interface design aspects and extends to accessibility, user experience, and organizational ergonomics.
- ISO 9241 helps in creating work environments that fit the user's physical requirements and cognitive abilities, enhancing performance and user satisfaction.



Standards | ISO 9241

- **ISO 9241** defines **usability** as **effectiveness**, **efficiency** and **satisfaction** with which users accomplish tasks

Part 1: General Introduction - Provides an overview of and introduction to the entire standard series.

Part 2: Guidance on task requirements - Offers guidance on the attributes of tasks that influence the usability of systems.

Part 3: Visual display requirements - Details requirements for visual display terminals.

Part 4: Keyboard requirements - Specifies the ergonomic design of keyboards.

Part 5: Workstation layout and postural requirements - Focuses on ergonomic requirements for office tasks with visual displays.

Part 6: Environmental requirements - Addresses the physical environment and ergonomics of the workstation.

Part 7: Display requirements with reflections - Deals with visual requirements including reflections.

Part 8: Requirements for displayed colours - Provides requirements for the use of color in displays.

Part 9: Requirements for non-keyboard input devices - Specifies requirements for devices other than keyboards like mice and trackballs.

Part 10: Dialogue principles - Outlines principles that should be applied to the design of dialogues between humans and systems.

Part 11: Usability: Definitions and concepts - Offers definitions of usability and describes how to measure it.

Part 12: Presentation of information - Provides guidelines for presenting and representing information on visual displays.

Part 13: User guidance - Offers guidelines for the design of user guidance like help systems and manuals.

Part 14: Menu dialogues - Specifies design requirements for menu interfaces.

Part 15: Command dialogues - Focuses on the design of command languages.

Part 16: Direct manipulation dialogues - Discusses the design of interfaces that allow direct manipulation of objects on the screen.

Part 17: Form-filling dialogues - Provides guidelines for the design of form-based dialogue systems.

Part 20: Accessibility guidelines for ICT equipment and services - Helps designers in making equipment and services accessible.

.....



Standards | ISO 9241

- **ISO 9241** defines **usability** as **effectiveness**, **efficiency** and **satisfaction** with which users accomplish tasks

Part 14: Menu dialogues - Specifies design requirements for menu interfaces.

Design Guidelines

- **Structure:** Menus should be structured to reflect the importance and frequency of use of the items. Most frequently used items should be placed where they are easily accessible.
- **Depth vs. Breadth:** The design should balance between having too many items at one level (breadth) and having many nested levels (depth). A shallow, broad menu is generally faster to use than a deep, narrow one.
- **Navigation:** Menus should allow for easy navigation, including the ability to go back to previous menus and to exit the menu system altogether.
- **Adaptability:** Menu systems should be adaptable to users' changing needs, allowing for personalization and customization where possible.

Guidelines



Guidelines (Orientações)

- They range from the most **general** to the most **specific**.
- Lots of manuals and reports, full of guidelines, example: [Smith & Moiser 1986]

- Data Entry
- Data Display
- Sequence Control
- User Guidance
- Data Transmission
- Data Protection

1.4 DATA ENTRY: Data Forms 1.4/5 Data Field Labels

For each data field, display an associated label to help users understand what entries can be made.

Example:

(Good)	NAME: _____
	ORGANIZATION: ____/____
	PHONE: ____-_____
(Bad)	NAME, ORGANIZATION AND PHONE _____ _____ _____

Reference:

BB 2.1.7

See also: 1.0/24 4.0/11



Guidelines (Orientações)

- GUIDELINES FOR DESIGNING USER INTERFACE SOFTWARE: [Smith & Moiser 1986]

1.1 Position Designation

Position designation refers to user selection and entry of a position on a display, or of a displayed item.

1.1/1 Distinctive Cursor

For position designation on an electronic display, provide a movable 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 be distinguishable from the others.

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 the others.

Reference

- [Whitfield Ball Bird 1983](#)

See also

on-line:

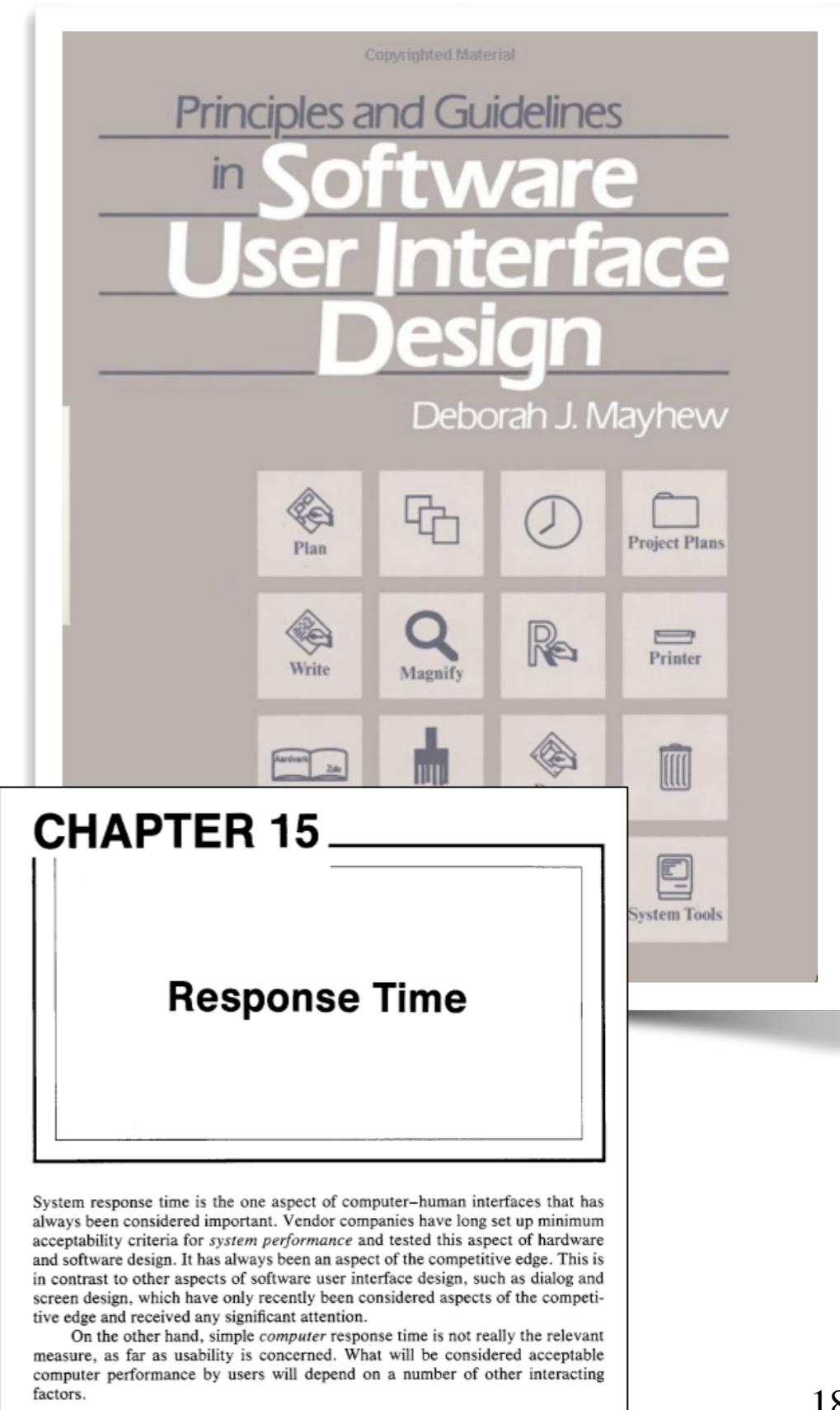
<http://hcibib.org/sam/>



Guidelines

- Others also **suggestive** and **general**: [Mayhew 1992]

