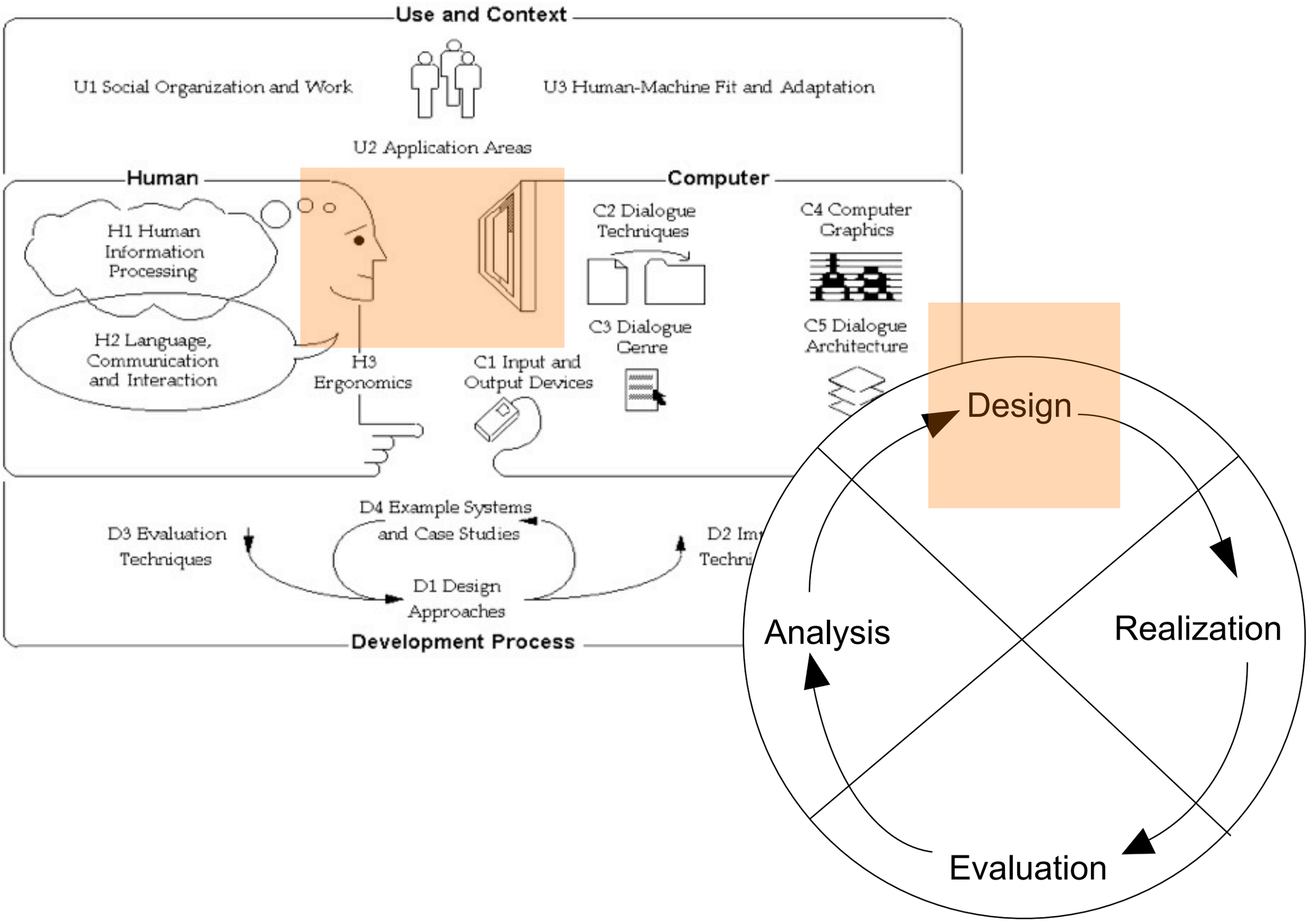
Mensch-Maschine-Interaktion 1

Chapter 3: Basic HCI principles

Basic HCI Principles

- **Users and Developers**
- 3 Usability Principles by Dix et al.
- 3 Usability Principles by Shneiderman
- Human Error
- Background: The Psychology of Everyday Action


Use and Context
U1 Social Organization and Work
U2 Application Areas
U3 Human-Machine Fit and Adaptation
Human
Computer
H1 Human Information Processing
H2 Language, Communication and Interaction
H3 Ergonomics
C1 Input and Output Devices
C2 Dialogue Techniques
C3 Dialogue Genre
C4 Computer Graphics
C5 Dialogue Architecture
D3 Evaluation Techniques
D4 Example Systems and Case Studies
D1 Design Approaches
Development Process
Design
Analysis
Realization
Evaluation

What the User Sees

- Users see only what is openly visible!

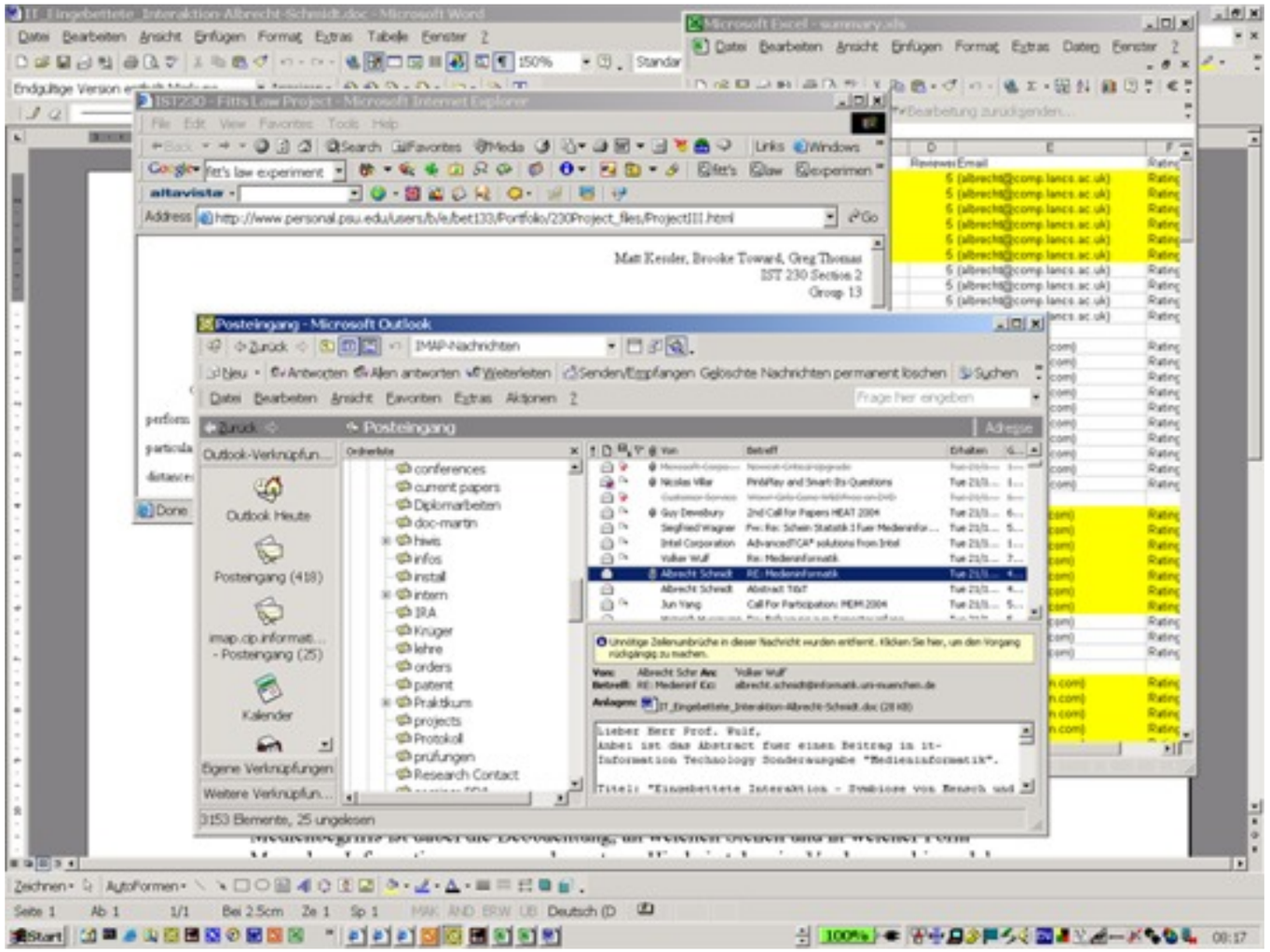
What the Developer Knows

- Users have little idea about:
  - architecture,
  - state transitions,
  - dependencies
  - application context
  - system restrictions
  - …

- And users often **do not want** to know about it.

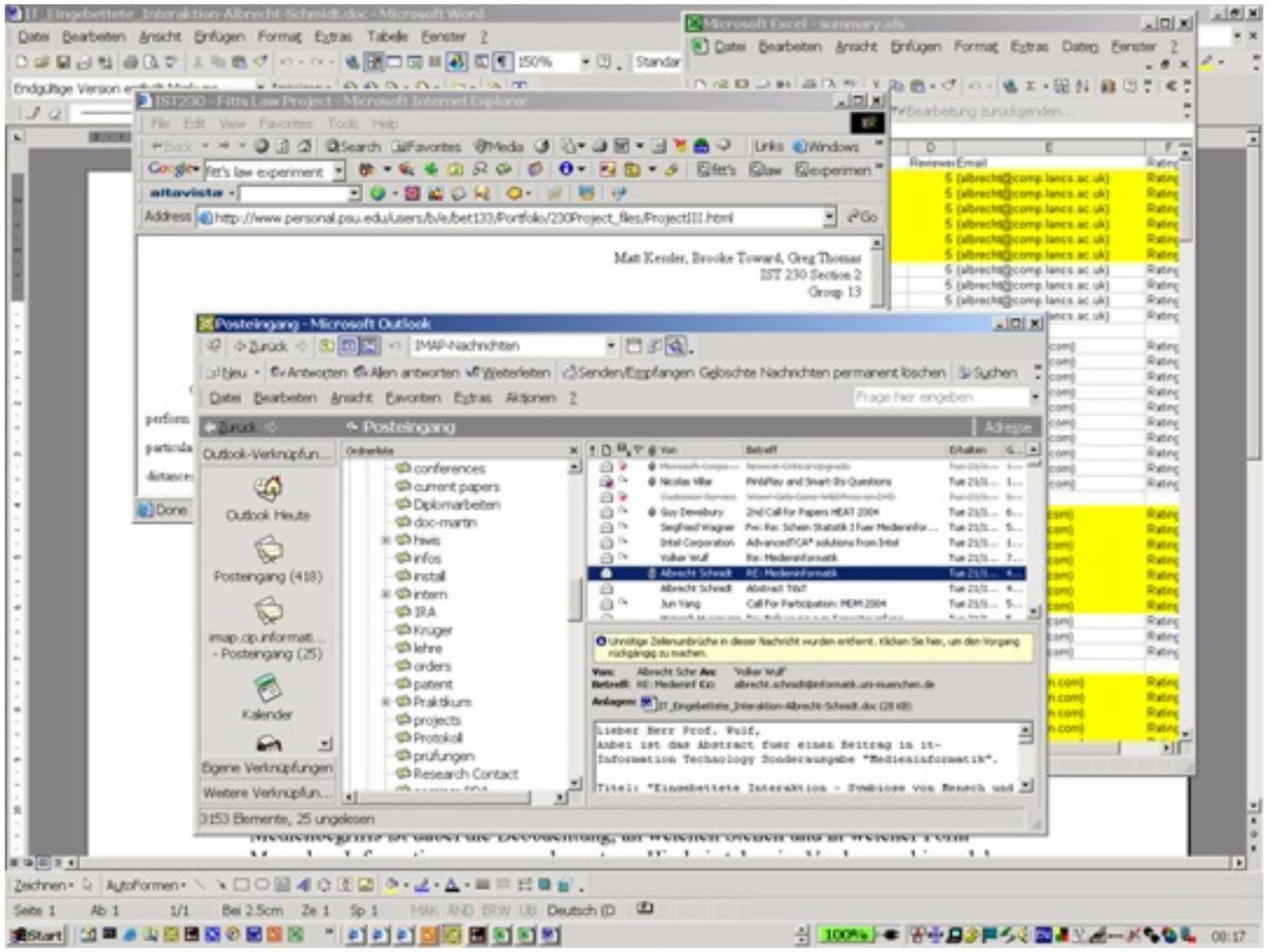
A Computer Screen and its Interpretation

- What do we see?
- What is shown?
- What is the meaning?



