Consistency (1)

- Consistency... be systematic
  - lexical
  - syntactic
  - semantic levels

- Why consistency?
  - Makes things easier to remember.
  - Aids in generalizability.
  - Helps reduce potential for error.

- Modeling approach
  - Grammars, e.g. BNF

Consistency (2)

- Lexical Consistency
  - Coding consistent with common usage, e.g.
    - red = bad, green = good
    - left = less, right = more
  - Consistent abbreviation rules
  - Equal length or first set of unambiguous chars.

- Devices used same way in all phases
  - Character delete key is always the same

- Syntactic Consistency
  - Error messages placed at same (logical) place
  - Always give command first - or last
  - Apply selection consistently, e.g. select text then apply tool or select tool and then apply to a text
  - Menu items always at same place in menu (muscle memory)

Chapter 2
Basics of HCl and History

- 2.1 Motivation
- 2.2 Principles for UI-Design
- 2.3 Understanding Errors
- 2.4 Consistency
- 2.5 Basic Models
- 2.6 A Brief History of HCl

Consistency (3)

- Semantic Consistency
  - Global commands always available
    - Help
    - Abort (command underway)
    - Undo (completed command)
  - Operations valid on all reasonable objects
    - If object of class "X" can be deleted, so can object of class "Y"

- Applicability
  - to command line user interfaces
  - Keyboard short cuts
  - Speech interfaces
  - Tool bars
  - Menus
  - Selection operation
  - Gestures

Consistency through Grammars

- Example - Task-Action-Grammer (TAG)
  - Task[direction,unit] -> symbol[direction] + letter[unit]
  - Symbol[direction=forward] -> "CTRL"
  - Symbol[direction=backward] -> "ALT"
  - Letter[unit=word] -> "W"
  - Letter[unit=paragraph] -> "P"

- Example - Commands
  - Move cursor on word forward: CTRL-W
  - Move cursor on word backward: ALT-W
  - Move cursor on paragraph forward: CTRL-P
  - Move cursor on paragraph forward: ALT-P
Chapter 2
Basics of HCI and History

- 2.1 Motivation
- 2.2 Principles for UI-Design
- 2.3 Understanding Errors
- 2.4 Consistency
- 2.5 Basic Models
- 2.6 A Brief History of HCI

How does the Format Brush work?

- compare it to bold, italic, underline, ...

Consistency in GUIs

- Format Brush
  1. place the cursor in the format you want to use
  2. switch the format brush on
  3. mark the area that should get the new format

- Bold face font (1)
  1. Mark the text that should become bold
  2. Click the toolbar button for bold
  3. Switch it off when ready

- Bold face font (2)
  1. Switch bold face font on (Click the toolbar button for bold)
  2. Write text

Models & Theories

- What are models and theories used for?
  - explanatory
  - predictive
  - descriptive/taxonomy

- Models on different levels
  - concept
  - human action
  - dialog
  - keystroke

Inconsistency

- Dragging file operations?
  - folder on same disk vs. folder on different disk
  - file to trashcan vs. disk to trashcan

- Sometimes inconsistency is wanted
  - E.g. Getting attention for a dangerous operation
  - Use inconsistency very carefully!

- Inconsistency at one level may be consistent at another
  - moving icon to file cabinet, mailbox, or trash causes icon to disappear (Xerox Star)
  - choices for when dragging file icon to printer icon:
    - delete the icon (and thus the file)
    - disappears "in" the printer from where it can be retrieved
    - return icon to original location

Example Motivation - Prediction

- Convert 712 GBP into EUR
- Hand is on the mouse to start with
- How long will it take?

Chapter 2
Basics of HCI and History

- 2.1 Motivation
- 2.2 Principles for UI-Design
- 2.3 Understanding Errors
- 2.4 Consistency
- 2.5 Basic Models
- 2.6 A Brief History of HCI

How does the Format Brush work?

- compare it to bold, italic, underline, ...

Consistency in GUIs

- Format Brush
  1. place the cursor in the format you want to use
  2. switch the format brush on
  3. mark the area that should get the new format

- Bold face font (1)
  1. Mark the text that should become bold
  2. Click the toolbar button for bold
  3. Switch it off when ready

- Bold face font (2)
  1. Switch bold face font on (Click the toolbar button for bold)
  2. Write text

Models & Theories

- What are models and theories used for?
  - explanatory
  - predictive
  - descriptive/taxonomy

- Models on different levels
  - concept
  - human action
  - dialog
  - keystroke

Inconsistency

- Dragging file operations?
  - folder on same disk vs. folder on different disk
  - file to trashcan vs. disk to trashcan

- Sometimes inconsistency is wanted
  - E.g. Getting attention for a dangerous operation
  - Use inconsistency very carefully!