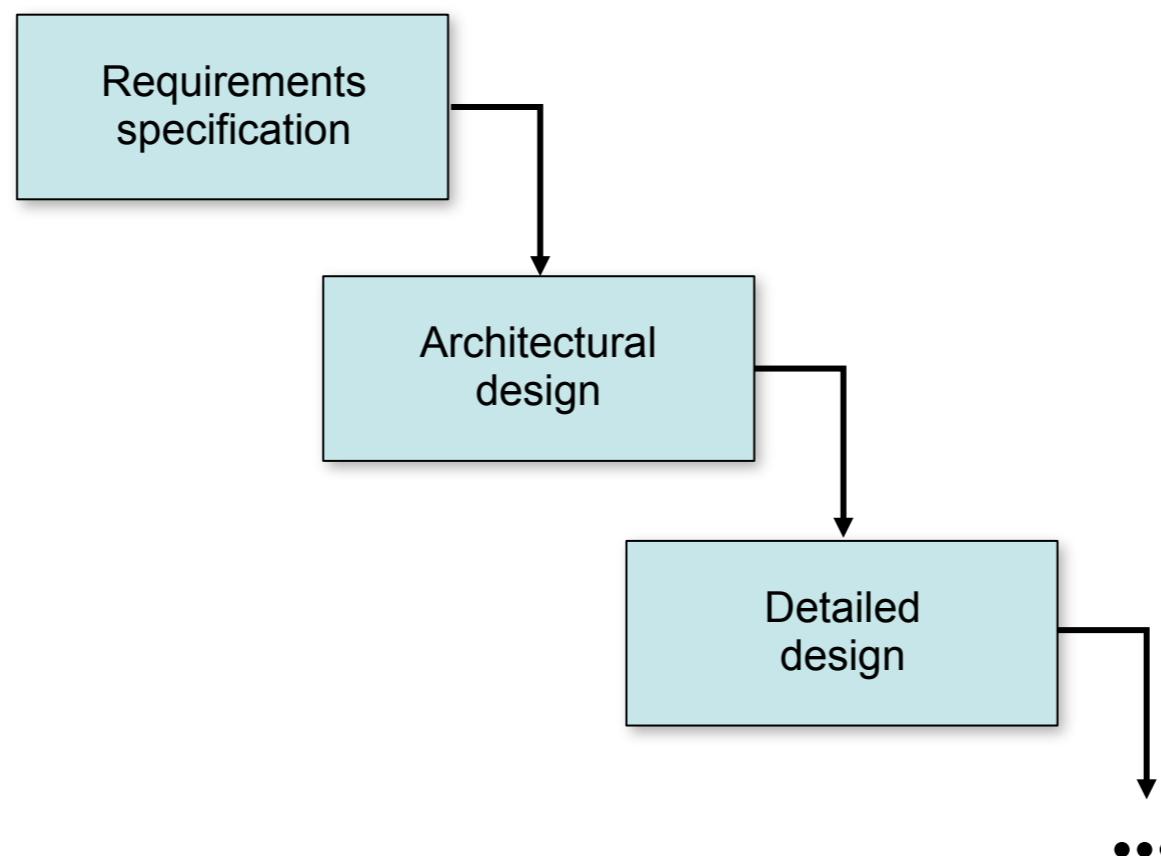
References	/8
Chapter 3 Conceptual Models	80
What Is a Mental Model?	81
Mental Models: Experimental Results	82
Why Do People Form Mental Models?	93
Designing Conceptual Models	95
Examples of Conceptual Models	97
Guidelines for Designing Conceptual Models	104
References	111
Chapter 4 Dialog Styles: Menus	113
What Is a Menu Interface?	114
When Is a Menu Dialog Style Appropriate?	117
Designing Menu Systems: Experimental Results	120
Designing Menu Systems: Principles and Guidelines	144
References	170
Chapter 5 Dialog Styles: Fill-in Forms	172
What Is a Fill-in Form Interface?	172
When Is a Fill-in Form Dialog Style	





Guidelines

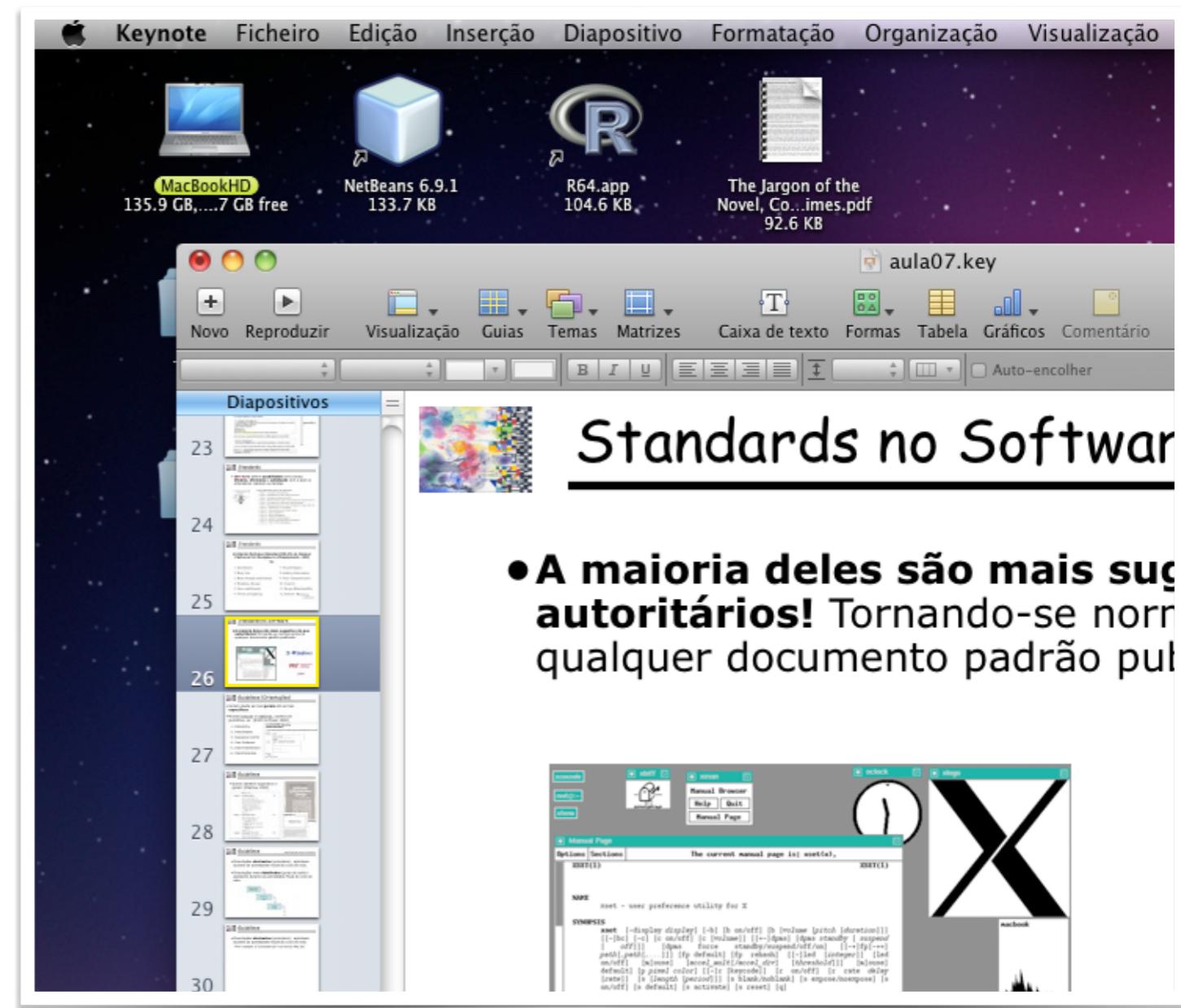
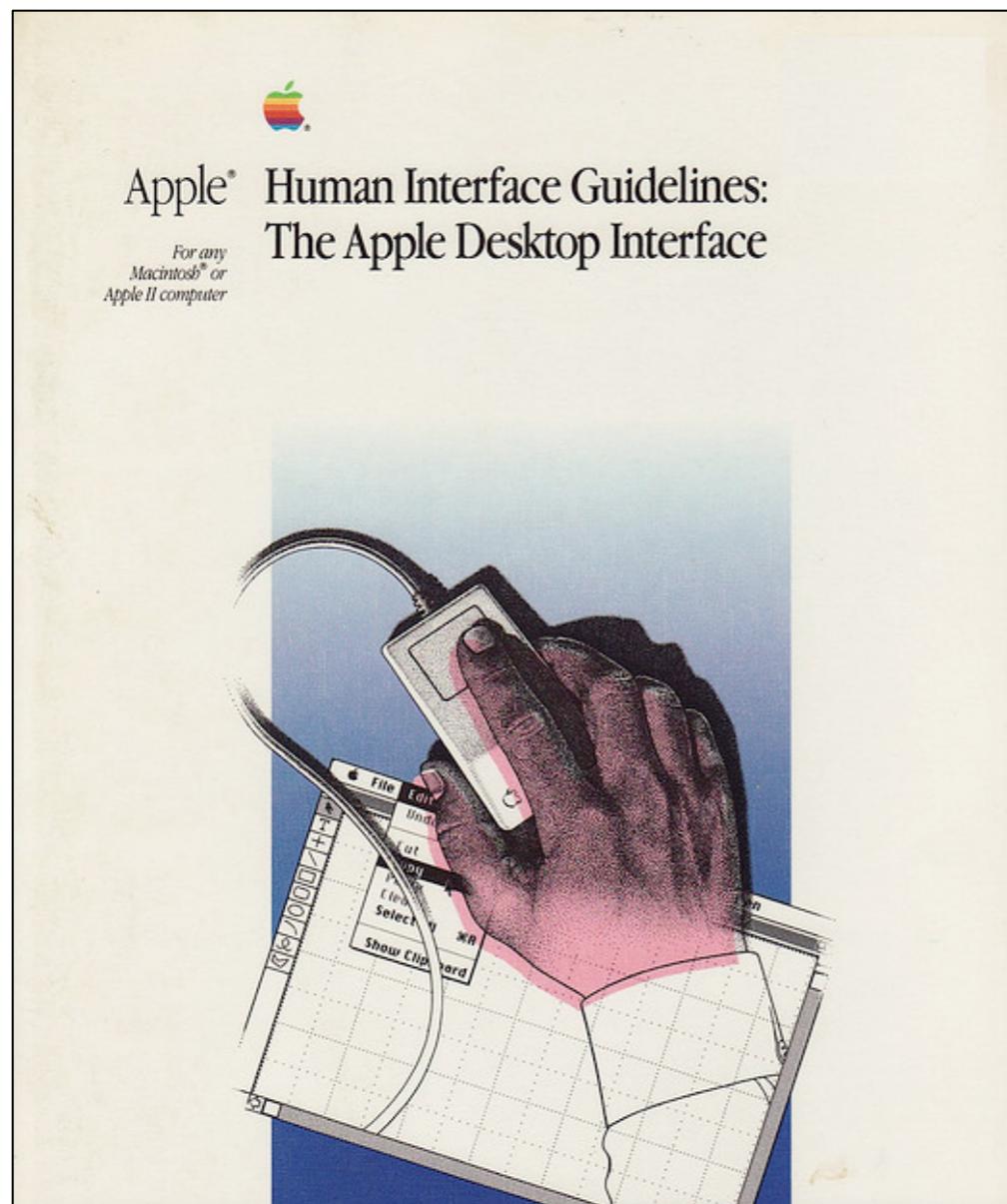
- The more **abstract** kind of guidelines (principles) are more suited to the initial activities of the life cycle.
- There are also more **detailed** guidelines — more adequate to be observed during the final life-cycle activities.