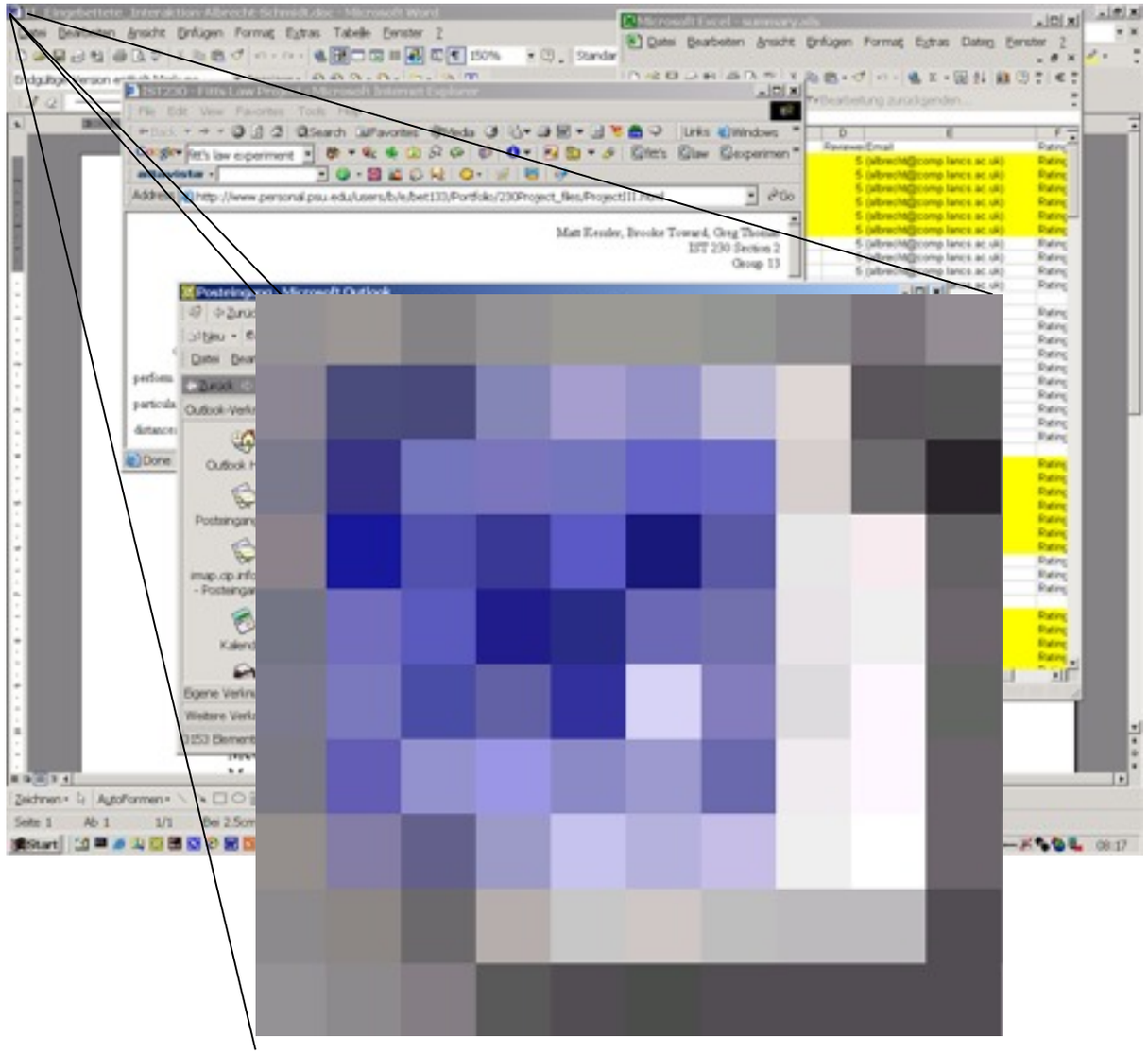
Answers from Skilled Computer Users

- Win2000 desktop
- Text and figures
- Icons and toolbars
- Overlapping windows
- Scroll bars and menus
- Task bar and status information
- Representations of documents



Basic (Naive) Technical Answers

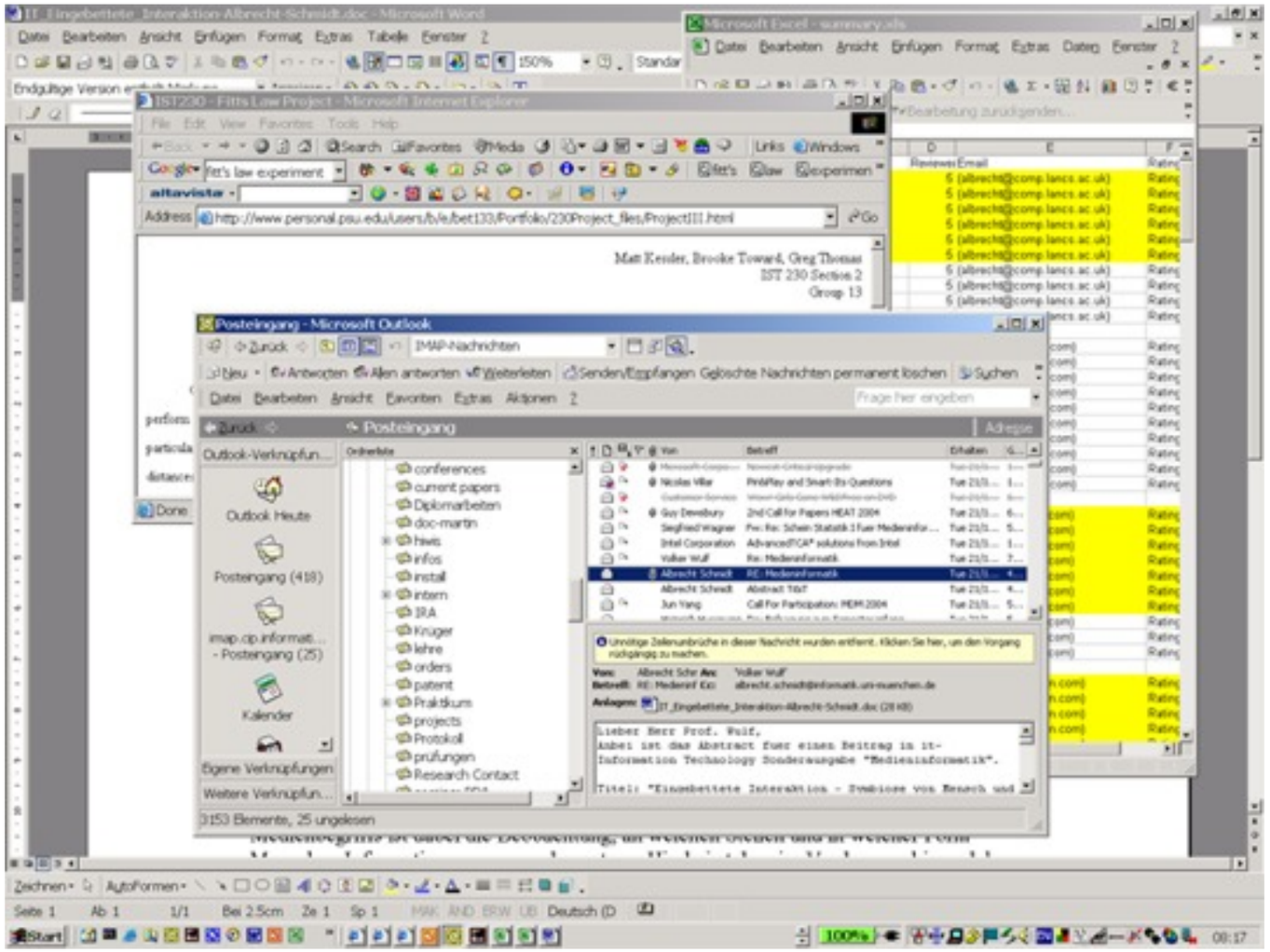
- 2-D surface
- Controllable pixels
- Image with a resolution of 1400x1050 pixels
- For each pixel the colour can be set
- The change of colour can be controlled rapidly



Perfect User's Answers

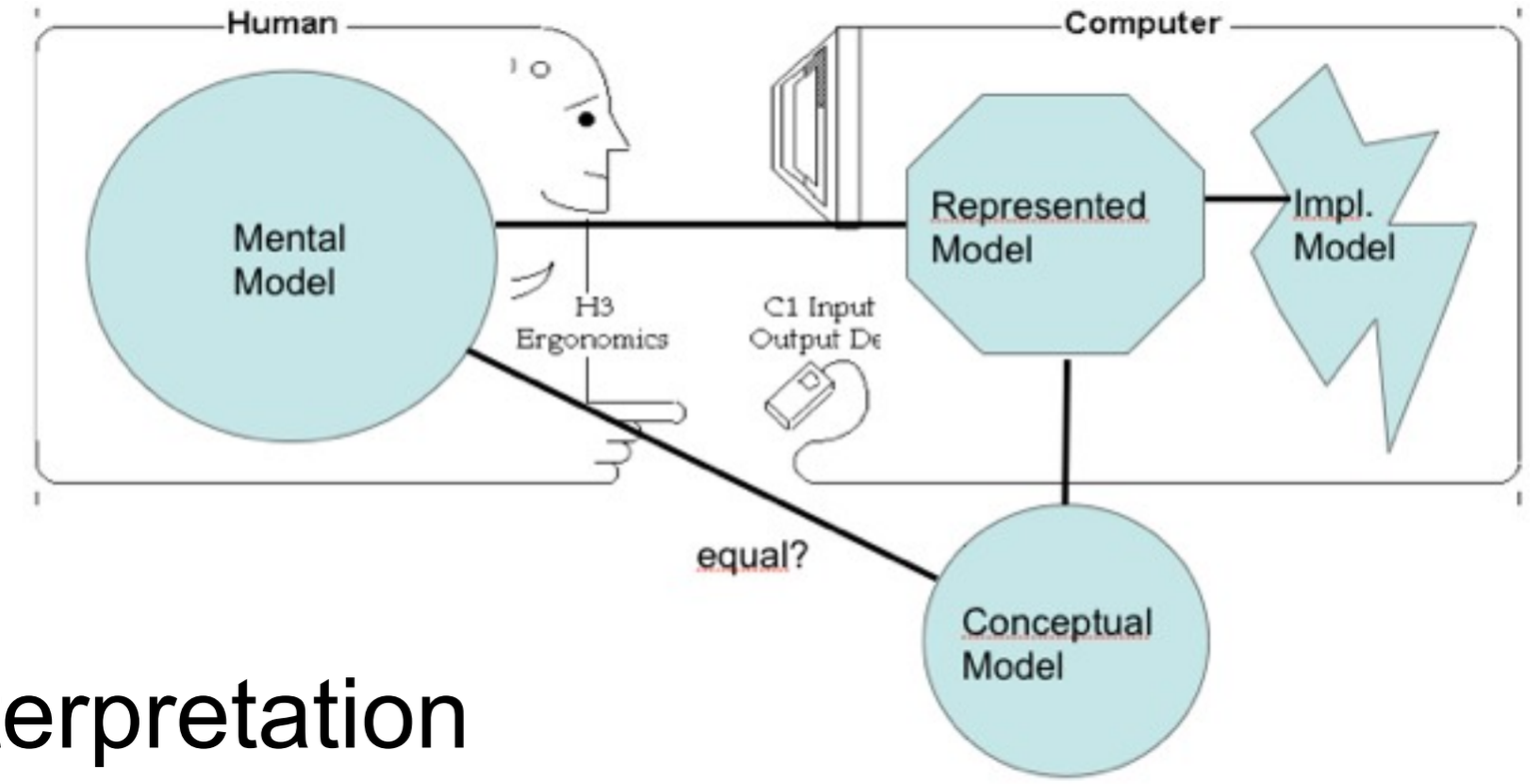
- My work environment

- Meeting notes
- Budget for next year
- Request to write a technical article
- Background information on a psychological phenomenon



What does this mean in terms of models?

- different levels in an interactive system
  - represented model
  - inplemented model



- different levels of interpretation
  - depending on computer skills
  - depending on domain knowledge
  - depending on familiarity
  - this will create the mental model!

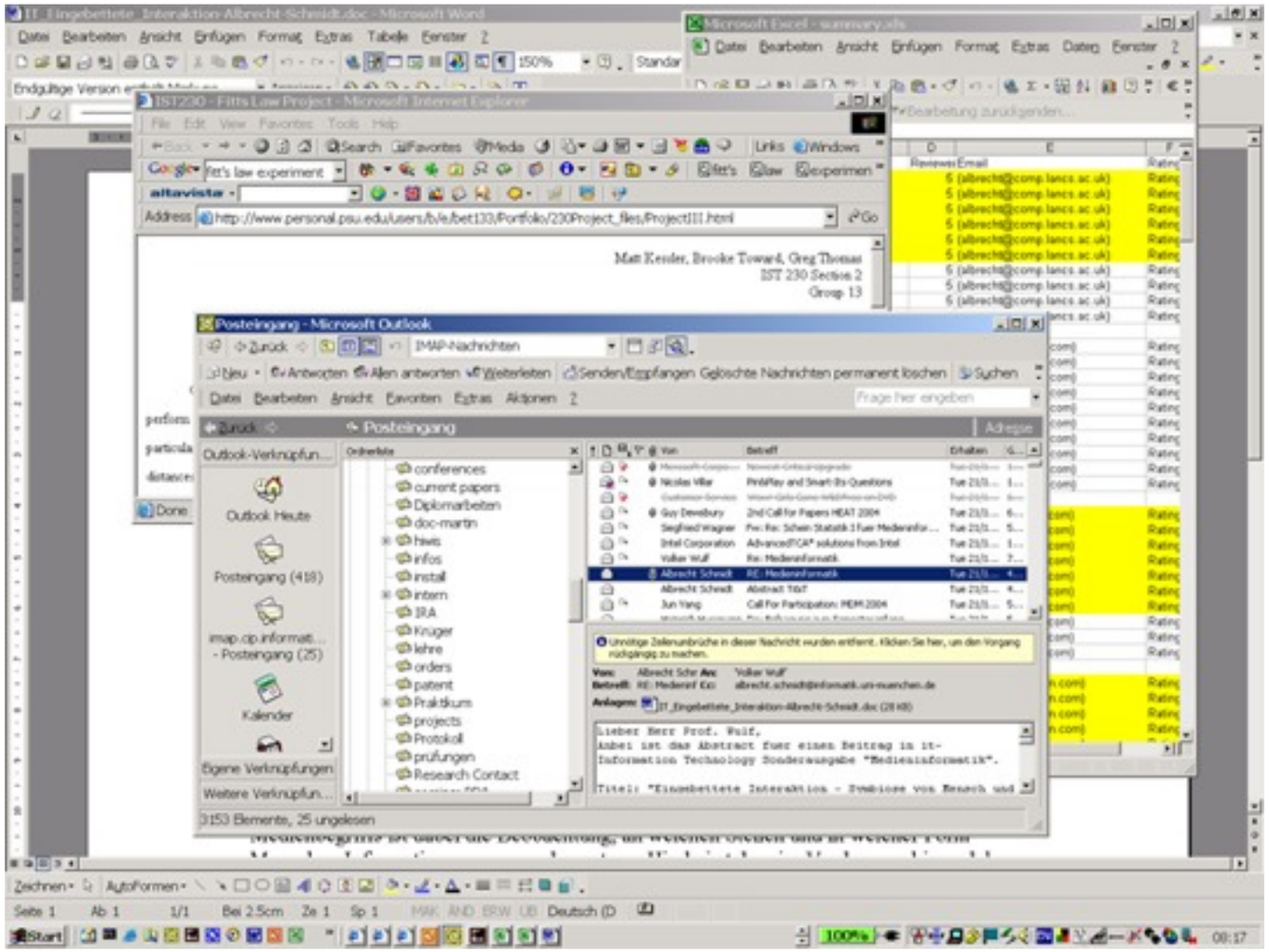
Metaphor Example 1 – Overlaying Windows

- What is the meaning of the fact that a window is behind another window?