- Inconsistency at one level may be consistent at another
  - moving icon to file cabinet, mailbox, or trash causes icon to disappear (Xerox Star)
  - choices for when dragging file icon to printer icon:
    - delete the icon (and thus the file)
    - disappears "in" the printer from where it can be retrieved
    - return icon to original location

Example Motivation - Prediction

- Convert 712 GBP into EUR
- Hand is on the mouse to start with
- How long will it take?
Plans and Situated Actions

Distributed Cognition

- complex interaction between people
- interaction with different devices
- interaction with information in different forms
- complex interaction with the physical environment
- interruptions as standard phenomenon of live

- Computer usage can not be seen isolated from that

- Suchman, 1990
  - human plans are often not orderly executed
  - plans are often adapted or changed
  - user's actions are situated in time and place
  - user's actions are responsive to the environment
  - distributed cognition – knowledge is not just in the user's head it is in the environment

Background: The Psychology of Everyday Action (Norman 2002, Chapter 2)

- People are blaming themselves for problems caused by design
  - If the system crashes and the user did everything as he is supposed to do the developer/system is blamed
  - If the system crashes and the user operated the system wrongly the user is blamed

- People have misconceptions about their actions
  - The model must not be fully correct – it must explain the phenomenon

- People try to explain actions and results
  - Random coincidence may lead to assumptions about causality

Action Cycle

Stages of Execution

- Goal
  - translated into
- An intention to act as to achieve the goal
  - translated into
- The actual sequence of actions that we plan to do
  - translated into
- The physical execution of the action sequence

Stages of Evaluation

- Perceiving the state of the world
  - followed by
- Interpreting the perception according to our expectations
  - followed by
- Evaluation of the interpretations with what we expected to happen (original intentions)
  - followed by
- Goal

Seven Stages of Action

1. Forming a goal
2. Forming an intention
3. Specifying an action
4. Executing the action
5. Perceiving the system state
6. Interpreting the system state
7. Evaluating the outcome

The World
Gulf of Execution

- The difference between the intentions and the allowable actions is the Gulf of Execution
  - How directly can the actions be accomplished?
  - Do the actions that can be taken in the system match the actions intended by the person?
- Example in GUI
  - The user wants a document written on the system in paper (the goal)
  - What actions are permitted by the system to achieve this goal?
- Good design minimizes the Gulf of Execution

Fitts’ Law

Predicting Movement Time (MT)

- $MT = a + b \log_2(2A/W)$
  - $A$: amplitude
  - $W$: width
  - $a$, $b$: constants dependent on the input device
  - Fitts’ law predicts that the time to acquire a target is logarithmically related to the distance over the target size.
  - $MT = a + b \log_2(A/W + 1)$
    - improvement of the original Fitts’ law

- Good design minimizes the Gulf of Execution

Gulf of Evaluation

- The Gulf of Evaluation reflects the amount of effort needed to interpret the state of the system how well this can be compared to the intentions
  - Is the information about state of the system easily accessible?
  - Is it represented to ease matching with intentions?
- Example in GUI
  - The user wants a document written on the system in paper (the goal)
  - Is process observable? Are intermediate steps visible?
- Good design minimizes the Gulf of Evaluation

Fitts’ Law – index of difficulty

- How difficult the motor pointing task is
- $ID$: Index of Difficulty
  - $ID = \log_2(A/W + 1)$
  - $ID$ has the unit bits
- $MT = a + b \cdot ID$
  - $a$: has the unit s
  - $b$: has the unit s/bits
- Collect data set and calculate $a$ and $b$
  - $a$ can be negative

Implications on Design

- Principles of good design (Norman)
  - Stage and action alternatives should be always visible
  - Good conceptual model with a consistent system image
  - Interface should include good mappings that show the relationship between stages
  - Continuous feedback to the user
- Critical points/failures
  - Inadequate goal formed by the user
  - User does not find the correct interface / interaction object
  - User may not be able to specify / execute the desired action
  - Inappropriate / mismatching feedback

Fitts’ law in practice

- $MT = a + b \log_2(A/W + 1)$
  - $A$: distance from starting position
  - $W$: size of target along line of motion (for a 2-D target use smaller of height or depth)
  - Common values $a=50ms$, $b=150ms/bit$

Experimental data for pointing devices

$MT = a + b \times ID$, where $ID = \log(2/A/W + 1)$. 

