Designing Interactive Systems I

Knowledge, Feedback, Errors, 7 Principles of Design

Prof. Dr. Jan Borchers
Media Computing Group
RWTH Aachen University

Winter Semester ’23/’24

https://hci.rwth-aachen.de/dis
Review

• What are the Seven Stages of Action?
  • Variations? Gulfs? Design implications?

• What are mappings, natural mappings? Types?

• What are constraints? How do they differ from affordances? Types?
Three Levels of Processing
1. Visceral Level

- Fast, completely subconscious
- Reflex action, impulse
- E.g., vertigo, feeling of warmth and happiness when basking in the sun
- Not exactly ‘emotions’, more like hard-coded responses
2. Behavioral Level

• The level of “classic usability”
• “Learned responses”, triggered by situations matching a pattern
• Mostly subconscious, fast, lower level of emotions
• E.g., sports, walking, etc.

• Behavioral action is associated with an expectation
  • Hope or fear: Am I doing the right set of actions? (feedback)
  • Relief or despair: Did things work out in the way I intended? (conceptual model)
3. Reflective Level

- Conscious thinking about events that have occurred
- Slow, deep thinking
- Highest level of emotions, e.g., guilt, pride, blame, praise
- Retained in memory
Design in Three Levels of Processing

- Visceral design: Make products “feel” great

- Behavioral design: Follow standard cognitive usability rules

- Reflective design: Create something users connect to (e.g., culture, meaning of a product)

- Excellent visceral and reflective design will make users forgive you small usability mistakes
Interplay with the Seven Stages of Action

Reflective
- Plan

Behavioral
- Specify
- Interpret

Visceral
- Perform
- Perceive

Goal

World
Knowledge in the World and in the Head

- **Experiment:**
  - Write down the digit layout of a telephone and a calculator keyboard
Knowledge in the World and in the Head

- Much knowledge is not in the head but in the world
- Despite less-than-perfect knowledge, precise behavior is possible—how?
- Behavior is determined by combination of knowledge in the world and in the head
- High precision of knowledge in the head is unnecessary
  - We only need the knowledge to be precise enough to distinguish the right behavior from the others possible
- Example: What is on the front and the back of the German 1 cent coin?
More Reasons Why This Works

• **Physical constraints** are in effect
  • They limit the actions possible
  • Example: What can be moved/combined/manipulated how when repairing your toaster?

• **Cultural constraints** are in effect
  • Social rules are learned once and are then applicable in many situations
  • Example: What to do upon entering a restaurant?
  • But: Cultural differences!
Knowledge in the Head & Constraints

• Traveling poets were able to recite poems with thousands of lines
  • Story works as semantic constraint
  • Rhyme works as “linguistic constraint”

• Constraints limit the amount of knowledge that needs to be learned

• Humans can minimize the amount/precision/depth of information to remember by arranging their environment and copying people’s behavior
  • This can even help people cover missing abilities (dyslexia) or mental disabilities
Example: Typing

- Exercise:
  - What kind of knowledge do beginners/intermediate/expert typists use?

- Beginner: Knowledge in the world (keyboard labeling)

- Intermediate: Knowledge in the world (peripheral vision, feeling keys) and in the head (knowing location of important keys by heart)

- Expert: All knowledge in the head, no eye contact to keyboard necessary anymore (cost/benefit tradeoff)
Example: City Map

• Exercise:
  • Try to write down exactly how to get from your home to the main university building

• Result:
  • Nobody has a perfect street/building map in their head; often entire parts are forgotten in route descriptions
  • Nevertheless we can get from A to B safely
  • Why? Signage and constraints (e.g., street numbers) supply external knowledge
Types of Knowledge

- **Declarative knowledge** ("what")
  - Facts (Bonn is southeast of Aachen)
  - Rules (stop at red traffic lights)
  - Easy to write down and teach (not learn!)

- **Procedural knowledge** ("how")
  - How to play an instrument
  - Hard to write down, subconscious
  - Hard to teach, best by demo/training

- Design can easily convey **declarative** knowledge
How Much Can We Remember?

- Random unconnected facts: little
  - “Press Ctrl-Alt-Delete to log on”
  - Not learnable per se, only via associations
  - Example: First 1,000 digits of π
  - If your recipe fails, you are lost

- Connected facts: more
  - Using associations
  - Example: motor bike directional indicator
The Daily Struggle

• Exercise:
  • How many online accounts with passwords do you have?
  • How many of these can you remember the passwords to?
  • For how many of them do you use the same password?

• Credit cards, bank accounts with bank codes, number plates, phone numbers/addresses/birthdays/age of friends, clothing sizes,…

• As the password requirements become more complex, the system becomes less secure, why?
  • We tend to move these things from the head into the world
Knowledge in the World: Characteristics

• Nothing to remember

• But: only there while you see it

• Especially difficult with things that are not very important to you

• Solution: Reminders
  • Paper agenda vs. PDA
  • Signal vs. message
Comparing Knowledge in the World and in the Head

• In the world:
  • Available as soon as visible
  • No learning needed
  • Low efficiency (interpreting needed)
  • High initial usability
  • Aesthetics difficult with much to display

• Remember: **Natural mappings** can save both learning and labeling

• In the head:
  • Less available
  • Less suitable for beginners
  • Harder to learn
  • But efficient
  • Invisible (less labels)
### Decision Structures

- To reduce chance of error, use either shallow or narrow decision trees
  - Shallow: No planning required, e.g., ice cream parlor menu
  - Narrow: No deep thinking required, e.g., cook book instructions, start your car, motorway exits
- Wide and deep structures:
  - Games like chess, etc.
  - Designed to occupy the mind
- Subconscious thought is effortless, associative, pattern-matching
- Conscious thought is slow, serial, demanding
Feedback
Feedback

- Feedback communicates to the user the current system state, success or failure of actions, and results of actions

- Good feedback:
  - Immediate
  - Informative and clear
  - The right amount
  - Prioritized
Sound

• Exercise:
  • Listen to everyday objects and their acoustic feedback (or think about it if not readily available in class)

• Examples: Pen cap, hard drive, bike lock, car door, telephone, software

• Sound is a unique information channel
  • Omnidirectional: blessing and curse

• But: Use it to convey meaning if possible!