Guidelines

- The more **abstract** kind of guidelines (principles) – applicable during the initial activities of the life cycle.
 - For example, **consistency** in the iOS menus



- **A maioria deles são mais suç autoritários!** Tornando-se nor qualquer documento padrão pul



Guidelines

- The more **abstract** kind of guidelines (principles) – applicable during the initial activities of the life cycle.
 - For example, **consistency** in the iOS menus

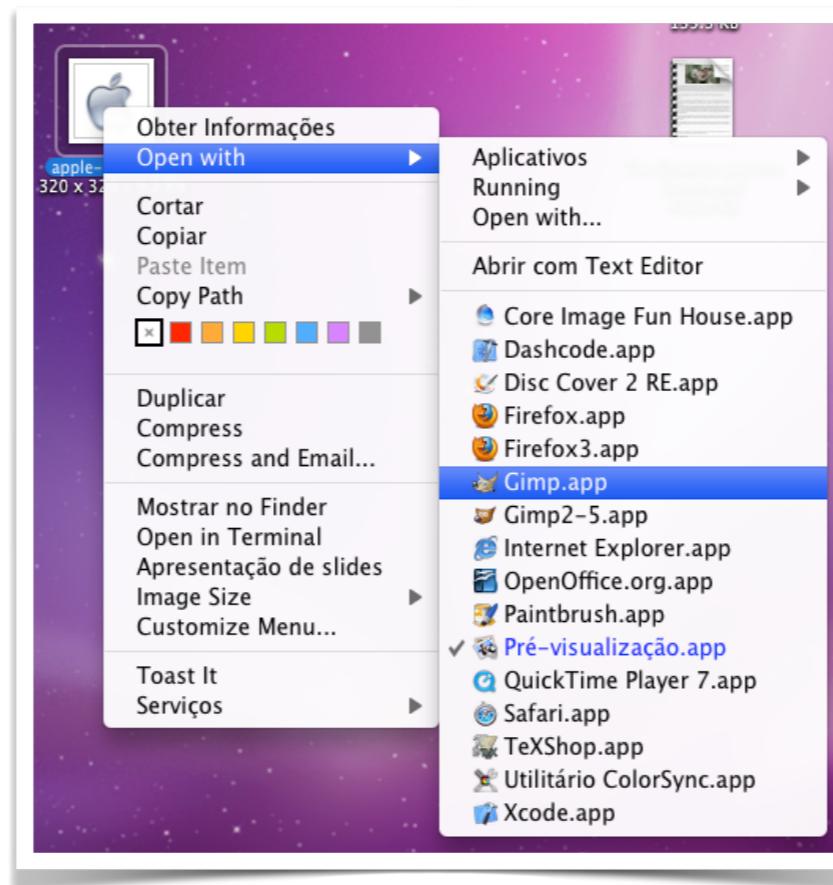
Exemplo de falha
de consistência,
nos menus do
Gimp.





Guidelines

- More detailed **guidelines** (style guides) - applicable during the final life cycle activities, an example is the “Noun-Verb” directive from Apple.



- The dialog directive of “**user-preemptiveness**”: *“The user, not the computer, initiates and controls all actions”.*



Guidelines -- iOS

Apple Developer Discover Design Develop Distribute Support Account

Human Interface Guidelines Resources

Apple UI Design Resources

Design apps quickly by using Photoshop and Sketch templates, plug-ins, and other preconfigured UI elements.

iOS Apps

Apple UI Design Resources include Photoshop, Sketch, and Adobe XD templates, along with comprehensive UI resources that depict the full range of controls, views, and glyphs available to developers using the iOS SDK. These resources help you design apps that match the iOS design language. Icon and glyph production files are preconfigured to automate asset production using Sketch slices or Adobe Generator for Photoshop CC. Color swatches, dynamic type tables, and fonts are also included.

For design guidance, see [Human Interface Guidelines > iOS](#).

The screenshot shows the 'Apple UI Design Resources' page. At the top, there's a navigation bar with links for 'Discover', 'Design', 'Develop', 'Distribute', 'Support', and 'Account'. Below that is a section titled 'Human Interface Guidelines' with a 'Resources' link. The main content area has a large heading 'Apple UI Design Resources' and a sub-section 'iOS Apps' with a detailed description. To the right, there's a preview of the iOS UI components section, showing a screenshot of an iPhone X's control center with volume and brightness sliders, and a toggle switch labeled 'Title'.

<https://developer.apple.com/design/resources/>



Guidelines -- iOS

Apple Developer Discover Design Develop Distribute Support Account

Human Interface Guidelines iOS ▾

Overview ▾

Themes

- iPhone X
- What's New in iOS 11
- Interface Essentials

App Architecture

User Interaction

System Capabilities

Visual Design

Icons and Images

Bars

Views

Controls

Extensions

Technologies

Resources



iOS Design Themes

As an app designer, you have the opportunity to deliver an extraordinary product that rises to the top of the App Store charts. To do so, you'll need to meet high expectations for quality and functionality.

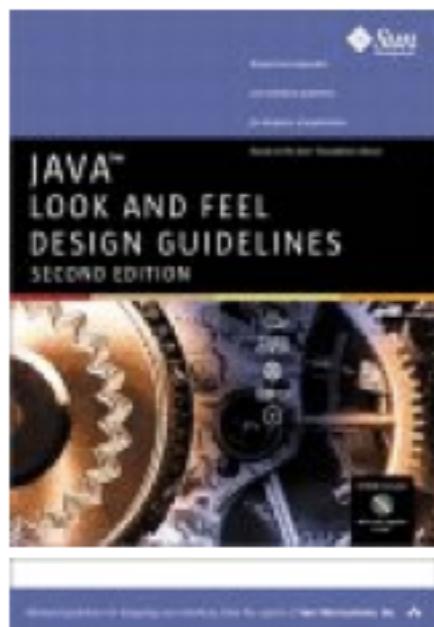


Guidelines - Java GUI Example

- For Java GUI there are several guidelines

Java Look and Feel Design Guidelines

Available online!



- VOLUME I. [Java™ Look and Feel Design Guidelines, second edition](#)



- VOLUME II. [Java™ Look and Feel Design Guidelines: Advanced Topics](#)



Guidelines - Java GUI Example

- Chapter 4: Visual Design

Themes

As a software developer, you can use the [theme mechanism](#) to control many of the fundamental attributes of the Java look and feel design, including colors and fonts. For instance, you might want to change the colors and fonts in your application to match your corporate identity. The theme mechanism enables you to specify alternative colors and fonts across an entire Java look and feel application. [\[Link\]](#).