- What is real?
  What is illusion?

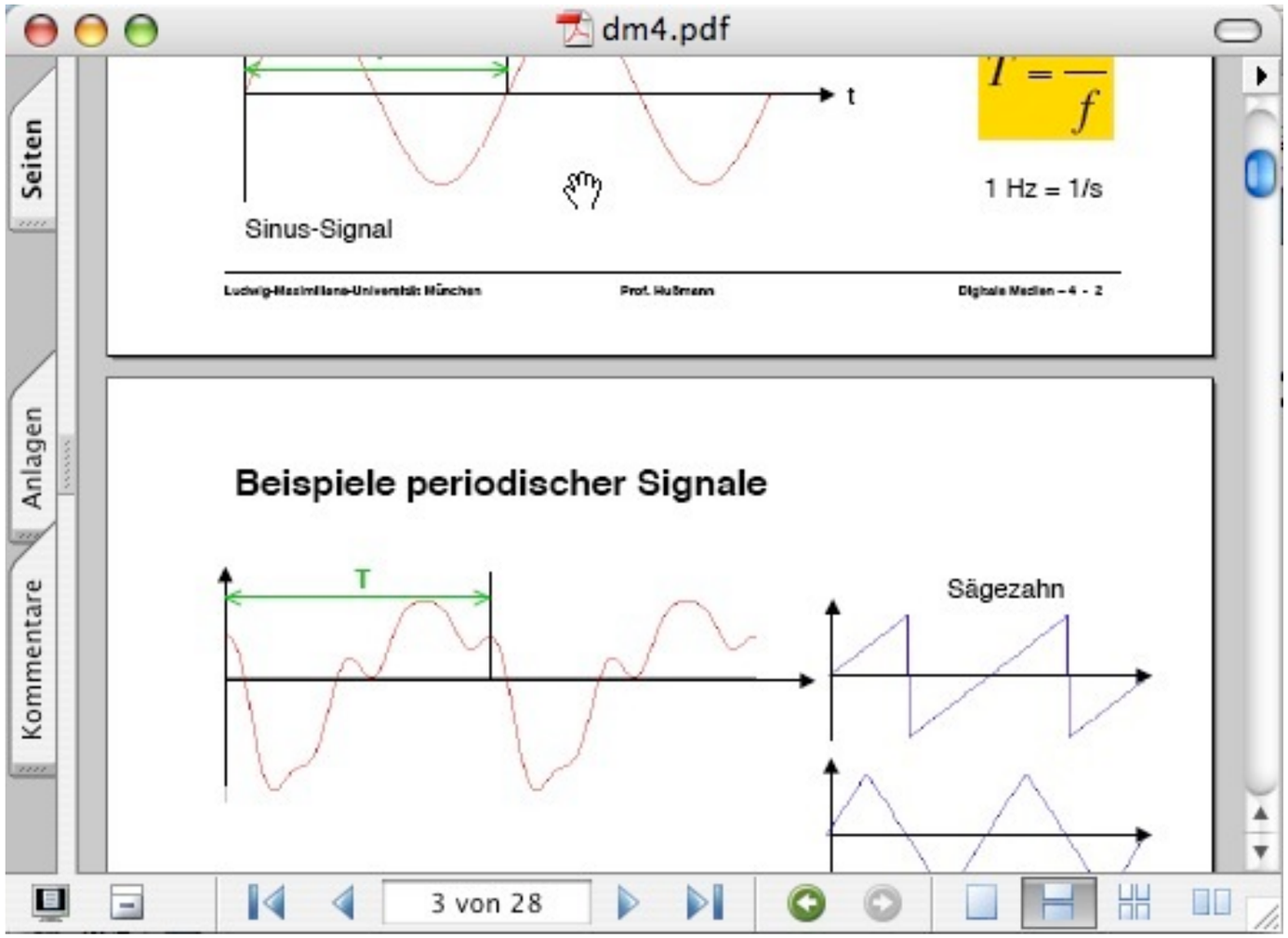
- What does iconizing do?

- Models?
  Conceptual…
  Implementation…
  Represented…



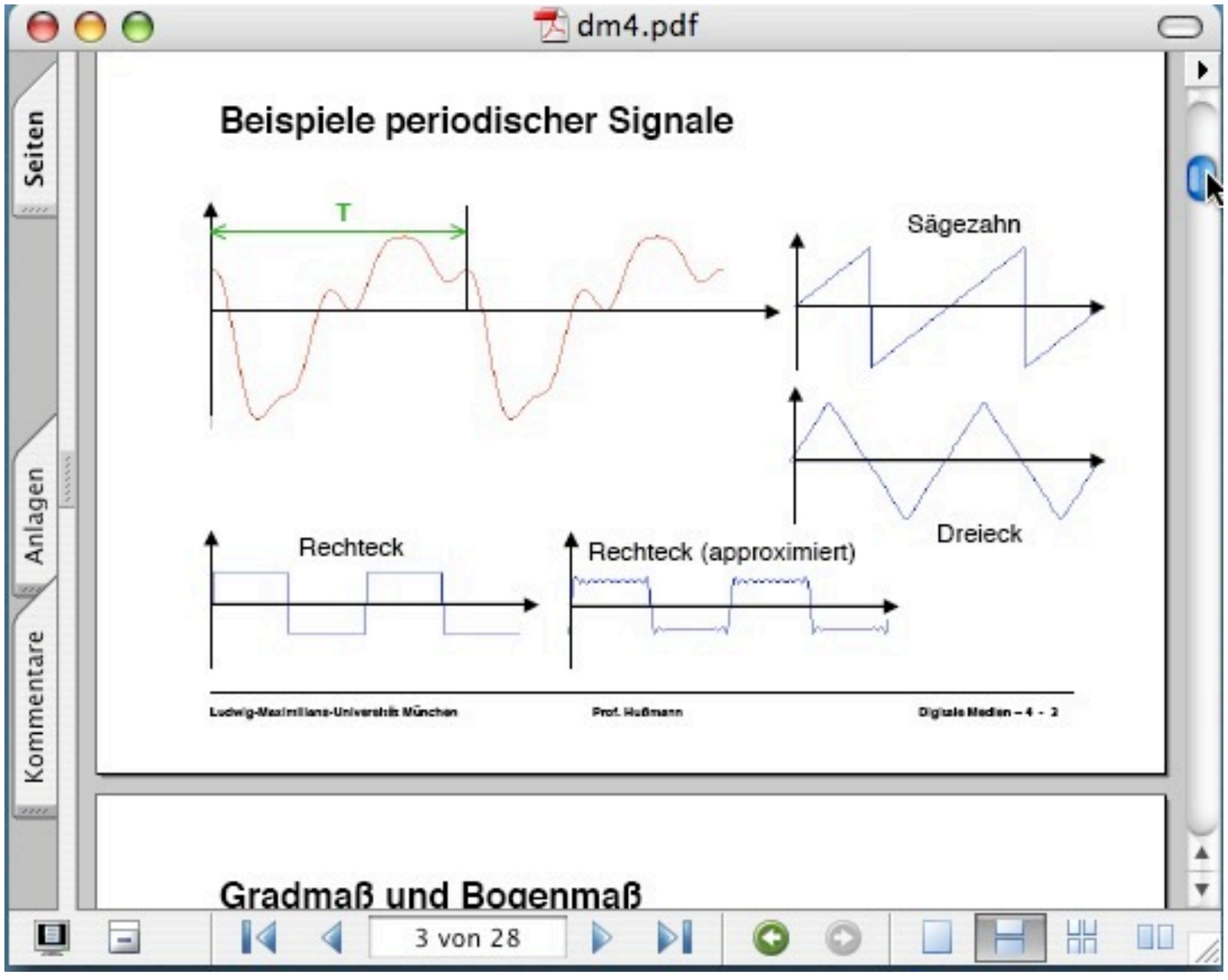
Metaphor Example 2 – Scrollbar vs. Hand

- Moving up the hand Moves up the document
- What happens in reality?
  What do we imagine?
  What is the metaphor?



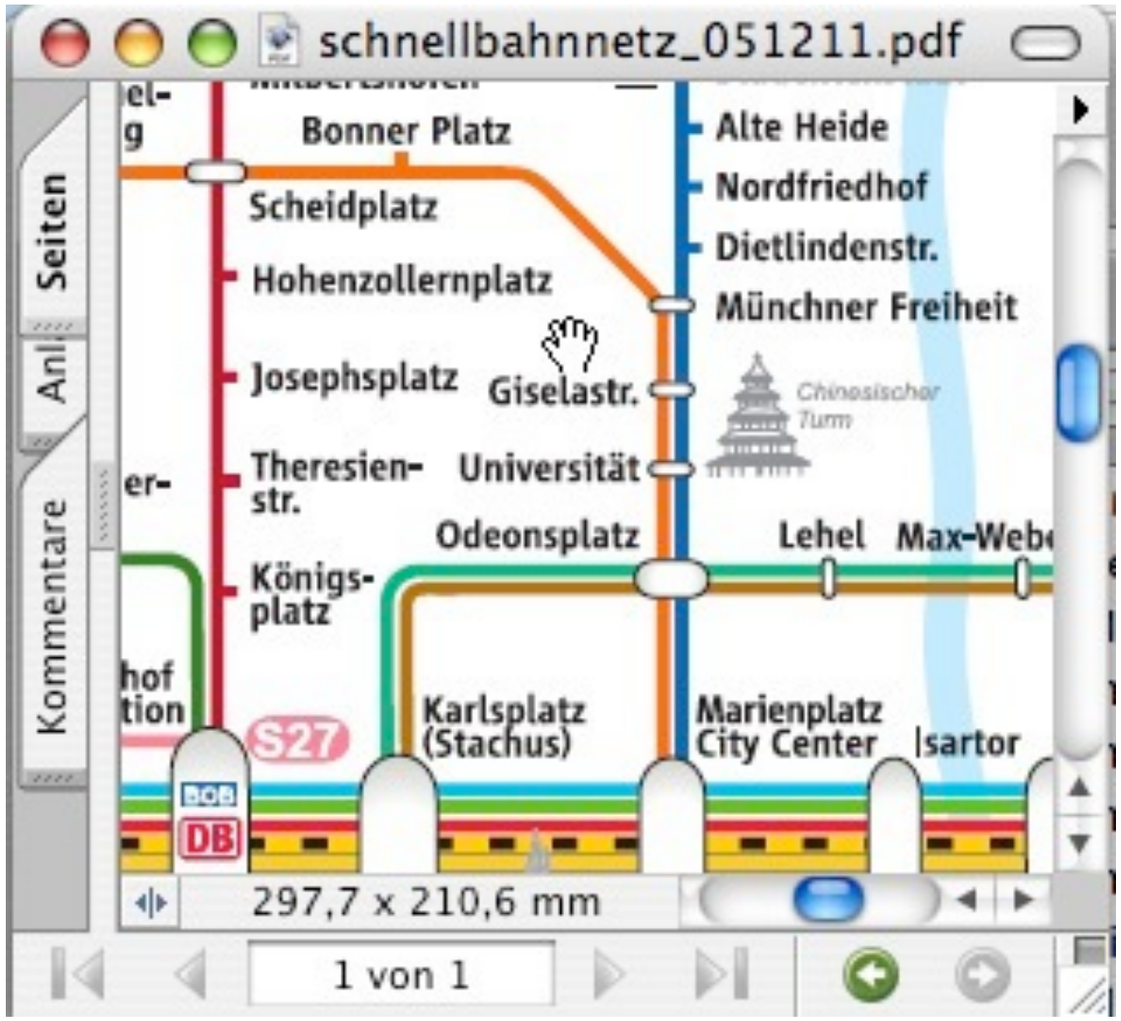
Metaphor Example 2 – Scrollbar vs. Hand

- Moving up the scroll bar moves down the document
- What happens in reality?
  What do we imagine?
  What is the metaphor?