• More on sound in DIS 2
Visibility and Feedback

• Invisible On/Off switch on the rear

• VCRs without on-screen programming required lengthy programming instructions without much visible feedback

• A good display is great to improve visibility, and therefore often usability

• This becomes more feasible as technology progresses (Augmented Reality/Ubicomp)
rear door handle
Feedforward

- Feedforward is to execution as feedback is to evaluation
- Information that helps you know what you can do
- Uses signifiers, constraints, and mappings
- Visual, but also haptic
  - Example: feeling keys before typing eyes-free on real vs. onscreen keyboard
Human Errors
Errors

• People make errors using everyday objects all the time

• Often blame themselves (untypical!)

• Often caused by taught helplessness
  • E.g., maths classes

• May lead to learned helplessness
  • Conspiracy of silence, depression

• Not only “dumb folk” have misconceptions of everyday life, and often those “wrong” models work better for everyday life
  • E.g., thermostats
Mistakes

• Result of conscious decision/thinking

• Often major events

• Reasons: Wrong goal, wrong plan, leaping to wrong conclusions, false causalities

• Hard to detect
Classes of Mistakes

- Memory-lapse: memory fails during goal-setting, planning, or evaluation
  - E.g., a mechanical failure because the mechanic was distracted while troubleshooting

- Knowledge-based: wrong evaluation of the situation because of incomplete knowledge
  - E.g., reporting the weight of an item in pounds instead of kilograms

- Rule-based: correct evaluation of the situation, but wrong course of action
  - E.g., blocking night club attendees from an emergency exit assuming they are avoiding payment
Slips

• Most everyday errors

• Small things going wrong

• Goal formed, but execution messed up

• Usually easy to discover

• Occur mostly in skilled behavior

• Often caused by lack of attention, busy, tired, stressed, bored, more important things to do,…

• We can only do one conscious thing at once
  • Jef Raskin, The Humane Interface: Walking and eating and solving a maths problem
Classes of Slips

• Action-based: the wrong action is performed
  • E.g., pouring a cup of coffee and milk and placing the cup in the fridge
  • Types: capture slips, description-similarity slips, mode errors

• Memory-lapse: memory fails, and the intended action is not done or its results not evaluated
  • E.g., forgetting to lock the door when leaving the house
Action-based Slips

• Capture slips
  • Two action sequences with similar initial but different later sequence
  • The sequence well practiced “captures” the unfamiliar one
  • Driving to work on a Sunday
  • Pocketing a borrowed pen
Action-based Slips

• Description-Similarity slip
  • Intention not described in enough detail, fitting 2 different action sequences
  • Often occurs if similar objects are physically close to each other (e.g., switches)
  • E.g., throwing t-shirt into toilet instead of laundry basket
  • Putting a lid onto the obviously wrong container
  • Pouring orange juice into your coffee pot
Action-based Slips

- Mode errors
  - Triggering the wrong action because the device is in a different mode than expected
  - Who has seen this in their favorite text editor: ":wq"?
  - Happens whenever devices resort to modes to cope with more functions than controls
  - The most prominent problem in many software user interfaces
Memory-Lapse Slips

• Memory lapses are common causes of errors

• Caused by interruption through other people or devices

• Forgetting to complete action sequence
  • E.g., walking into your bedroom, then wondering what you wanted to do here

• Sometimes because main part of goal is accomplished
  • E.g., ATM card in machine, originals in copier

• Minimize by
  • reducing the number of required steps
  • providing reminders of the steps
  • applying forcing functions
In-Class Exercise: Slips

- Think of three examples of slips that happened to you. What type are they?
  - Capture (driving to work)
  - Description-similarity (shirt in toilet)
  - Mode (vi)
  - Memory-lapse (ATM)
Detecting Errors

• Detecting slips is easier than mistakes, but requires visible feedback
  • Example: “Adjust the window!”

• Action-based slips are easier to detect than memory-lapses because the feedback is tangible

• Mistakes are hard to detect because nothing signals a wrong goal

• Problem: Finding the right level at which to correct
  • Are we doing this bottom-up?
  • The wrong car key
  • Confirmation is unlikely to catch errors
    • “Remove file blah.txt?”
  • Soft, reversible actions are better (e.g., trashcan), but people begin to rely on it
The Paradox of Automation

- When automation works, tasks are done as well or even better than by people.

- The paradox is that automation can take over dull and simple tasks, but not complex ones.

- When automation doesn't work, the results are unpredictable and could be dangerous, e.g., self-driven cars.
Designing for Error

- Assume all possible errors will be made
- Minimize the chance of errors occurring
- Minimize their effect if they are made
- Make them easy to detect
- Make them easy to reverse (undo)
- Watch people using your system (and their slips and mistakes)
- Don’t punish, don’t ignore
- Warning signals are ignored, warning features bypassed if inconvenient
Operation Could not be completed.

client-error-not-possible
You should have selected the CUE file and not this one.

I'll do it for you automatically this time, but don't do it again!
What to Do Now

- Read Norman’s book until page 216