Figure 22 Primary Colors in Default Color Theme

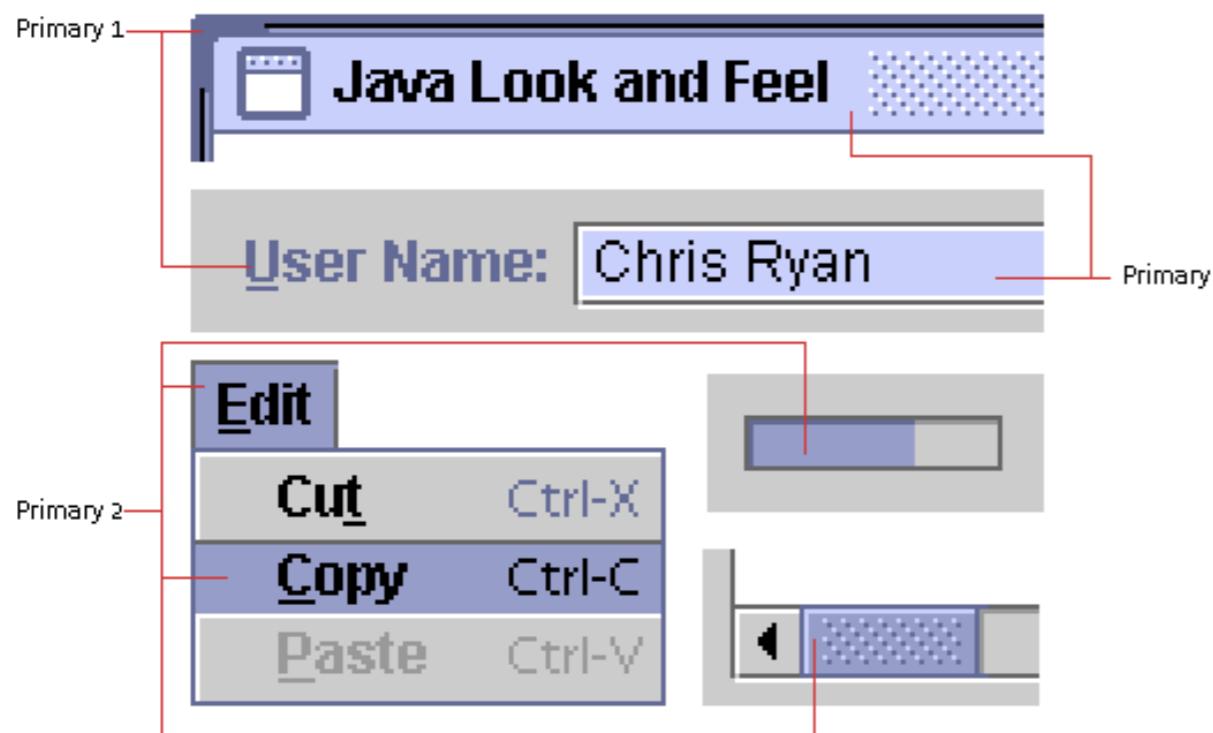
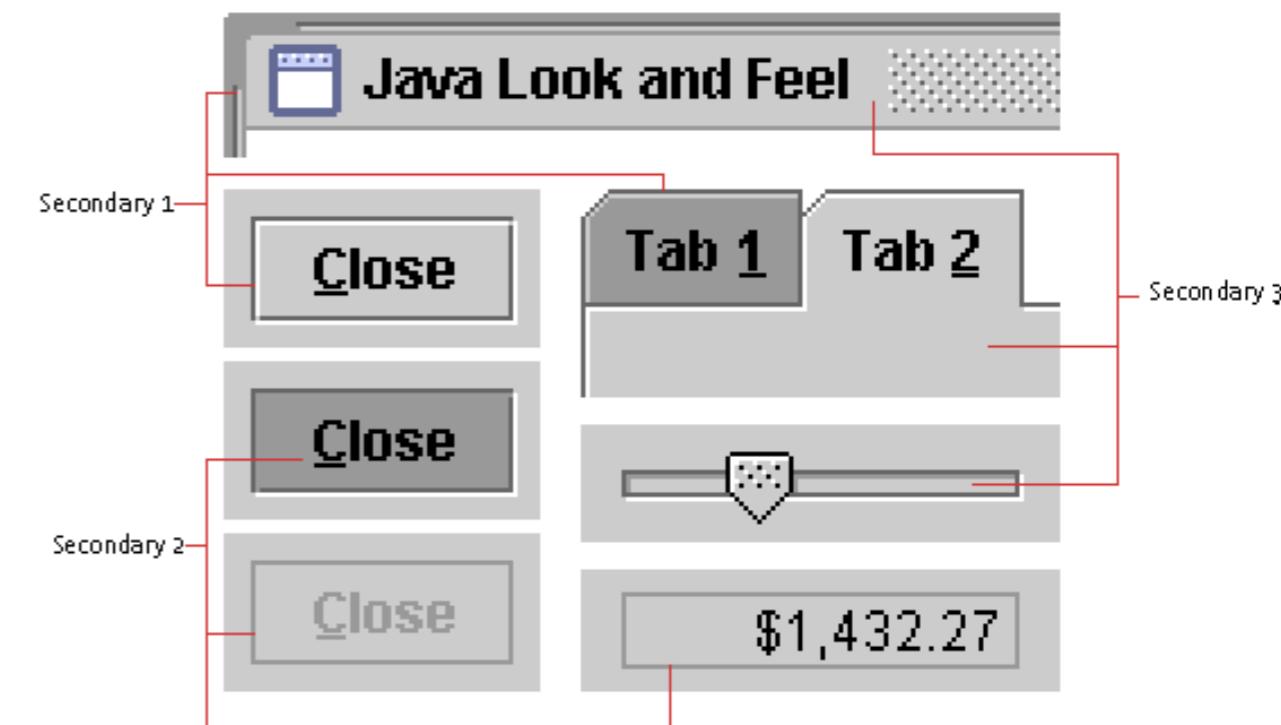


Figure 23 Secondary Colors in Default Color Theme



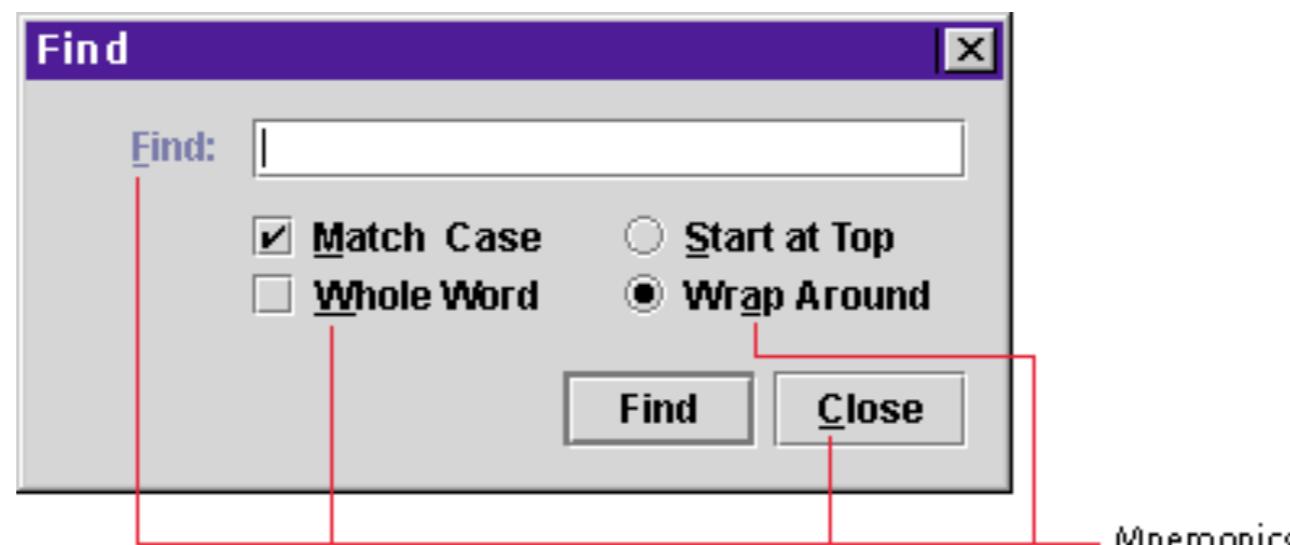


Guidelines - Java GUI Example

- For Java GUI there are several guidelines

Mnemonics and Keyboard Shortcuts

You should provide mnemonics and keyboard shortcuts throughout your application. A mnemonic is an underlined alphanumeric character that shows users which key to press (in conjunction with the Alt key) to activate a command or navigate to a component ...





Guidelines - Java GUI Example

- Chapter 6: Responsiveness

Determining Acceptable Response Delays

The term **response delay** refers to how long an application takes to acknowledge or fulfill a particular user request. Providing responsiveness in an application depends on achieving response delays that are acceptable to users [[Link](#)].

Table 13 Maximum Acceptable Response Delays for Typical Events

User Interface Events	Maximum Acceptable Response Delay
Mouse click; pointer movement; window movement or resizing; key press; button press; drawing gesture; other user-input event involving hand-eye coordination	0.1 second (100 milliseconds)
Displaying progress indicators; completing ordinary user commands (for example, closing a dialog box); completing background tasks (for example, reformatting a table)	1.0 second
Displaying a graph or anything else that a typical user would expect to take time (for example, displaying a new list of all a company's financial transactions for an accounting period)	10.0 seconds
Accepting and processing all user input to any task	10.0 seconds



Guidelines - Java GUI Example

- The various "Look and Feel" themes are designed for aesthetic coherence, across multiple platforms