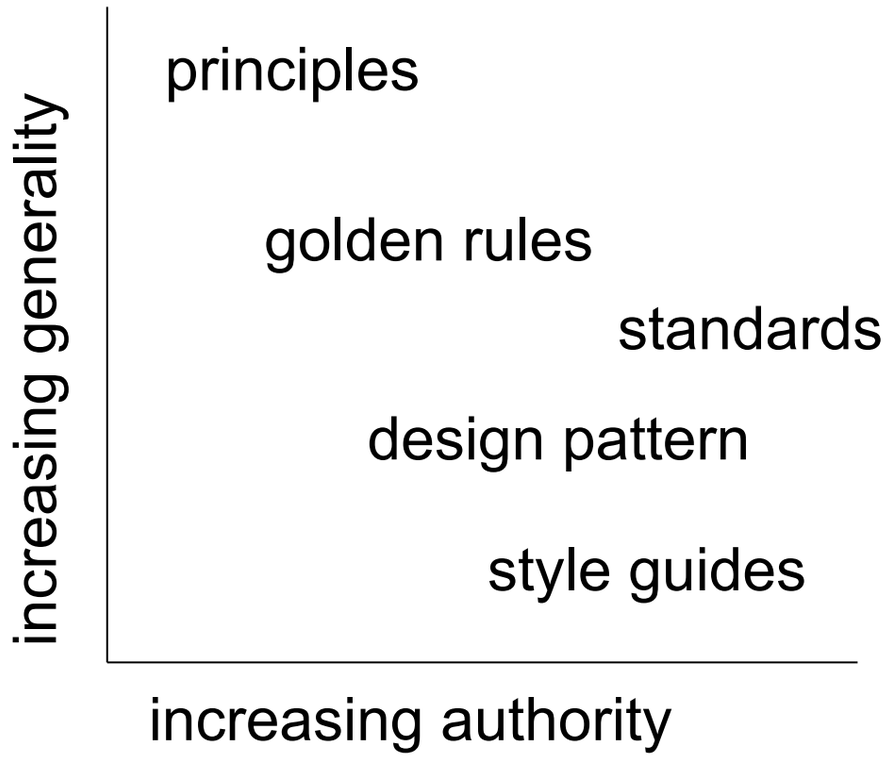
Metaphor Example 2 - Scrollbar vs. Hand

- Adequacy of interaction mechanism depends on content displayed



Types of Design Rules

- Principles
  - abstract design rules
- Golden rules and heuristics
  - more concrete than principles
- Standards
  - (very) detailed design rules
- Design pattern
  - generic solution for a specific problem



- Style guides
  - provided for devices, operating systems, widget libraries (Authority: whether or not a rule must be followed or whether it is just suggested, Generality: applied to many design situations or focused on specific application situation.)

Usability 101 (by Jakob Nielsen)

- “Usability is a quality attribute that assesses how easy user interfaces are to use. The word ‘usability‘ also refers to methods for improving ease-of-use during the design process.”

- Usability has five quality components:
  - **Learnability**: How easy is it for users to accomplish basic tasks the first time they encounter the design?
  - **Efficiency**: Once users have learned the design, how quickly can they perform tasks?
  - **Memorability**: When users return to the design after a period of not using it, how easily can they reestablish proficiency?
  - **Errors**: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
  - **Satisfaction**: How pleasant is it to use the design?

Basic HCI Principles

- Users and Developers
- **3 Usability Principles by Dix et al.**
- 3 Usability Principles by Shneiderman
- Human Error
- Background: The Psychology of Everyday Action
- Constraints, Mappings, Affordances




3 Usability Principles by Dix

- **Learnability**
  - Predictability
  - Synthesizability
  - Familiarity
  - Generalizability
  - Consistency
- **Flexibility**
  - Dialogue initiative
  - Multithreading
  - Task migratability
  - Substitutivity
  - Customizability
- **Robustness**
  - Observability
  - Recoverability
  - Responsiveness
  - Task conformance

[Section 7.2 in Dix.
“Human Computer Interaction”]

Principles to Support Usability

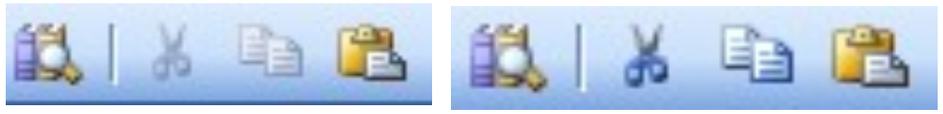
- **Learnability**
  - the ease with which new users can begin effective interaction and achieve maximal performance
- **Flexibility**
  - the multiplicity of ways the user and system exchange information
- **Robustness**
  - the level of support provided to the user in determining successful achievement and assessment of goal-directed behavior

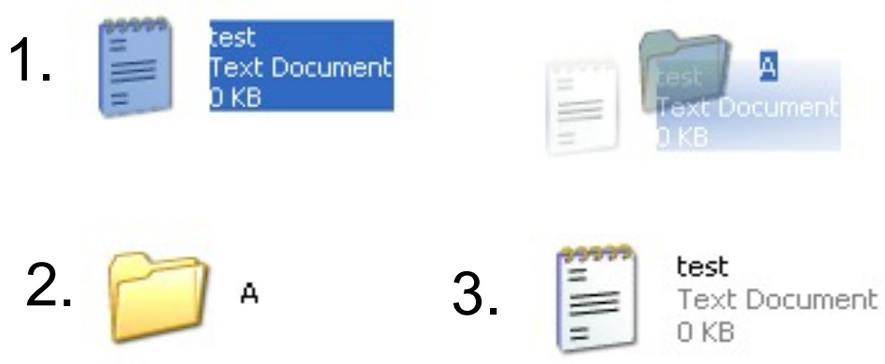
Dix, A. J., Finlay, J., Abowd, G., Beale, R. Principles to support usability, Human-Computer Interaction, 260-273, Third Edition

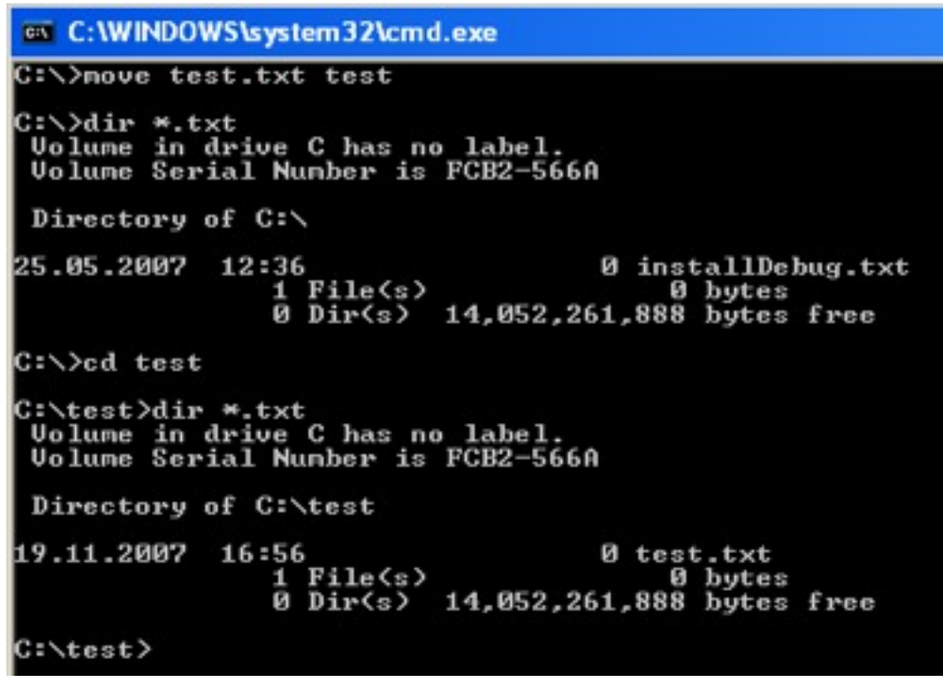
Principles of Learnability (1 / 2)

the ease with which new users can begin effective interaction and achieve maximal performance

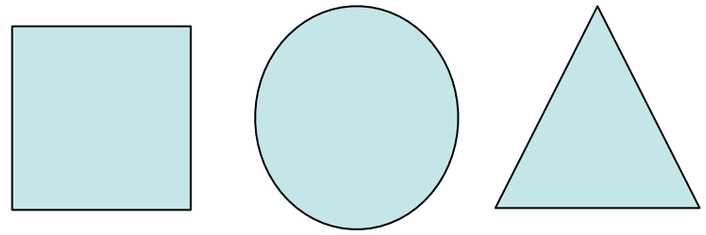
- **Predictability**
  - determining effect of future actions based on past interaction history
  - operation visibility
- **Synthesizability**
  - ability of the user to assess the effect of past operations on the current state
  - the user should see the changes of an operation
  - immediate vs. eventual feedback







Principles of Learnability (2 / 2)

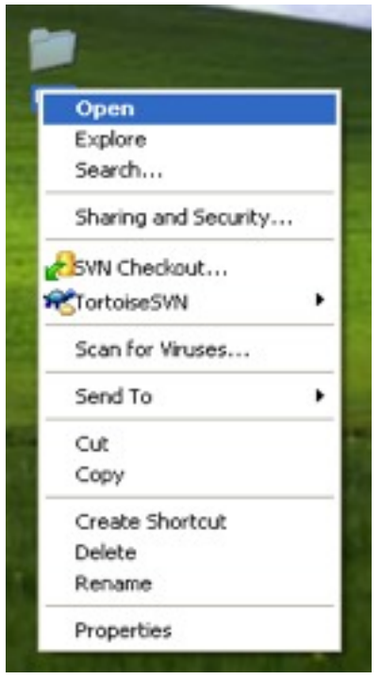


- **Familiarity**
  - how prior knowledge applies to new system
  - affordance (guessability)
- **Generalizability**
  - extending specific interaction knowledge to new situations
- **Consistency**
  - likeness in input/output behavior arising from similar situations or task objectives

Principles of Flexibility (1 / 6)

the multiplicity of ways the user and system exchange information

- **Dialogue initiative**
  - freedom from system-imposed constraints on input dialogue
  - user preemptiveness: user initiates dialog
  - system preemptiveness: system initiates dialog

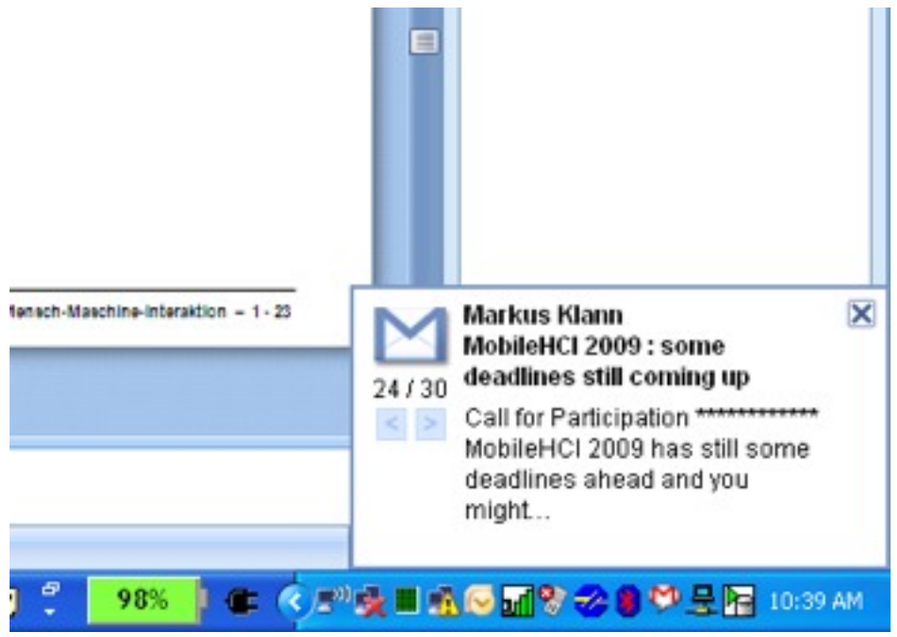


user preemptiveness

system preemptiveness

Principles of Flexibility (2 / 6)

- **Multithreading**
  - ability of system to support user interaction for several tasks at a time
  - **concurrent** multimodality: simultaneous communication of information pertaining to separate tasks
    - multi-modal dialog
    - editing text and beep (incoming mail) at the same time
  - **interleaving** multimodality: permits temporal overlap between separate tasks, dialog is restricted to a single task
    - in any window system: window = task
    - modal dialogs
    - interaction with just one window at a given time

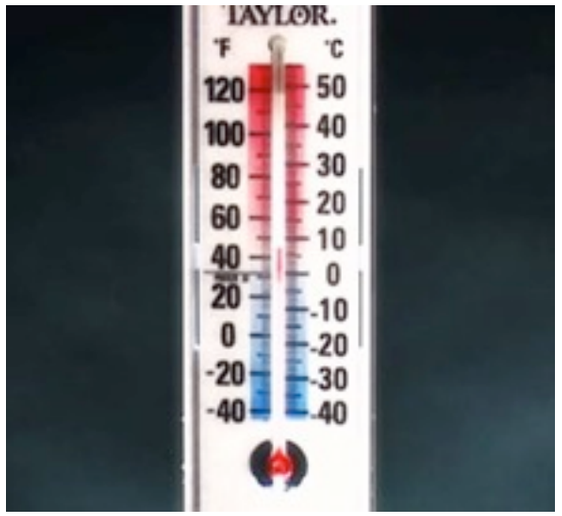


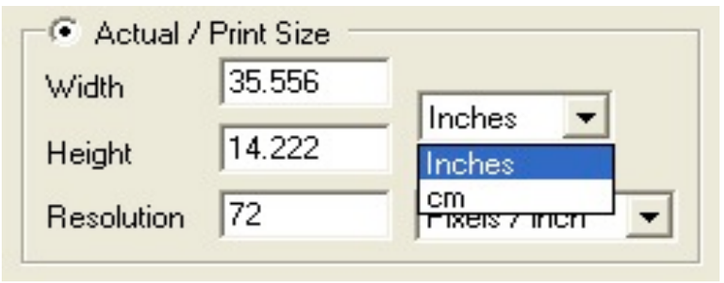
Principles of Flexibility (3 / 6)