---

### Hick’s Law

- The time needed to make a selection is proportional to the log number of alternatives given.
- $H$ is the information-theoretic entropy of a decision.
- $T = b \times H$.
- In alternatives of equal probability $H = \log(2(n + 1))$.
- Alternatives of unequal probability $H = \sum p_i \log(1/p_i + 1)$.
- Common practical values: $b=150$ ms/bit.

---

### GOMS

**Goals, Operators, Methods, Selection Rules**

- GOMS techniques produce quantitative and qualitative predictions of how people will use a proposed system.
- Different models proposed.
- Basics:
  - Goals – goal a user wants to accomplish (in real scenarios hierarchical).
  - Operators – operation (at a basic level) that are used to achieve a goal.
  - Methods – sequence of operators to achieve a goal.
  - Selection Rules – selection of method for solving a goal (if alternatives are given).

---

### Object-Action Interface Model (OAI)

- Targeted at GUIs and applications in real world domains.
- Steps
  1. Understanding the task, including
     - Universe of the real world, objects, atoms
     - Actions user can apply to objects, intention to steps
  2. Create a metamorphic representation of interface objects and actions
     - Object representation – metaphor to pixel
     - Actions – from plan level to specific clicks.

---

### Example (adapted from Dix 2004, p. 423):

Close the window that has the focus (Windows XP)

- Compare three options:
  - **GOAL: CLOSE-WINDOW**
    - **KEY-SHORTCUT**
      - hold-ALT-key
      - press-F4-key
    - **CONTEXT-MENU**
      - Move-mouse-win-head
      - Open-menu (right click)
      - Left-click-close
    - **CLOSE-BUTTON**
      - Move-mouse-button
      - Left-click-button

  Rule 1: **USE-CLOSE-BUTTON** method if no other rule is given.
  Rule 2: **USE-KEY-SHORTCUT** method if no mouse is present.

---

### Example (adapted from Dix 2004, p. 424):

Copy a journal article

- **GOAL: PHOTOCOPY-PAPER**
  - **GOAL: COLLECT-COPY**
  - **GOAL: LOCATE-ARTICLE**
  - **GOAL: COPY-PAGE** repeat until no more pages
  - **GOAL: ORIENT-PAGE**
  - **GOAL: POSITION-PAGE**
  - **GOAL: OPEN-COVER**
  - **GOAL: SELECT-PAGE**
  - **GOAL: CLOSE-COVER**
  - **GOAL: PRESS-COPY**
  - **GOAL: VERIFY-COPY**
  - **GOAL: LOCATE-OUTPUT**
  - **EXAMINE-COPY**

(outer goal satisfied)

- **GOAL: RETRIEVE-ORIGINAL**
  - **GOAL: OPEN-COVER**
  - **GOAL: TAKE-ORIGINAL**
  - **CLOSE-COVER**

Likely that the users forget this.
Example (adapted from Dix 2004, p. 430):

Example of a Cash-Machine
Why you need to get your card before the money.

- Design to lose your card.
- Design to keep your card.

GOAL: GET-MONEY
- GOAL: USE-CASH-MACHINE
- INSERT-CARD
- SELECT-GET-CASH
- ENTER-AMOUNT
- COLLECT-MONEY (outer goal satisfied!)
- COLLECT-CARD

GOAL: GET-MONEY
- GOAL: USE-CASH-MACHINE
- INSERT-CARD
- SELECT-GET-CASH
- ENTER-AMOUNT
- COLLECT-CARD
- COLLECT-MONEY (outer goal satisfied!)

GOMS - Example

In order to understand GOMS models that have arisen in the last decade and the relationships between them, an analyst must understand the basic components of the model (goals, operators, methods, and selection rules) and the different computational forms that GOMS models take. In this section, we will describe these concepts in subsequent sections; we will categorize existing GOMS models according to these concepts.