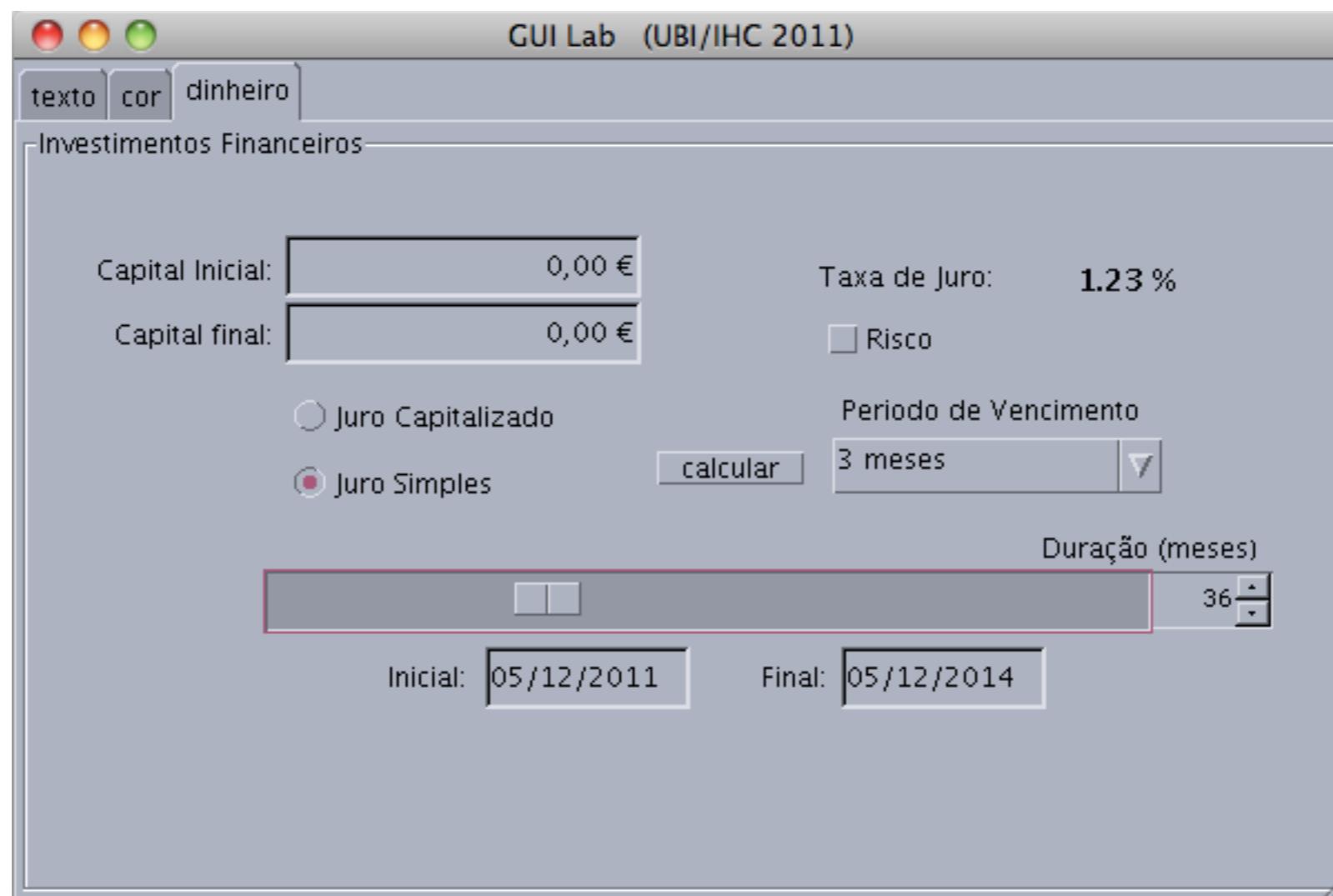
Platform	Look and Feel
Solaris, Linux with GTK+ 2.2 or later	GTK+
Other Solaris, Linux	Motif
IBM UNIX	IBM*
HP UX	HP*
Classic Windows	Windows
Windows XP	Windows XP
Windows Vista	Windows Vista
Macintosh	Macintosh*



Guidelines - Java GUI Example

- For Java GUI there are several guidelines

“Motif”



`com.sun.java.swing.plaf.motif.MotifLookAndFeel`



Guidelines - Java GUI Example

- For Java GUI there are several guidelines

“GTK”



com.sun.java.swing.plaf.gtk.GTKLookAndFeel



Guidelines - Java GUI Example

- For Java GUI there are several guidelines

“Metal”

`javax.swing.plaf.metal.MetalLookAndFeel`



Guidelines - Java GUI Example

- For Java GUI there are several guidelines

Exemple o a specific definition o a “Look & Feel”.

```
public static void main(String args[]) {  
    try {  
        UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }  
  
    java.awt.EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            new MainFrame().setVisible(true);  
        }  
    });  
}
```



Guidelines - Java GUI Example

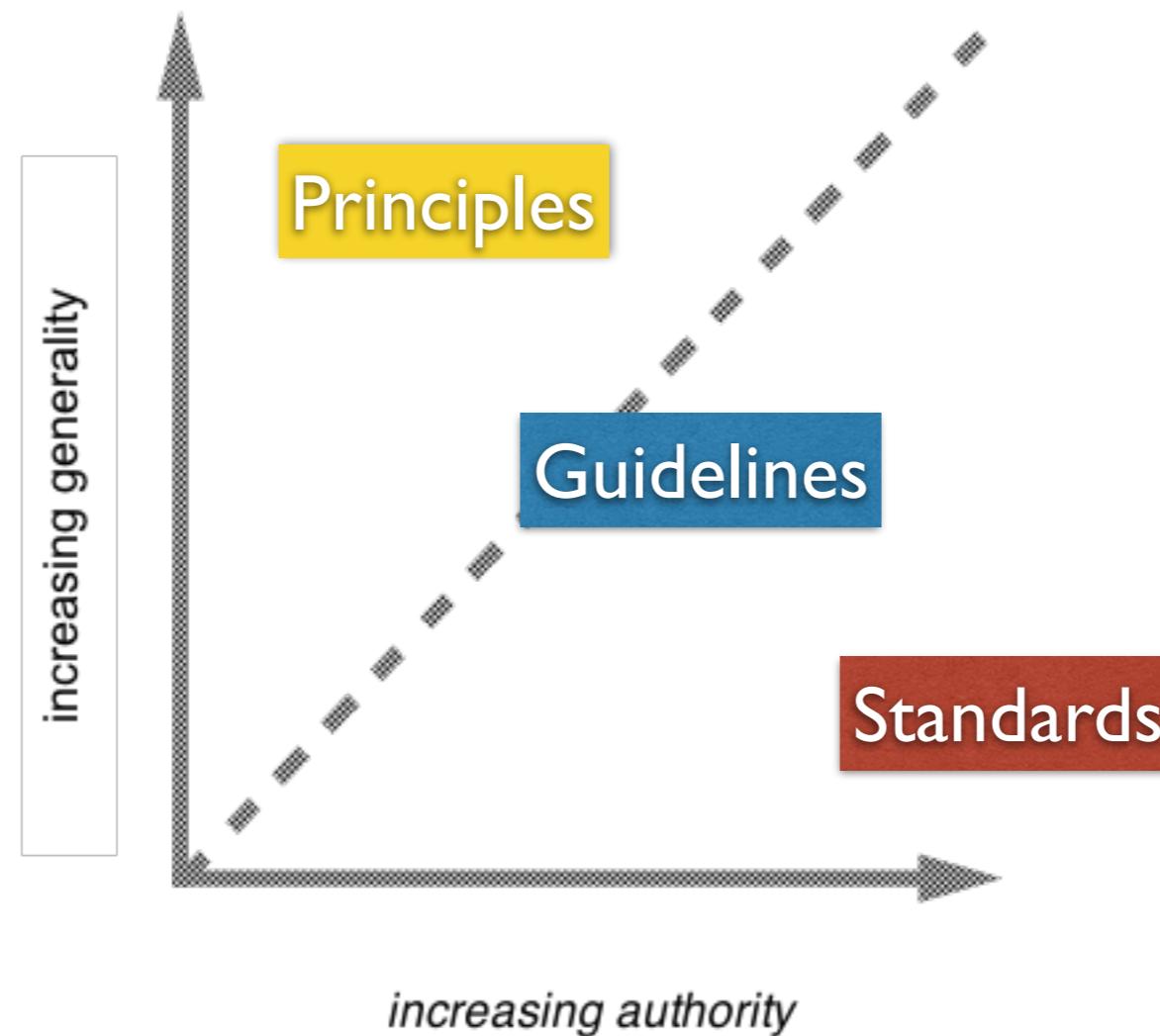
- In JavaFX use CSS skins

<http://javafxtuts.com/javafx-css/>

```
13 public class Javafxtuts extends Application {  
14  
15     @Override  
16     public void start(Stage primaryStage) {  
17         HBox root = new HBox();  
18         //Set space or padding using setPadding() method  
19         root.setPadding(new Insets(20));  
20  
21         //assiging a class to the button  
22         Button button=new Button("my button");  
23         //Adding a class to the button  
24         button.getStyleClass().add("btn");  
25  
26         //assiging a class to the button1  
27         Button button1 =new Button("Button1");  
28         //set id to the button.  
29         button1.setId("btn1");  
30  
31  
32         root.getChildren().addAll(button,button1);  
33         Scene scene = new Scene(root, 300, 150);  
34         //To add a external css file we do as  
35         String style= getClass().getResource("New.css").toExternalForm();  
36         //now add the external css file to the scene  
37         scene.getStylesheets().add(style);  
38  
39         primaryStage.setTitle("javafxtuts.com");  
40         primaryStage.setScene(scene);  
41         primaryStage.show();  
42     }  
43  
44     /**  
45      * @param args the command line arguments  
46      */  
47  
48     public static void main(String[] args) {  
49         launch(args);  
50     }  
51 }
```



Using design rules



Design rules

- Suggest how to increase usability
- Different kinds, containing different levels of **generality** and **authority**.



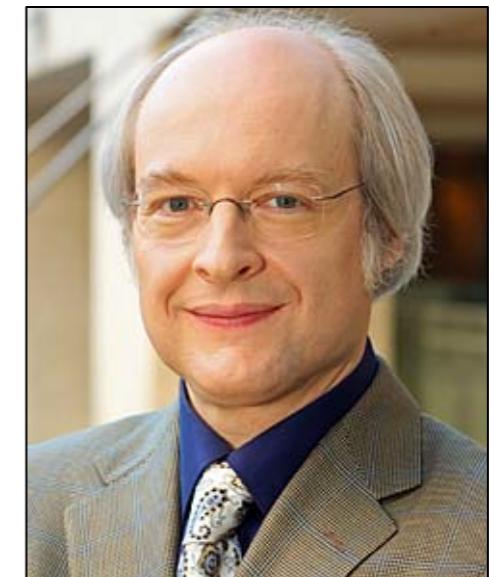
Golden Rules and Heuristics

- They are helpful “checklist” for a good design
- You always get better with these instead of using none
- There are different collections:
 - **Nielsen:** The 10 Heuristics.
 - **Norman:** The 7 Principles
 - **Shneiderman:** The 8 Golden Rules



The Nielsen's 10 Heuristics

- *Visibility of system status*
- *Match between system and the real world*
- *User control and freedom*
- *Consistency and standards*
- *Error prevention*
- *Recognition rather than recall*
- *Flexibility and efficiency of use*
- *Aesthetic and minimalist design*
- *Help users recognize, diagnose, and recover from errors*
- *Help and documentation*



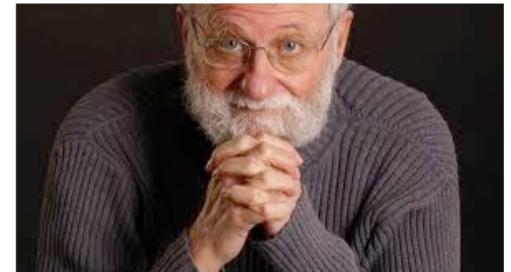
Jakob Nielsen



Norman's 7 Principles

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

- dicas mentais
- automação
- redesenhar.