- **Task migratability**
  - passing responsibility for task execution between user and system
  - example: spell checking

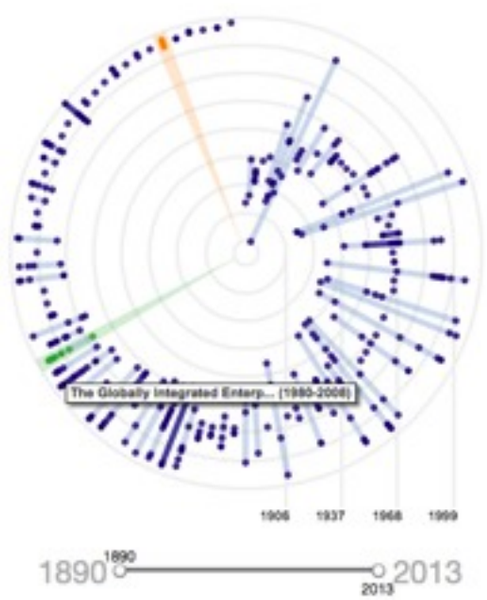
Principles of Flexibility (4 / 6)

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





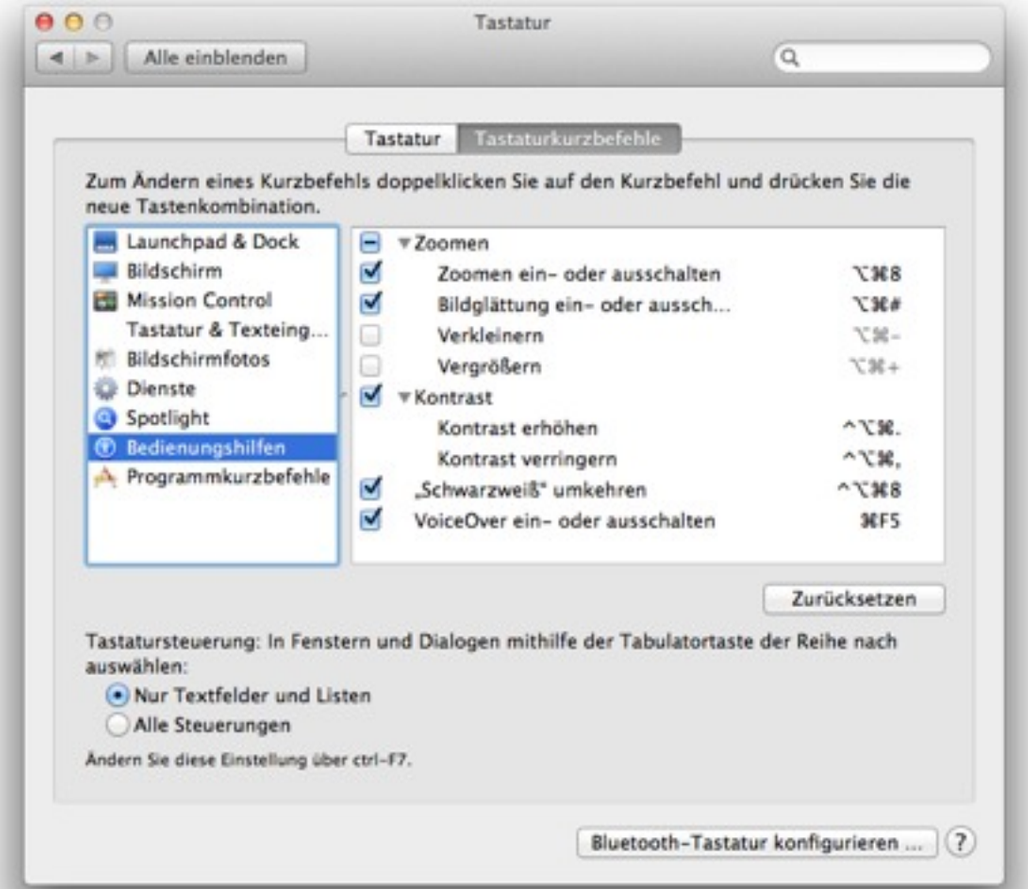
  - „equal opportunity UI“: blurs distinction between input and output



	A	B	C	D
1				
2	Summand 1	1	2	1
3	Summand 2	2	2	2
4	Summand 3	3	3	3
5	Total sum	6	7	6

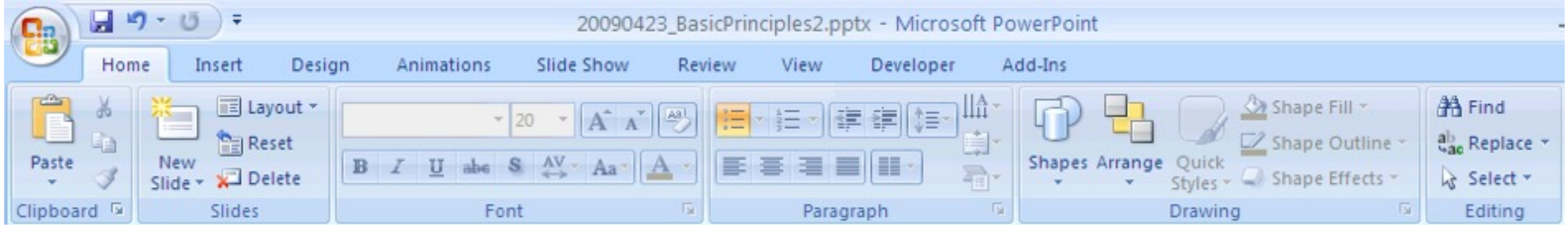
Principles of Flexibility (5 / 6)

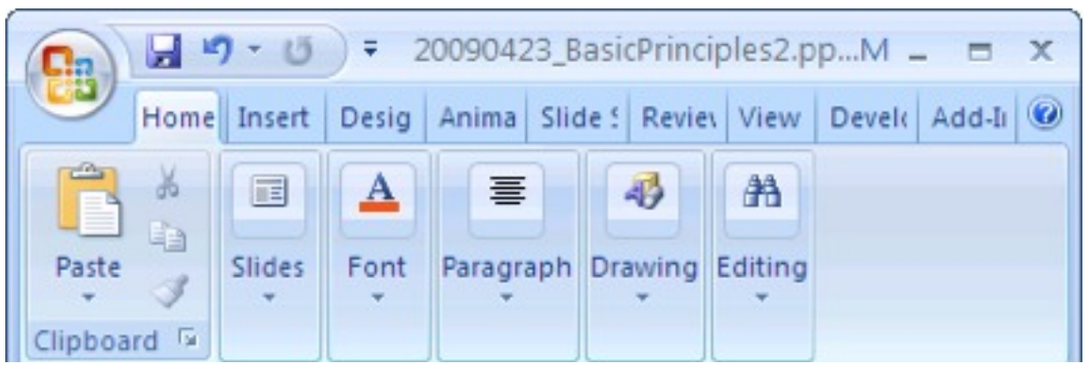
- **Customizability**
  - modifiability of the user interface by the user (adaptability) or system (adaptivity)
  - **adaptability**: users ability to adjust the form of input and output
  - adaptivity: automatic customization of the user interface by the system

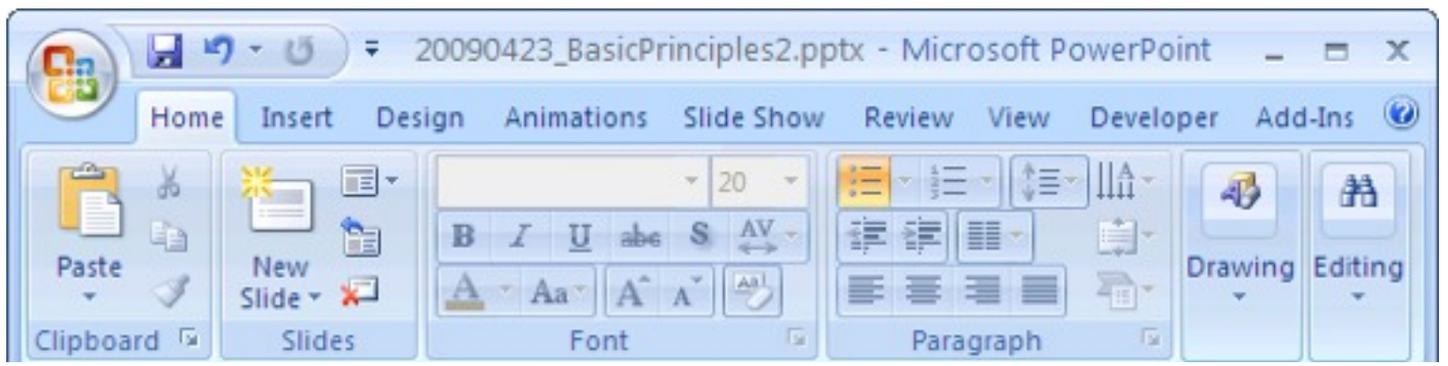


Principles of Flexibility (6 / 6)

- **Customizability**
  - modifiability of the user interface by the user (adaptability) or system (adaptivity)
  - adaptability: users ability to adjust the form of input and output
  - **adaptivity**: automatic customization of the user interface by the system







Principles of Robustness (1 / 2)

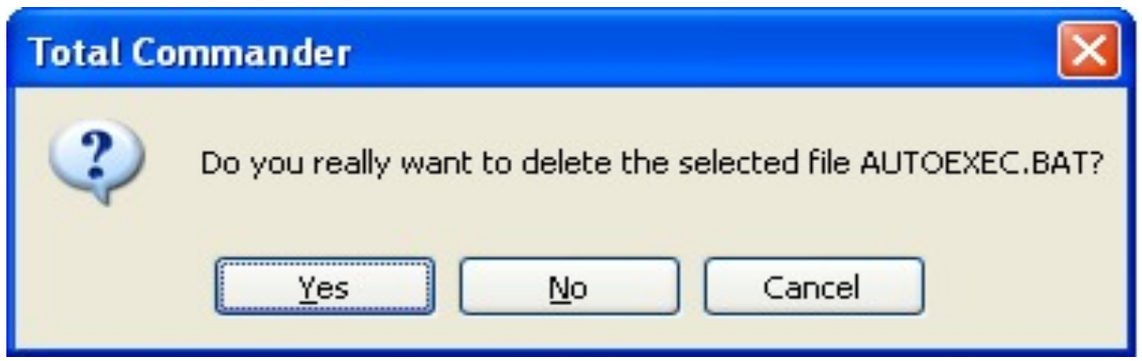
- → Level of support provided to the user in determining successful achievement and assessment of goal-directed behavior

- **Observability**

  – ability of the user to evaluate the internal state of the system from its perceivable representation

- **Recoverability**

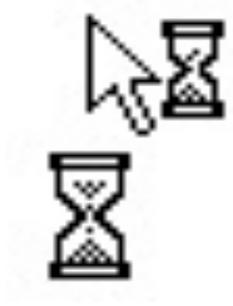
  – ability of the user to correct a recognized error
  – reachability (states): forward (redo) / backward (undo) recovery
  – commensurate effort (more effort / steps for deleting a file than for moving it)

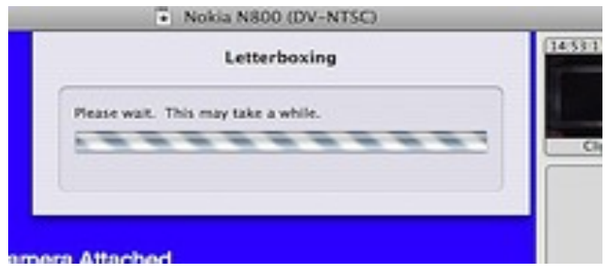


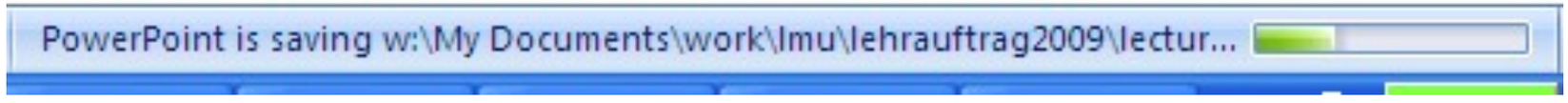
Principles of Robustness (2 / 2)

- **Task conformance**
  - degree to which system services support all of the user's tasks
  - task completeness; task adequacy

- **Responsiveness**
  - how the user perceives the rate of communication with the system
  - preferred: short durations and instantaneous responses (< 100ms)
  - stability and indication of response time







Letterboxing: Please wait. This may take a while.