Times for basic operators

- Experimentally measured

Keystroke-Level Model (KLM)

- Simplified analysis
- Only operators on keystroke-level
- No goals, no methods, no selection rules
- List of basic operators to do a task
  - KEystrokes or button presses (K)
  - Pointing with the mouse to a target (P)
  - Hand movement between mouse and keyboard (H)
  - Mental operators (M) placed by heuristics
  - Drawing (D)
  - System response (R)


Basic time estimation

- http://www.cc.gatech.edu/classes/cs6751_97_winter/Topics/user-model/ Dix et al. page: 438

Calculate overall time required

- \( T_{\text{task}} = T_{\text{acquire}} + T_{\text{execute}} \)
- \( T_{\text{execute}} = T_K + T_B + T_P + T_H + T_D + T_M + T_R \)
  - \( T_K \) = time for key presses
  - \( T_B \) = time for button presses / clicks
  - \( T_P \) = time for pointing
  - \( T_H \) = time moving hand between mouse and keyboard
  - \( T_D \) = time for drawing
  - \( T_M \) = time for mentally preparing
  - \( T_R \) = time for system response
### Example
- **Start the command shell in windows**
- **What to do?**
  - Click 'Start'
  - Click 'Execute'
  - Think of command
  - Type 'cmd'
  - Hit 'return key'
- KLM
  - P[to start] 1.10s
  - B[left click] 0.20s
  - P[to execute] 1.10s
  - B[left click] 0.20s
  - H 0.40s
  - M 1.35s
  - K[d] 0.28s
  - K[m] 0.28s
  - K[d] 0.28s
  - K[return] 0.28s
- \[ T = 2^P + 2^B + 4^K + H + M \]
  - 5.47s

### KLM - Example
- Convert 712 GBP into EUR
- Hand is on the mouse to start with

### Further reading
**User Interface Design With Matrix Algebra**
Harold Thimbleby
- Algebra analysis of interactive systems
- Proving properties of interactive systems

### Chapter 2
**Basics of HCI and History**
- 2.1 Motivation
- 2.2 Principles for UI-Design
- 2.3 Understanding Errors
- 2.4 Consistency
- 2.5 Basic Models
- 2.6 A Brief History of HCI

### Evolution of HCI ‘interfaces’
- 50s - Interface at the hardware level for engineers - switch panels
- 60-70s - Interface at the programming level - COBOL, FORTRAN
- 70-90s - Interface at the terminal level - command languages
- 80s - Interface at the interaction dialogue level - GUIs, multimedia
- 90s - Interface at the work setting - networked systems, groupware
- 00s - Interface becomes pervasive
  - RF tags, Bluetooth technology, mobile devices, consumer electronics, interactive screens, embedded technology
A Brief History of HCI

- Early machines used batch processing (e.g., punch card machines)
- Terminals with command line interfaces
- Graphical user interfaces with pointing device
- Multimodal user interfaces

Changing Interaction Paradigms

- Replacement of command-language
- Direct manipulation of the objects of interest
- Continuous visibility of objects and actions of interest
- Graphical metaphors (desktop, trash can)
- Windows, icons, menus and pointers
- Rapid, reversible, incremental actions

- Origins of direct manipulation and graphical user interfaces
  - Ivan Sutherland’s Sketchpad, 1963, object manipulation with a light pen (grabbing, moving, resizing)
  - Douglas C. Engelbart, 1968, Mouse, NLS
  - XEROX ALTO (50 units at Universities in 1978)
  - XEROX Star (1981)
  - Apple Macintosh (1984)

VisiCalc - Widespread use of an Interactive Application

- Instantly calculating electronic spreadsheet
- Early killer app for PCs
- Significant value to non-technical users
Lessons Learned from History

- Technology drives new user interface concepts and interaction metaphors
- New user interfaces create new applications
- Designs and user interface concepts evolve
- You cannot hide the user interface - good ideas spread out
- The first to come out with a new user interface is not necessarily the most successful

Technologies to look out for:
- Eye gaze detection
- Speech and gesture recognition
- EEG, ECG, EMG interfaces (e.g. http://www.biosemi.com/products.htm)

More GUIs

- Amiga 1985
- NextStep 1989
- Win 3.11 1992
- OS/2 1992

References

- S. Suchman, Plans and situated action. 1990
  http://www-2.cs.cmu.edu/~amulet/papers/uichistory.tr.html
- Software Arts and VisiCalc
  http://www.bricklin.com/historyintro.htm
- A. Cooper. About Face 2.0: Chapter 1 - Goal-Directed Design