Donald Norman

2. Simplify the structure of tasks

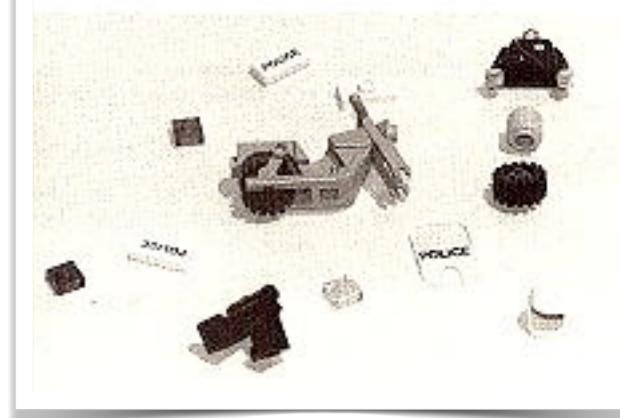
3. *Make things visible: bridge the gulfs of Execution and Evaluation*

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*

When there is no mapping we should define standards.

Note: The example of critical controls on a car.

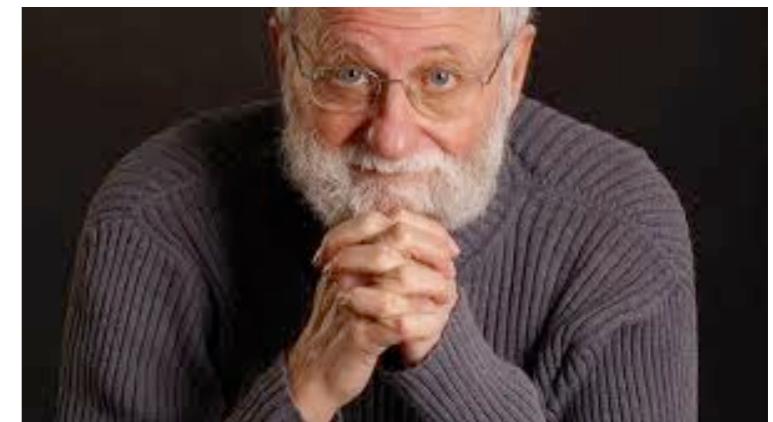




Norman's 7 Principles

4. Get the mappings right

A funny story of bad mapping:



Donald Norman



The *Leitz* projector

Taste (7) für Diawechsel am Gerät

Diawechsel vorwärts = kurz drücken,
Diawechsel rückwärts = länger drücken.

Button (7) for changing the slides

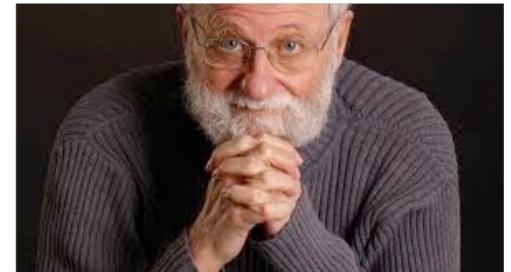
Slide change forward = short press,
Slide change backward = longer press.



Norman's 7 Principles

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

- dicas mentais
- automação
- redesenhar.



Donald Norman

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.*

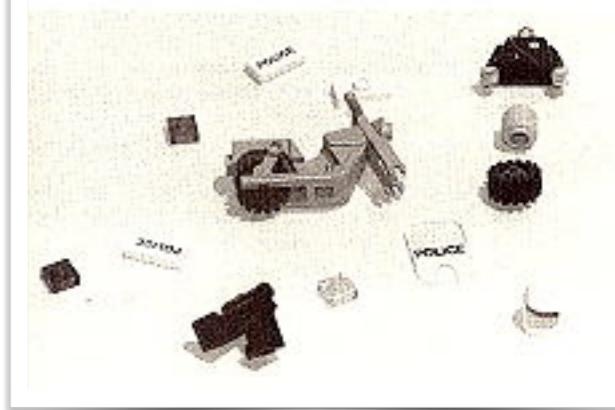
6. Design for error



7. *When all else fails, standardize*

When there is no mapping we should define standards.

Note: The example of critical controls on a car.





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*

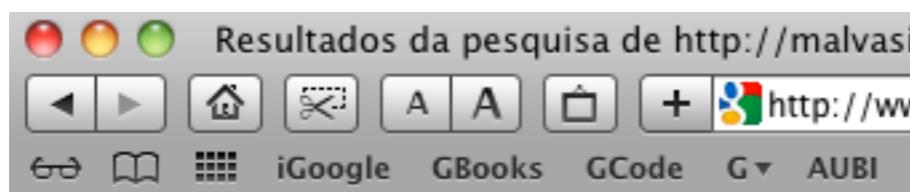


Ben Shneiderman



HCI Design Patterns

- Way to **reuse** knowledge from previous good design experiences
- A pattern is an **invariant** solution for a recurrent problem, within a specific **context**.
- Examples:
 - “Natural light in two walls” (Architecture)
 - “Return to a safe haven” (HCI)



- Patterns do not exist in isolation, but are **linked** to other patterns in languages allowing the generation of complete designs



HCI Design Patterns

● Pattern Attributes:

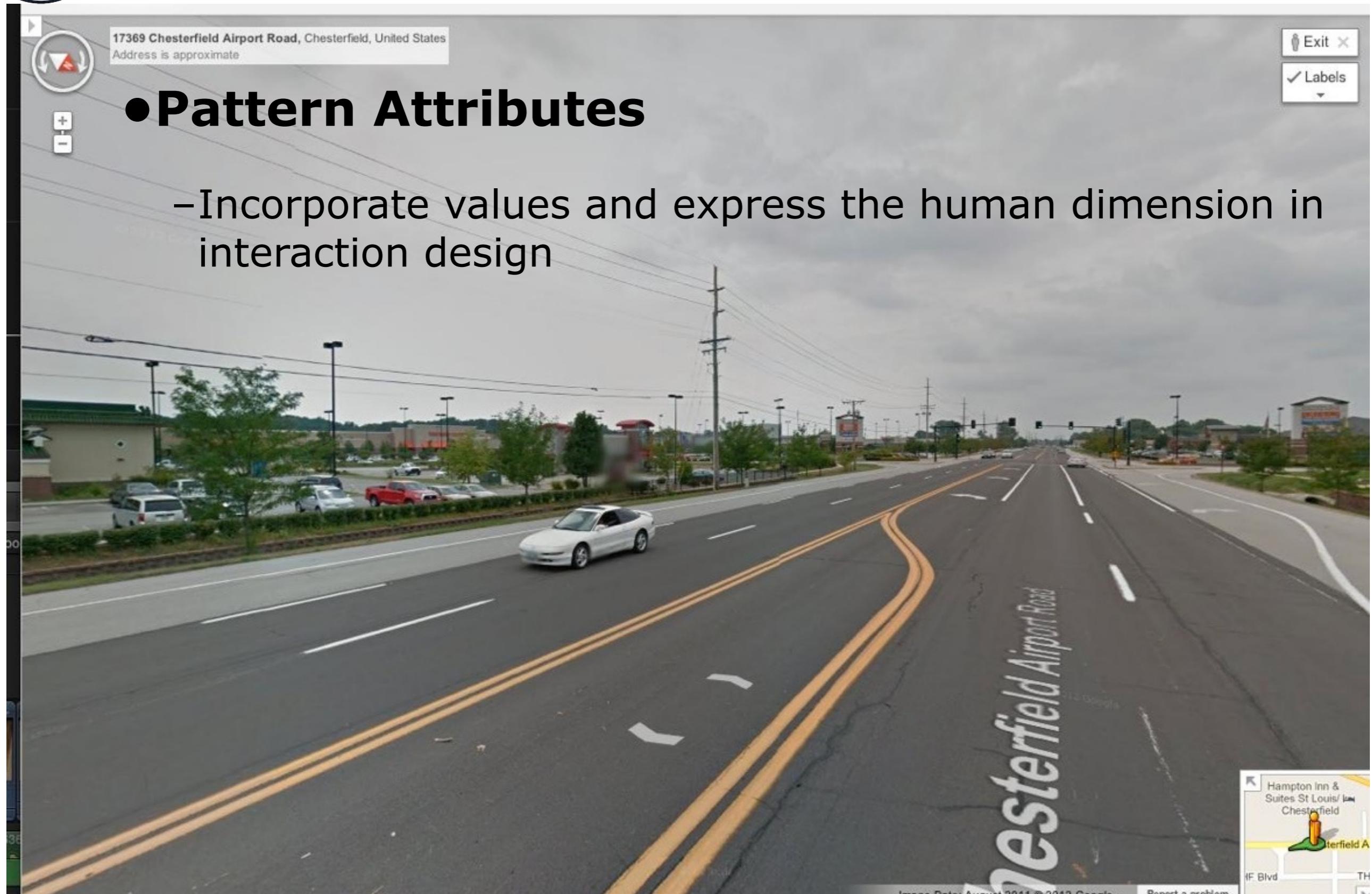
- Capture the **practical**, non-theoretical design;
- Captures the **common essential** properties of good design examples;
- Represent the knowledge of design at **several levels**: social, organizational, conceptual, detailed;
- Are **intuitive** and easy to read and can therefore be used for communication between all stakeholders;
- Incorporate values and **express** the human dimension in interaction design.