Basic HCI Principles

- Users and Developers
- 3 Usability Principles by Dix et al.
- **3 Usability Principles by Shneiderman**
- Human Error
- Background: The Psychology of Everyday Action
- Constraints, Mappings, Affordances

Principles for User Interface design

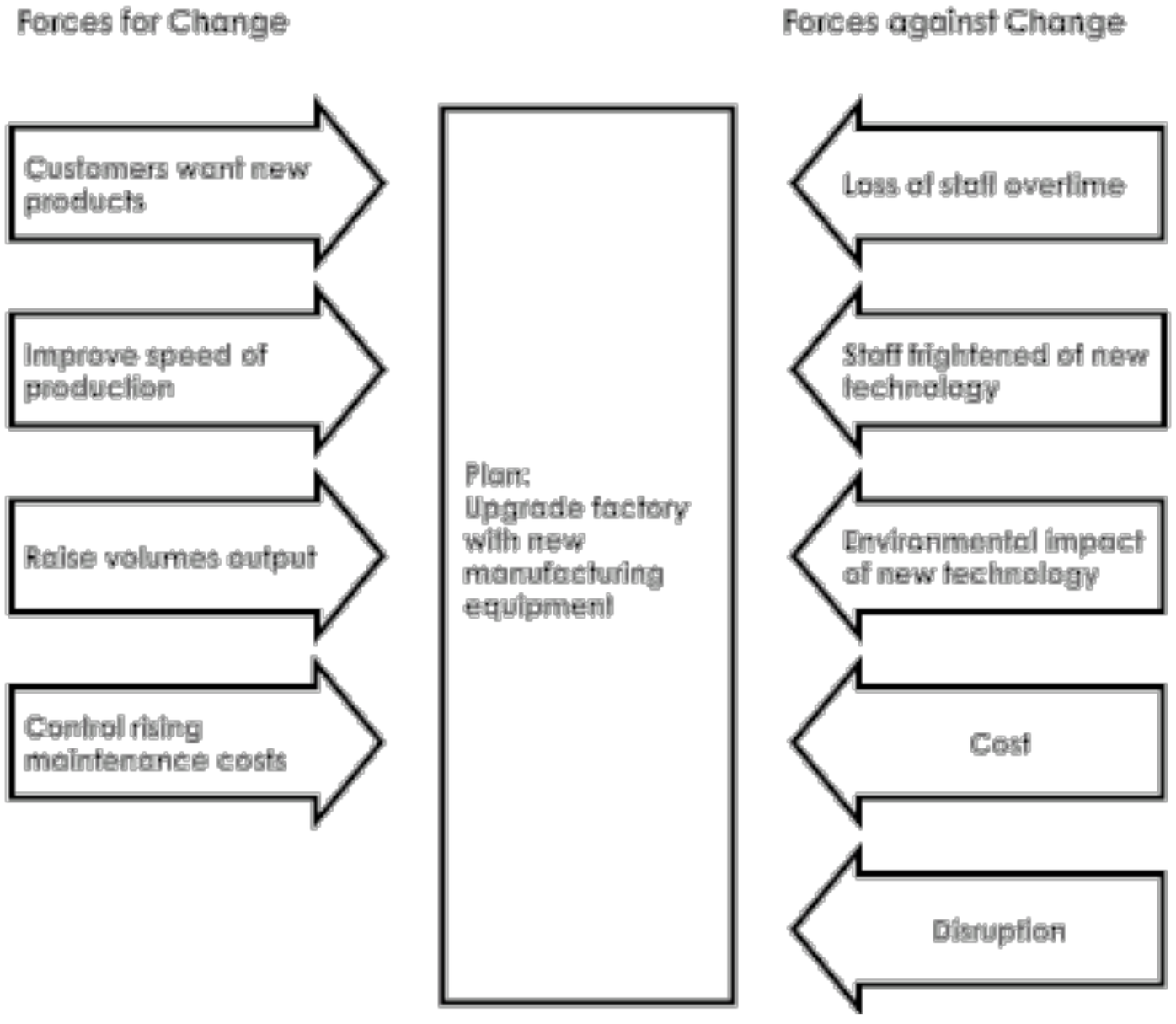
- Implementation and technology-independent principles
  - Provide a rough guideline for design
  - To be supplemented by more detailed analyses (see later)
- Ben Shneiderman's list of principles
  - **Principle 1 : Recognize User Diversity**
  - Principle 2 : Follow the 8 Golden Rules
  - Principle 3 : Prevent Errors

Principle 1: Recognize User Diversity

- Obvious and simple (?) - nevertheless in reality extremely difficult
- To be done before the design
- Basic concepts to structure the problem
  - Usage profiles
    - Different types of users
    - Different types of usage scenarios
    - Dependent on the situation of the user
  - Task profiles
    - What is the goal of the user?
    - How does the user want to achieve the goal?

Usage Profiles – Stakeholders

- Possible stakeholders
  - Shareholders
  - Government
  - Senior executives
  - Your coworkers
  - Suppliers
  - The press
  - Interest groups
  - Customers
  - Analysts
  - The public
  - The community
  - Your family



Force-field Analysis

http://www.mindtools.com

Usage Profiles – Approach

- Identify stakeholders
  - Brainstorming
  - Review past projects
  - Interviews
- Categorise stakeholders
  - Amount of interest
  - Amount of influence
  - Positive / negative attitude
  - Reasons for attitude
- Draw a force-field analysis and keep it in mind throughout the project

Power vs. interest grid

http://www.mindtools.com

Usage Profiles – More than the People

- "Know thy user" (Wilfred J. Hansen, User Engineering Principles for Interactive Systems, 1971)
- Starting point for design: what is the background of the user?
  - Different people have different requirements for their interaction
- Complex multi-dimensional classification problem!
- Issues to be taken into account
  - Goals, motivation, personality
  - Education, cultural background, training
  - Age, gender, physical abilities, …
  - Multiple user communities, various combinations of background
- Well-known and frequently used classification
  - Novice users
  - Knowledgeable intermittent users
  - Expert frequent users

Task Profiles

- The goal: find out what the user is trying to do!
  - Needs of users, goals and resulting tasks
- Supported tasks should be determined before the design starts
  - Determine granularity of atomic tasks: flexibility vs. ease of use
- Functionality should only be added if identified to help solving tasks
  - Temptation: add unneeded functionality only because it is “cheap” to achieve!
- Frequency of actions (relative to user profiles) leads to design choices
  - The more frequent an action, the easier its invocation
  - Example:
    - very frequent actions invoked by special keys (e.g. DEL)
    - intermediately frequent actions invoked by keyboard shortcut, special button, …
    - infrequent actions invoked through menu selections, form fillings, …

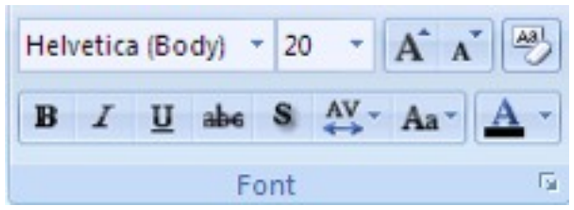
Hypothetical Frequency of Tasks (Example: a travel booking system)

Task Position	Group reservation	Change of itinerary	Booking child care	Comparing sales agent performance
Sales agent	0.2	0.1	0.1	0
Manager	0	0	0	0.3
Family	0.05	0.05	0.3	0
Business traveler	0.01	0.2	0.01	0

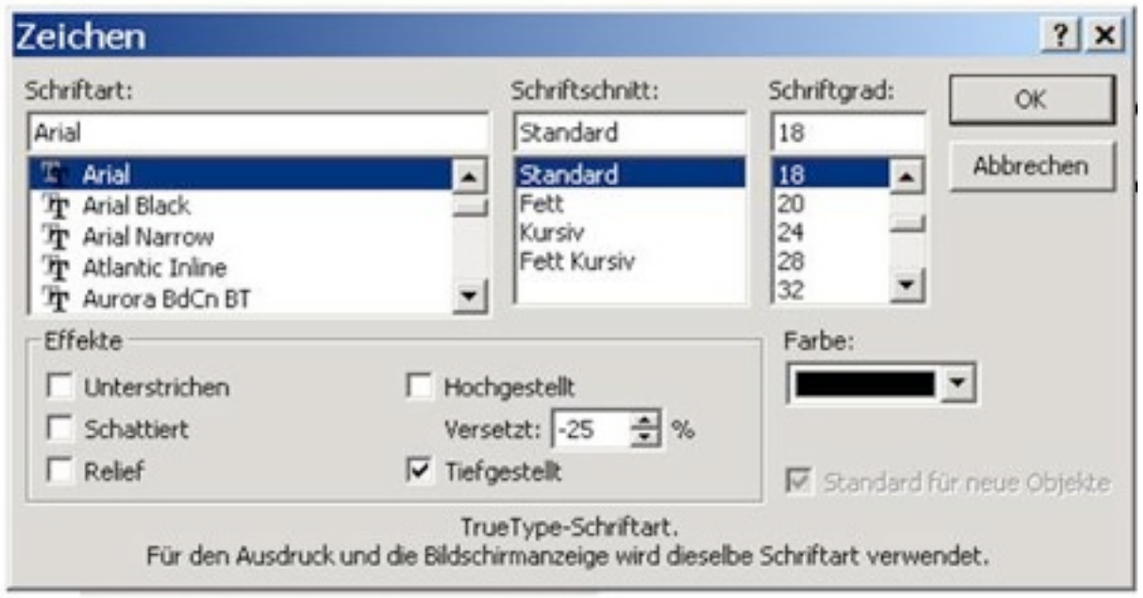
Task Frequency – Examples

Word 2003

Powerpoint 2007

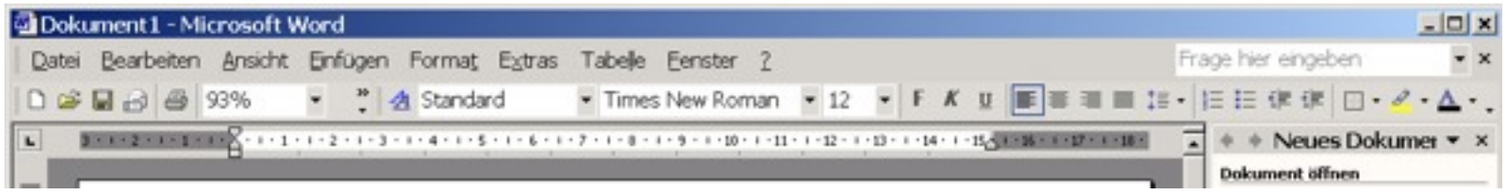


Word 2007

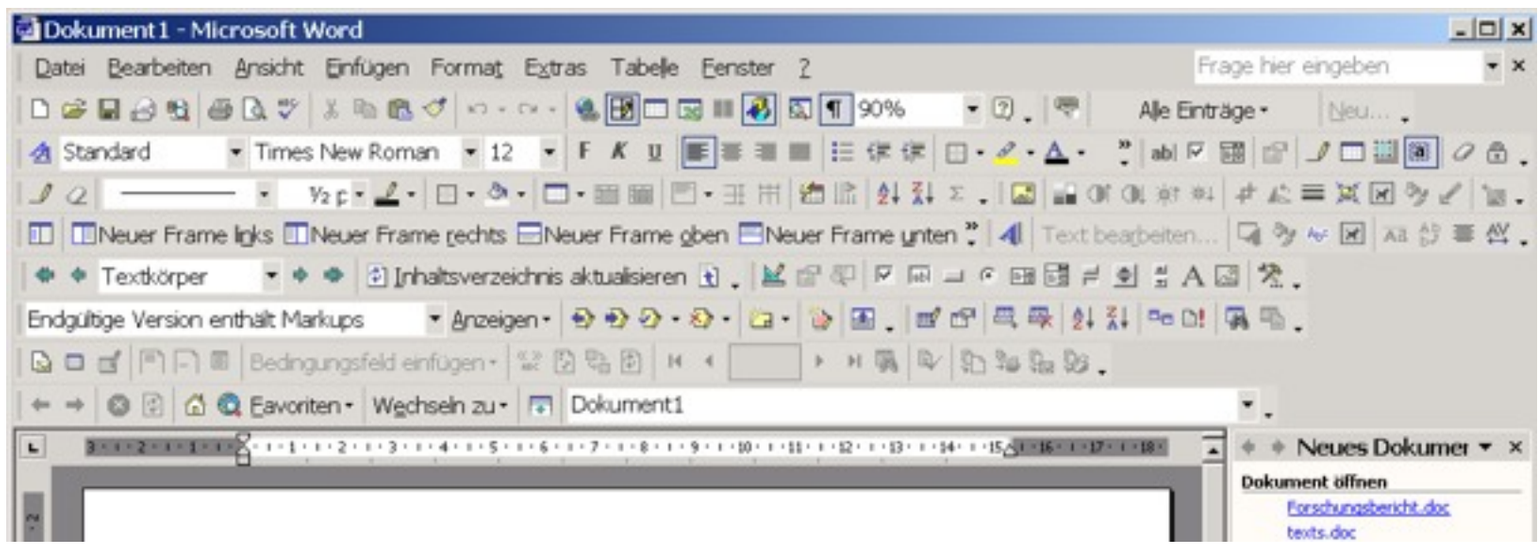


- Bold format is available in the toolbar
- Subscript requires menu and dialog
- Assumption for the standard UI is that user needs more often bold than subscript
- For users with different needs customization is possible

Task Frequency: Trade-off between Quick Access and Over-crowed Interface



- Example toolbar
  - More tasks directly available in the toolbar make it quicker to do these tasks
  - Increasing the number of options in the toolbar increase the time needed to locate them
  - Screen area that is used

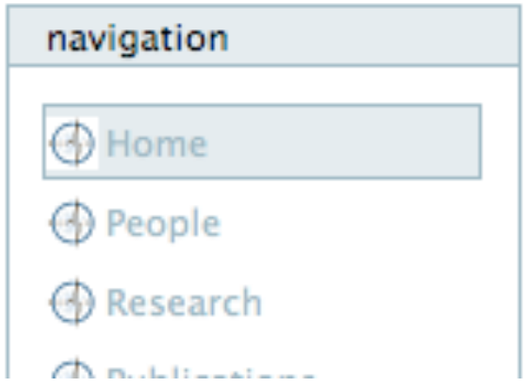


Principles for User Interface design

- Implementation and technology-independent principles
  - Provide a rough guideline for design
  - To be supplemented by more detailed analyses (see later)
- Ben Shneiderman's list of principles
  - Principle 1 : Recognize User Diversity
  - **Principle 2 : Follow the 8 Golden Rules**
  - Principle 3 : Prevent Errors

Rule 1: Consistency

- Many forms of consistency:
  - Consistent sequences of actions in similar situations
  - Identical terminology used in prompts, menus, help screens
  - Consistent color, capitalization, layout, fonts etc.
- Bad example: WWW
  - No real guidelines and no authority
    - How are links represented?
    - Where is the navigation?
  - Styles and “fashion” change quickly…



Consistency: Levels

- Consistency levels
  - lexical
  - syntactic
  - semantic

Consistent
Delete/insert character
Delete/insert word
Delete/insert line
Delete/insert paragraph

- **Inconsistent** – variant 1
  Delete/insert character
  Delete/insert word
  Remove/insert line
  Delete/insert paragraph

- **Inconsistent** - variant 2
  Take-away/insert character
  Delete/add word
  remove/put-in line
  eliminate/create paragraph

- **Inconsistent** - variant 3
  Character deletion/insertion
  Delete/insert word
  Line deletion/insertion
  Delete/insert paragraph

Lexical / Syntactic Consistency

- 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 in the same way in all phrases
  - Character delete key is always the same

Semantic Consistency

- Global commands that are 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”

Consistency: Capture through Grammars

- Task-Action-Grammar (TAG), Reisner 1981
  - 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 one word forward: CTRL-W
  - Move cursor one word backward: ALT-W
  - Move cursor one paragraph forward: CTRL-P
  - Move cursor one paragraph backward: ALT-P

Inconsistencies

- Dragging file operations?
  - folder on same disk vs. folder on different disk
  - file to trash can vs. disk to trash can
- Fitts' Law suggests bigger buttons for more often used operations

- Sometimes inconsistency is wanted
  - E.g. Getting attention for a dangerous operation
  - Consistency on semantic level may cause inconsistency on syntactic level
  - Example:
    - Confirmation of operation is default option
    - Confirmation of reformat command?

Rule 2: Shortcuts

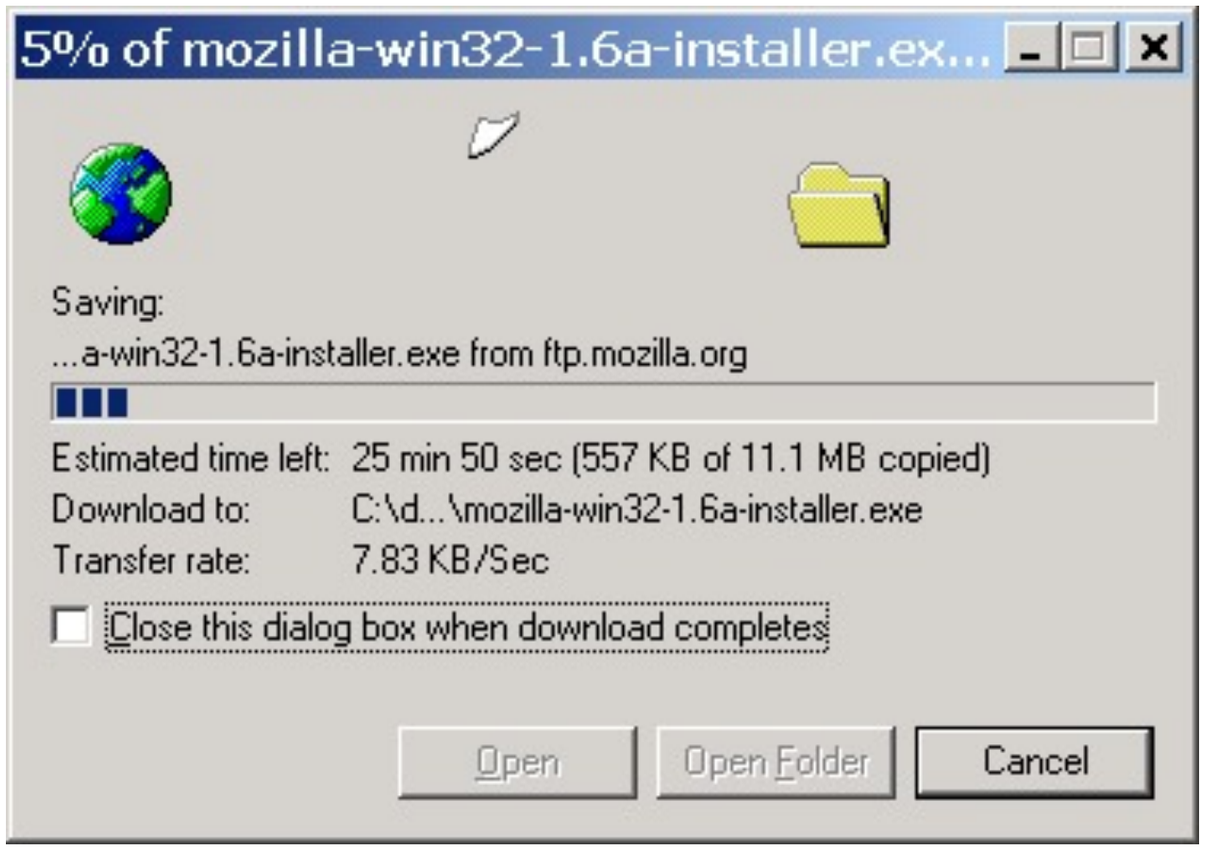
- Enable shortcuts: Improves speed for experienced users
- Shortcuts on different levels
  - Access to single commands, e.g. keyboard **shortcuts** (CTRL+S) or toolbar
  - Customizing of commands and environments, e.g. printer **presets** (duplex, A4, …)
  - Reusing actions performed, e.g. **history** in command lines, macro functionality
- Shortcuts to single commands are related to consistency
  - CTRL+X, CTRL+C, CTRL+V in Microsoft & Apple applications for cut, copy and paste
  - However CTRL+S (saving a document) is only implemented in some applications…
  - Apple applications are more consistent in shortcuts (e.g. CTRL-S) due to early guidelines/toolkits for developers

Rule 3: Feedback

- For any action performed the user should have appropriate and informative feedback
- For frequent actions it should be modest, peripheral
- For infrequent actions it should be more substantial

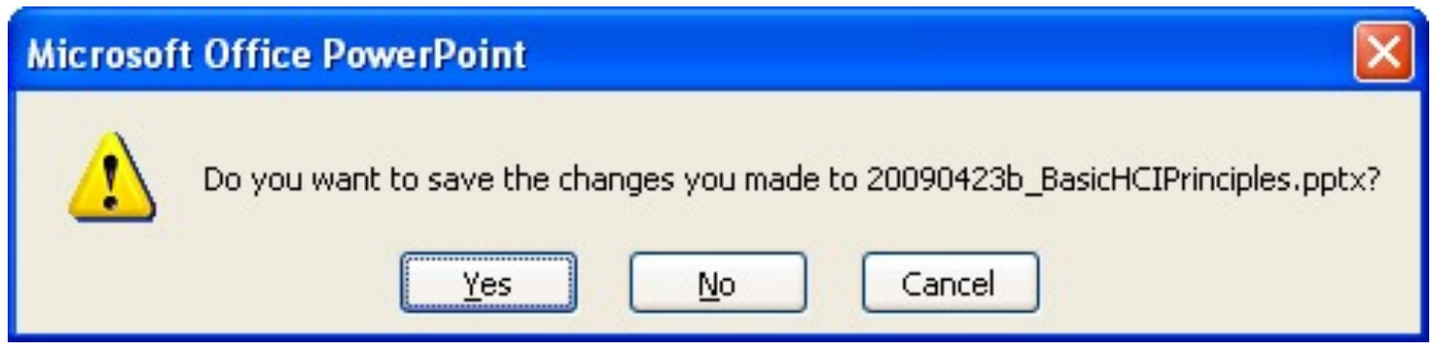
Rule 4: Closure

- Sequences of actions should have a beginning, middle, and end.
  - Satisfaction of accomplishment = relief
- On different levels –
  - E.g. in the large: Web shop - it should be clear when I am in the shop, and when I have successfully checked out
  - E.g. in the small: a progress bar



Rule 5: Prevent Errors

- Create UIs that make it hard to make errors
  - Examples:
    - Menus instead of commands
    - Options instead of alphanumeric field (only certain values allowed)
- Detect errors or possible errors
  - Examples
    - Leaving an editor without saving
    - Writing to a file that already exists



- Provide safety for the user
- Different options for handling:
  - Involve the user (current practice)
  - Prevent the error or its consequences on system level (e.g. create backups/versions when a file is overwritten, OSX 10.8)

Rule 6: Easy Reversal of Actions

- As a basic rule – all actions should be reversible
  - Relieves anxiety of users, encourages exploration of unfamiliar options
- Providing UNDO functions (possibly with infinite depth)
- Allow undo of groups of actions
- Undo is not trivial if user is not working sequentially
  - E.g. write a text, copy it into the clipboard, undo the writing: ==> the text is still in the clipboard!
- Reversal of action becomes a usage concept
  - Browser back-button is used for navigation (for the user a conceptual reversal of action)
  - Formatting of documents – e.g. “lets see how this looks, … don’t like it, … go back to the old state”

Rule 7: Feeling in Control

- Users (in particular experienced) like to feel to be in control of the system
- Gaines, 1981:
  - User should initiate actions (initiator instead of responder)
  - Avoid non-causality
- The system should be predictable
  - No surprising system actions, no tedious but unavoidable sequences of data entries, no unexpected silence or waiting state
  - Otherwise anxiety and dissatisfaction arise
- Note: some current developments are in contrast, e.g.:
  - Proactive computing
  - Intelligent agents
- General trade-off between transparency and intelligence of system

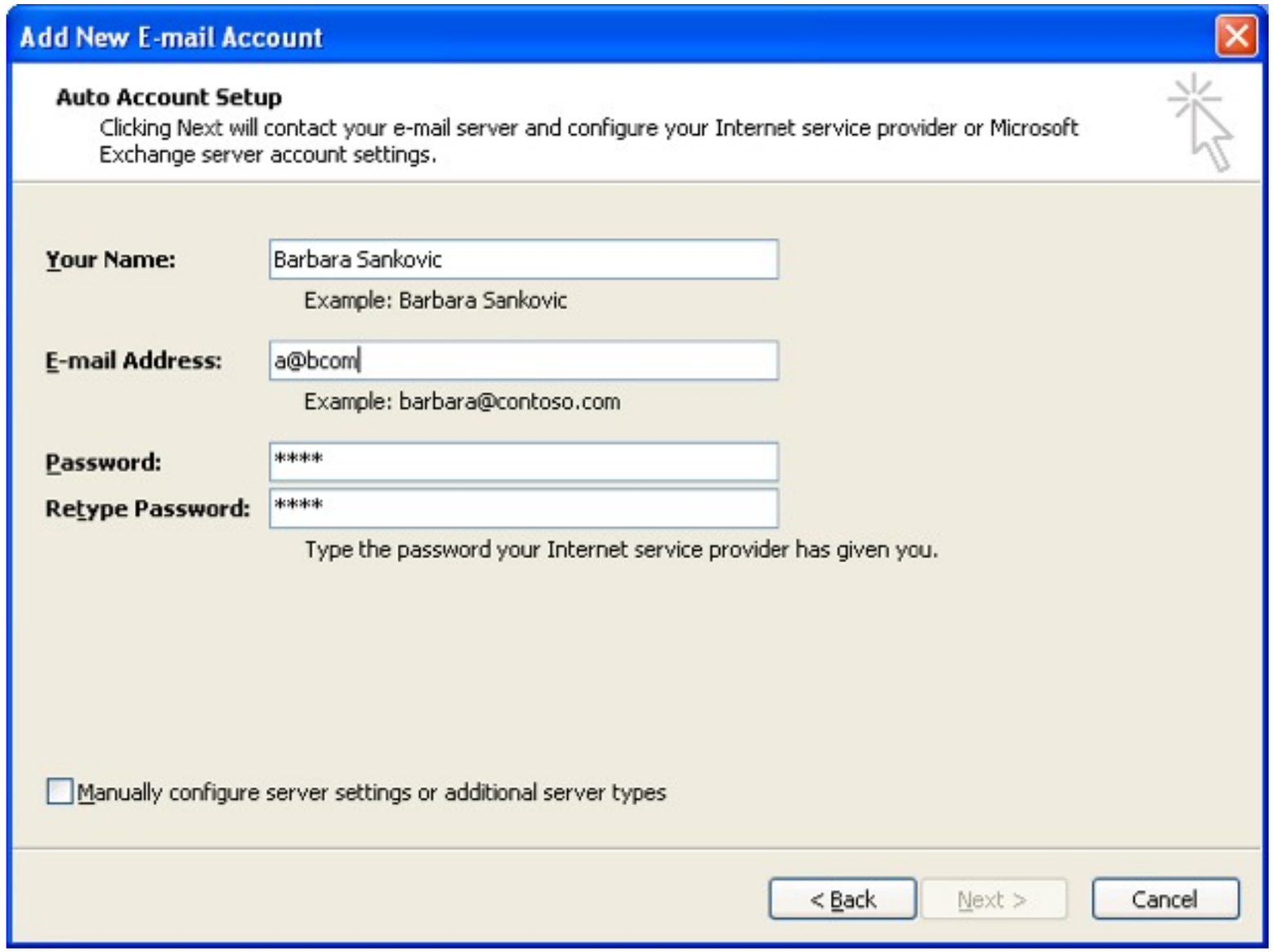
Rule 8: Reduce Short-term Memory Load

- The system should remember, not the user
  - George A. Miller, 1956: The magical number Seven, Plus or Minus Two
  - Humans can recall 7 +/- 2 chunks of information for a short time
- Interface designs have to be simple to comply with human memory
- Examples that create problems
  - Multi-page forms where the user has to know at form N what she filled in in form N-1
  - Abbreviations introduced in one step and used in the following (e.g. user selects a destination – as the name of a city – and the system does the following steps by showing the airport code)
- Helpful:
  - Keep dialogues compact (avoid splitting of pages)
  - Use memory aids (visual or audio) for mnemonics
- Apply the rule with care!
  - Sometimes complex menu structures are unavoidable
  - With sufficient training and support, also cryptic mnemonics are acceptable for frequent users