HCI Design Patterns

● Pattern Attributes

- Incorporate values and express the human dimension in interaction design





HCI Design Patterns

● Pattern Attributes

- Incorporate values and express the human dimension in interaction design





HCI Design Patterns

● Pattern Attributes

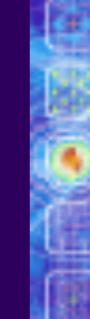
- Incorporate values and express the human dimension in interaction design





HUMAN-COMPUTER INTERACTION

THIRD
EDITION



DIX
FINLAY
ABOWD
BEALE

chapter 12

Cognitive Models

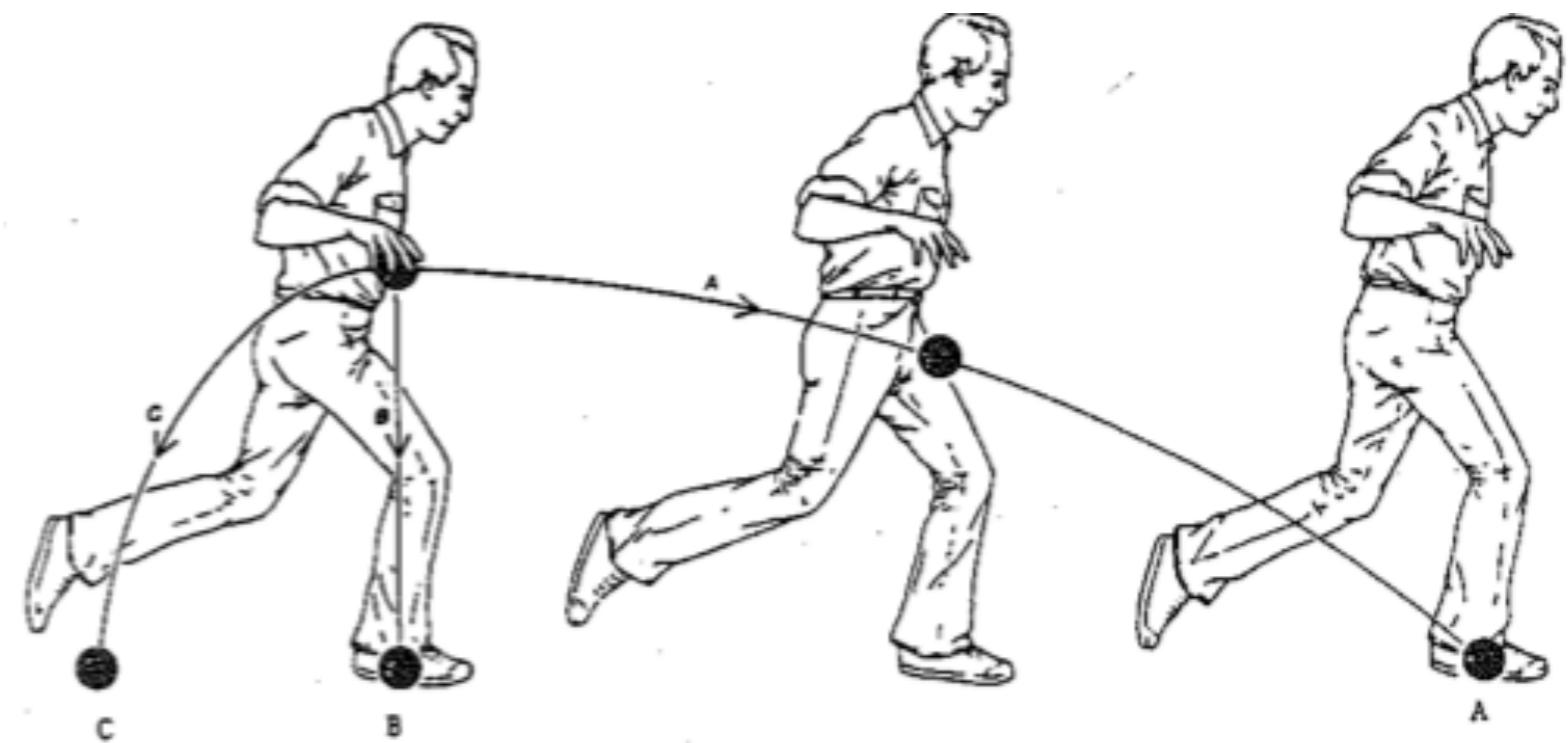
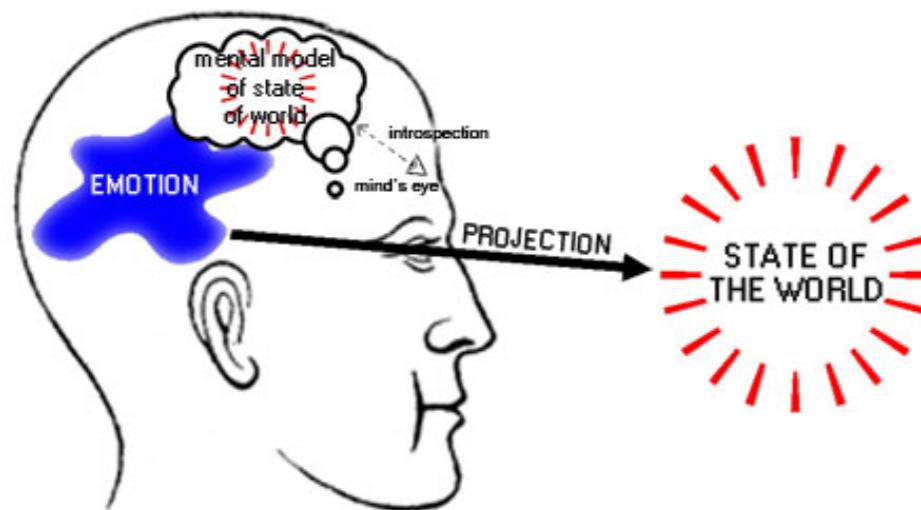


HUMAN-COMPUTER INTERACTION

THIRD
EDITION

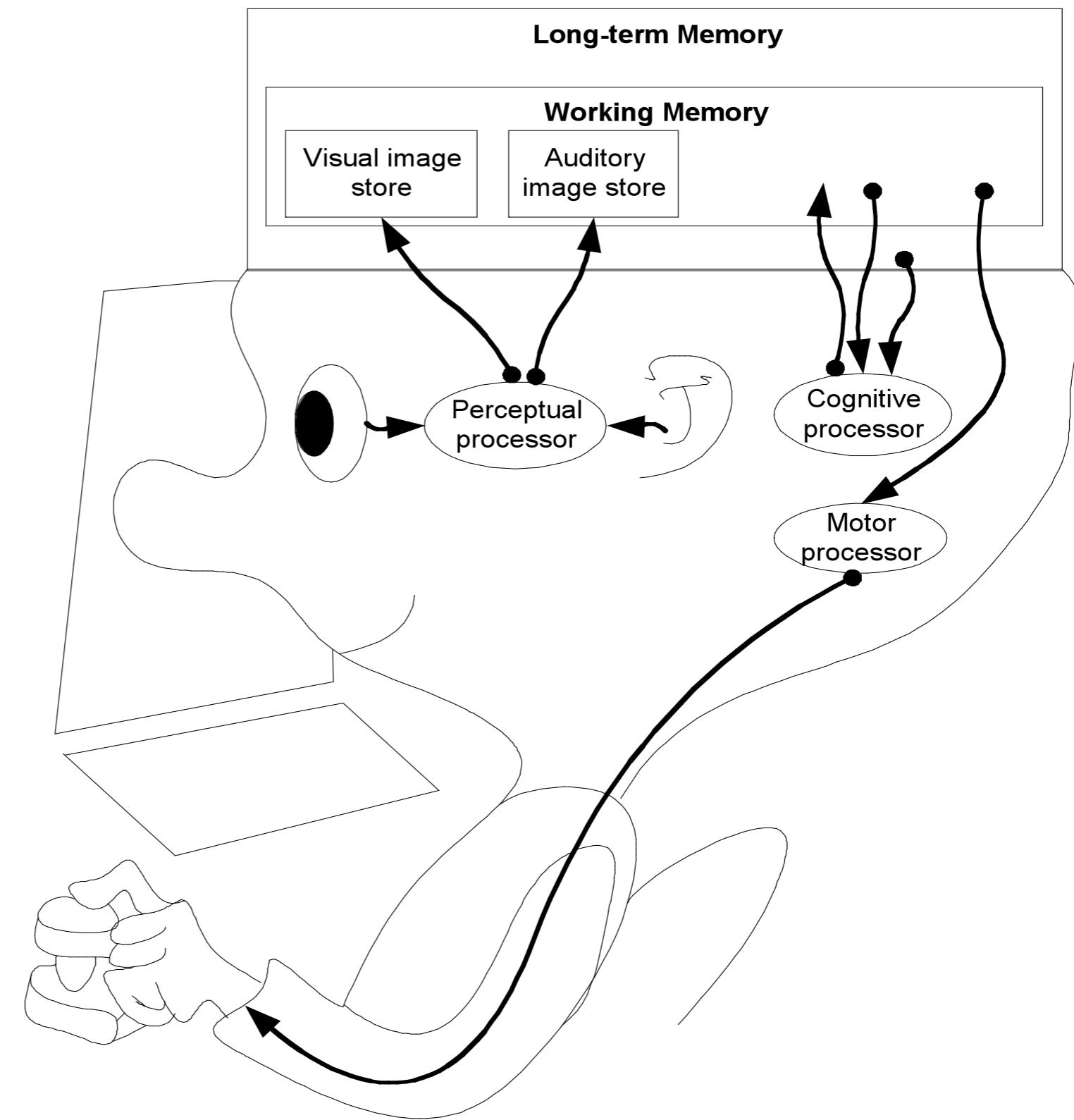
DIX
FINLAY
ABOWD
BEALE

Humans create mental models to explain behavior



Cognitive Models

How do users
perceive,
think
and act



Donald Norman's Model

- Seven stages
 1. User establishes the goal
 2. Formulates intention
 3. Specifies actions at interface
 4. Executes action
 5. Perceives system state
 6. Interprets system state
 7. Evaluates system state with respect to goal
- Norman's model concentrates on user's view of the interface





Donald Norman's Model

Some systems are harder to use than others

Why?

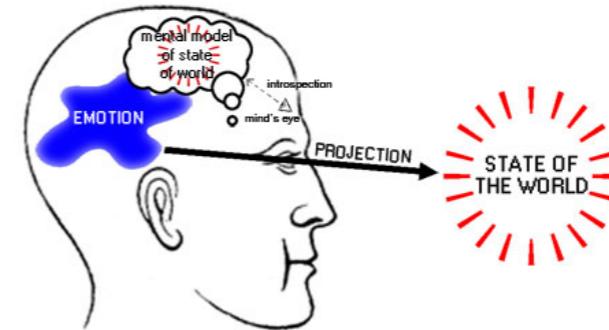
Gulf of Execution

user's formulation of actions \neq actions allowed by the system

Gulf of Evaluation

user's expectation of changed system state \neq actual presentation of this state

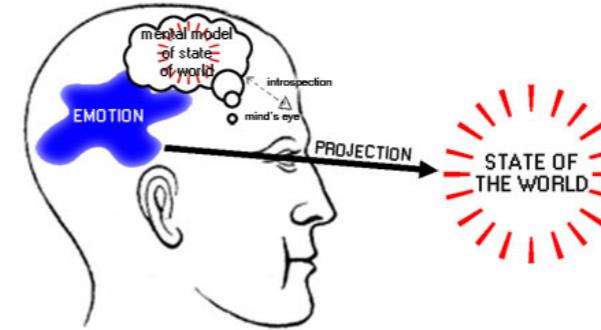
Cognitive Models



- **Hierarchical models**
Represent a user's task and goal structure
- **Linguistic models**
Represent the user-system grammar
- **Physical and device models**
Represent human motor skills
- **Cognitive architecture**
Underline all all of these cognitive models

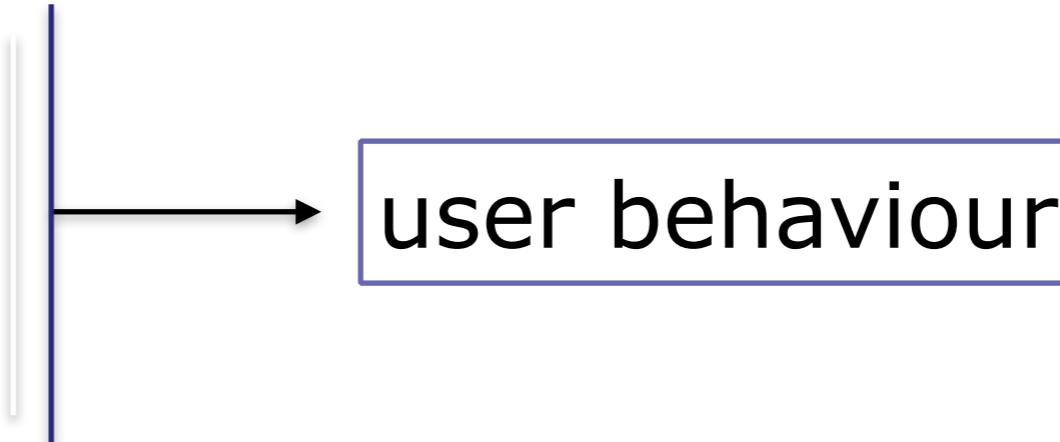


Cognitive Models



- They model aspects of user:

- understanding
- knowledge
- intentions
- processing



- Common categorisation:

- Competence vs. Performance
- Computational flavour
- No clear divide

Hierarchical models

Goal and task hierarchies

- Mental processing as **divide-and-conquer**
- Example: to make a sales report we have to
 - produce report**
 - gather** data
 - . **find** book names
 - . . **do** keywords search of names database
 - . . . *... further sub-goals*
 - . . . **sift** through names and abstracts by hand
 - . . . *... further sub-goals*
 - . **search** sales database - further sub-goals
 - layout** tables and histograms => further sub-goals
 - write** description => further sub-goals

The **GOMS** Cognitive Model

Goals

- What the user wants to achieve

Operators

- Basic actions the user performs

Methods

- Multiple possible paths to goal achievement

Selection

- Means of choosing between competing methods

GOMS example

GOAL: CLOSE-WINDOW

- . [select **GOAL: USE-MENU-METHOD**
 - . MOVE-MOUSE-TO-FILE-MENU
 - . PULL-DOWN-FILE-MENU
 - . CLICK-OVER-CLOSE-OPTION**GOAL: USE-CTRL-W-METHOD**
 - . PRESS-CONTROL-W-KEYS]

For a particular user (e.g. Sam):

Rule 1: Select USE-MENU-METHOD unless another rule applies

Rule 2: If the application is GAME, select CTRL-W-METHOD

Issues for goal hierarchies

- Granularity
 - Where do we start?
 - Where do we stop?
- Routine learned behaviour, not problem solving
 - The unit task
- Conflict
 - More than one way to achieve a goal
- Error Detection (e.g. some British ATMs)

GOMS and Closure

GOAL: GET-MONEY

- **GOAL:** USE-ATM
- • INSERT-CARD
- • ENTER-PIN
- • ENTER-AMOUNT
- • COLLECT-MONEY

<< outer goal now satisfied goal stack popped >>

- • COLLECT-CARD – subgoal operators missed



What was the problem?

Closure achieved within a sub-goal, before complete all actions in that level.

A success case

DESIGN FOCUS

GOMS saves money



Some years ago the US telephone company NYNEX were intending to install a new computer system to support their operators. Before installation a detailed GOMS analysis was performed taking into account the cognitive and physical processes involved in dealing with a call. The particular technique was rather different from the original GOMS notation as described here. Because an operator performs several activities in parallel a PERT-style GOMS description was constructed [192, 154]. The PERT analysis was used to determine the critical path, and hence the time to complete a typical task. It was discovered that rather than speeding up operations, the new system would take longer to process each call. The new system was abandoned before installation, leading to a saving of many millions of dollars.



Exercise

*Create a **GOMS** description of the task of photocopying an article from a journal. Discuss the issue of closure in terms of your GOMS description.*

Exercise

GOAL: PHOTOCOPY-PAPER

- **GOAL:** LOCATE-ARTICLE
- **GOAL:** PHOTOCOPY-PAGE **repeat** until no more pages
 - **GOAL:** ORIENT-PAGE
 - **GOAL:** OPEN-COVER
 - **GOAL:** SELECT-PAGE
 - **GOAL:** POSITION-PAGE
 - **GOAL:** CLOSE-COVER
 - **GOAL:** PRESS-COPY-BUTTON
 - **GOAL:** VERIFY-COPY
 - LOCATE-OUT-TRAY
 - EXAMINE-COPY
- **GOAL:** COLLECT-COPY
 - LOCATE-OUT-TRAY
 - REMOVE-COPY
- **GOAL:** RETRIEVE-JOURNAL
 - OPEN-COVER
 - REMOVE-JOURNAL
 - CLOSE-COVER

Exercise

GOAL: PHOTOCOPY-PAPER

- **GOAL:** LOCATE-ARTICLE
- **GOAL:** PHOTOCOPY-PAGE **repeat** until no more pages
 - **GOAL:** ORIENT-PAGE
 - **GOAL:** OPEN-COVER
 - **GOAL:** SELECT-PAGE
 - **GOAL:** POSITION-PAGE
 - **GOAL:** CLOSE-COVER
 - **GOAL:** PRESS-COPY-BUTTON
 - **GOAL:** VERIFY-COPY
 - **GOAL:** LOCATE-OUT-TRAY
 - **GOAL:** EXAMINE-COPY
- **GOAL:** COLLECT-COPY
 - LOCATE-OUT-TRAY
 - REMOVE-COPY (*outer goal satisfied => error*)
- **GOAL:** RETRIEVE-JOURNAL
 - OPEN-COVER
 - REMOVE-JOURNAL
 - CLOSE-COVER