Summary – 8 Golden Rules

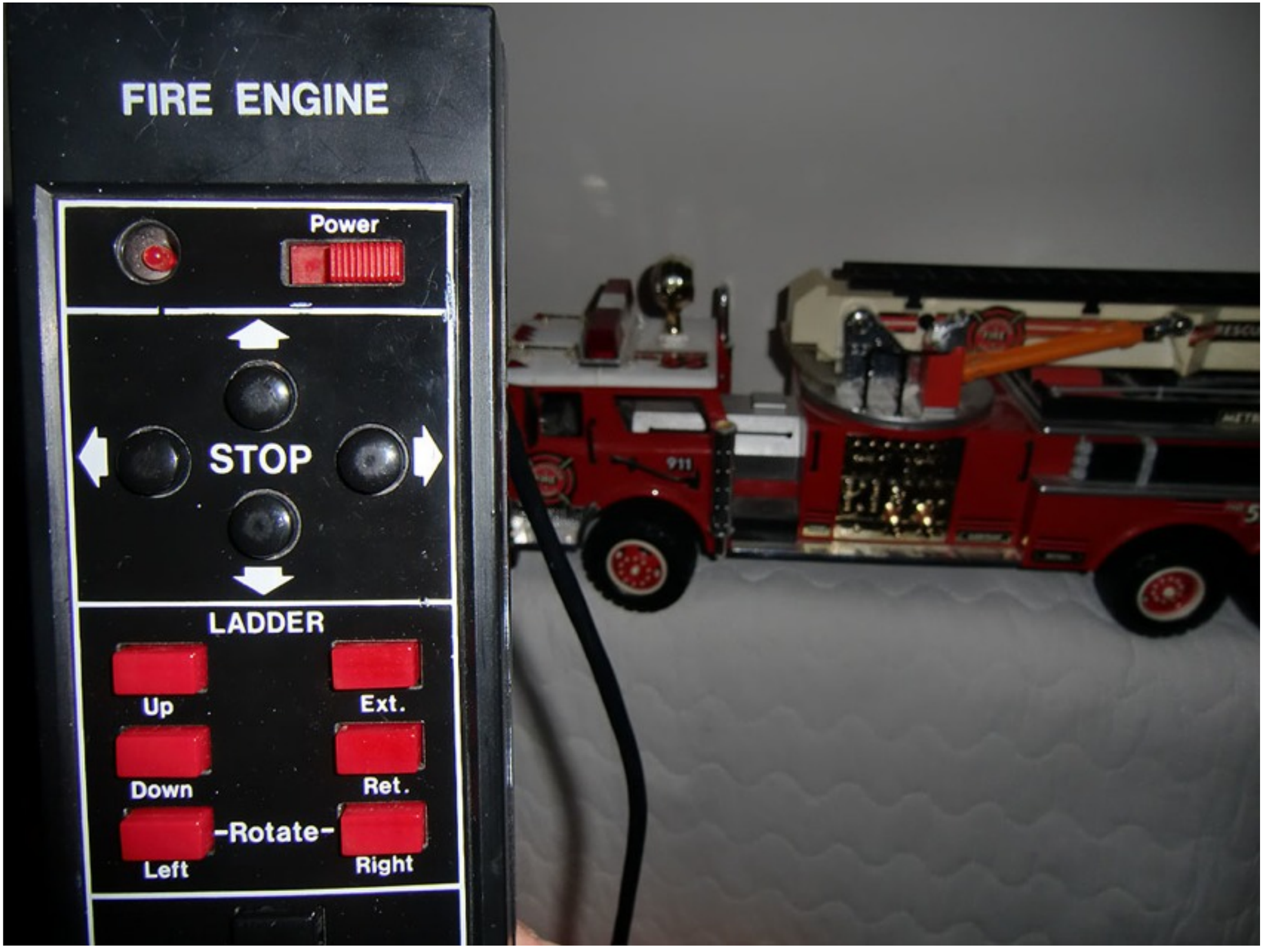
MS Outlook 2007

- Consistency
- Shortcuts
- Feedback
- Closure
- Prevent Errors
- Reversal
- Control
- Memory Load



1) Does not show there is a (potential) error in the email address – just greys out the ‘Next’ button.
2) When passwords do not match, it allows ‘Next’ but gives a detailed error message.

A recent example: discussion!



Principles for User Interface design

- Implementation and technology-independent principles
  - Provide a rough guideline for design
  - To be supplemented by more detailed analyses (see later)
- Ben Shneiderman's list of principles
  - Principle 1 : Recognize User Diversity
  - Principle 2 : Follow the 8 Golden Rules
  - **Principle 3 : Prevent Errors**

Prevent Errors - Classical Techniques

- (Note: golden rule 5 discusses the same topic on higher level…)
- A few classical “tricks” to prevent errors (Source: Shneiderman)
- Correct matching pairs
  - Examples: { } in program text, <B>bold</B> in HTML
  - Prevention: insert both brackets in one action; or remind of missing bracket
- Complete sequences
  - Assistance to complete a sequence of actions to perform a task
    - For advanced users: planning and editing the sequence
  - Examples: log-on sequences, wizards, scripts
- Command correction
  - Aim: Trying to prevent users entering incorrect commands
    - Examples: file completion on Unix / helpful error messages / menus instead of commands

What is an “error” after all?

Basic HCI Principles

- Users and Developers
- 3 Usability Principles by Dix et al.
- 3 Usability Principles by Shneiderman
- **Human Error**
- Background: The Psychology of Everyday Action
- Constraints, Mappings, Affordances

Human Errors, 1986

- Space Shuttle Challenger accident
- NASA overrode safety warnings from engineers about the seals of the solid rocket boosters. Engineers warned that the O-ring seals failed repeated tests under the cold conditions the morning of the Challenger launch, but NASA ignored the red flags and went ahead anyway. What seemed like a small part eventually turned catastrophic.

- Chernobyl nuclear reactor accident
- At Chernobyl, a group of scientists intentionally deactivated several safety systems in order to test a cooling system at reactor 4 which led to the worst nuclear accident in history.

Human Error as the Ultimate Explanation

Deadly crash on German monorail

Twenty-three people died and 10 were injured when an elevated magnetic train ploughed into a maintenance vehicle in north-western Germany.


Rescuers had to use ladders and cranes to reach the train

The train, which floats on a monorail via a magnetic levitation system called maglev, was going at nearly 200km/h (120 mph) when it crashed near Lathen.

Damaged carriages were left balancing on track 5m (16ft) in the air, hampering rescue efforts.

'Human error'

The maintenance vehicle hit by the train had two crew members.

A spokesman for IABG, the company which operates the train, said the accident had been caused by human error, rather than a technical fault.

http://news.bbc.co.uk/1/hi/world/europe/5370564.stm

Bei der Analyse der Unfallursachen stützt sich der Bericht laut «Nordwest-Zeitung» auf zwei Gutachten zu dem Unglück: Nach Ansicht der Gutachter verstieß der Fahrdienstleiter gegen die Betriebsvorschriften, weil er die elektronische Streckensperre nicht setzte. Als weitere Ursache wird die Missachtung des Vier-Augen-Prinzips im Leitstand der Teststrecke genannt.

http://www.netzeitung.de/politik/deutschland/720674.html

Human Errors and Management

TAIPEI TIMES

Published on TaipeiTimes
http://www.taipeitimes.com/News/taiwan/archives/2003/10/18/2003072381

Fighter pilots find panic button at last

MISTAKE MANAGEMENT: Two crashes blamed on human error have prompted the developers of the IDF to remind the air force about a built-in emergency function

By Brian Hsu
STAFF REPORTER
Saturday, Oct 18, 2003, Page 4

Although Taiwan's Indigenous Defense Fighter (IDF) has an emergency function that minimizes the chance of a plane crash due to human error, pilots have only now found out about it.

The previous two accidents involving IDFs this year were caused by human error, defense sources said yesterday.

"The crash was also caused by the negative G-force which the flight instructor created

…In an attempt to prevent similar accidents in future, the air force has asked the AIDC to help teach pilots how to use the fighter's emergency function.

Human Errors

The Official Google Blog

Insights from Googlers into our products, technology, and the Google culture.

"This site may harm your computer" on every search result?!?!

1/31/2009 09:02:00 AM

What happened? Very simply, human error. Google flags search results with the message "This site may harm your computer" if the site is known to install malicious software in the background or otherwise surreptitiously. We do this to protect our users against visiting sites that could harm their computers. We maintain a list of such sites through both manual and automated methods. We work with a non-profit called StopBadware.org to come up with criteria for maintaining this list, and to provide simple processes for webmasters to remove their site from the list.

We periodically update that list and released one such update to the site this morning. Unfortunately (and here's the human error), the URL of '/' was mistakenly checked in as a value to the file and '/' expands to all URLs. Fortunately, our on-call site reliability team

Human Error and Commercial Success

- http://pittsburghlive.com/x/tribune-review/business/s_385507.html

PITTSBURGH TRIBUNE-REVIEW

Back to headlines

TT Larger Text Tt Smaller Text

Barring human error made area firm a health leader

By Rick Stouffer
TRIBUNE-REVIEW
Wednesday, October 19, 2005

More than 30 years ago, bar codes began showing up on the bottoms, backs or sides of everything from blocks of cheese to 2-by-4s.

Medicine, however, was a late arrival to tracking equipment and medications using bar code technology. In the early 1990s, it was a Pittsburgh-based start-up, Automated Healthcare, that jump-started the use of the vertical black and white lines for tracking medicine in hospitals.

"It really was quite amazing that we were bar coding ketchup, but not bar coding things that could kill you if an error was made," said Sean McDonald, who founded Automated Healthcare in 1990, sold it to drug distribution giant McKesson in 1996 for $65 million, then stayed for five years to continue running the company. Today, the company is known as McKesson Automation.

McKesso

Founded:
Healthca
Sean Mc
student a
Universit

Acquired
Healthca
by drug d
McKesso
million.

Headqua

Presiden
Souerwin

Trying to prevent error

- Redundancy increases safety
  - E.g. labels and physical constraints
- Constraints can only work at their own level
- But: things can go wrong elsewhere

Defektes Nakosegerät

Unfallopfer mit Lachgas beatmet -Tödliche Klinik-Panne

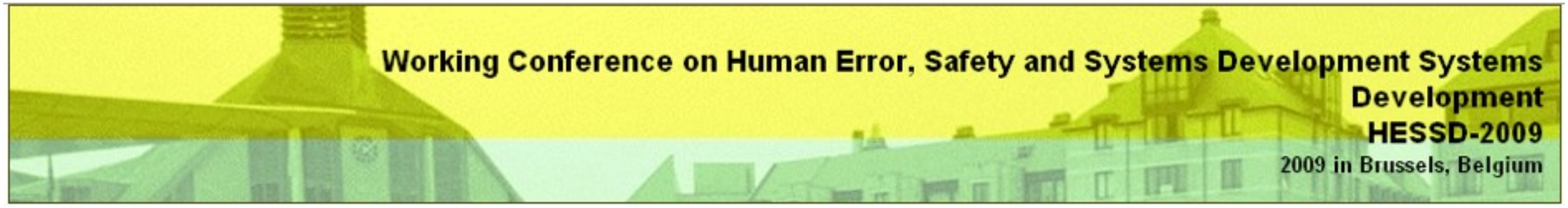
Dieser Artikel stellt eine am **25.03.04** um **13:59** veröffentlichte Nachricht dar.

AKTUELLE NACHRICHTEN

Traunstein (rpo). Lachgas statt Sauerstoff - in einer bayerischen Klinik musste diese Verwechslung ein 19-Jähriger mit dem Leben bezahlen.

Durch ein falsch zusammengebautes Narkosegerät ist in einem bayerischen Krankenhaus ein Patient ums Leben gekommen. Der 19-Jährige war nach einem Verkehrsunfall in der Notaufnahme der Klinik in Trostbergan statt mit Sauerstoff mit Lachgas beatmet worden, wie die Staatsanwaltschaft Traunstein am Donnerstag sagte. Ermittelt werde gegen einen Mitarbeiter der Herstellerfirma, der das Gerät zuvor repariert hatte. Dabei seien die Anschlüsse für Lachgas und Sauerstoff vertauscht worden.

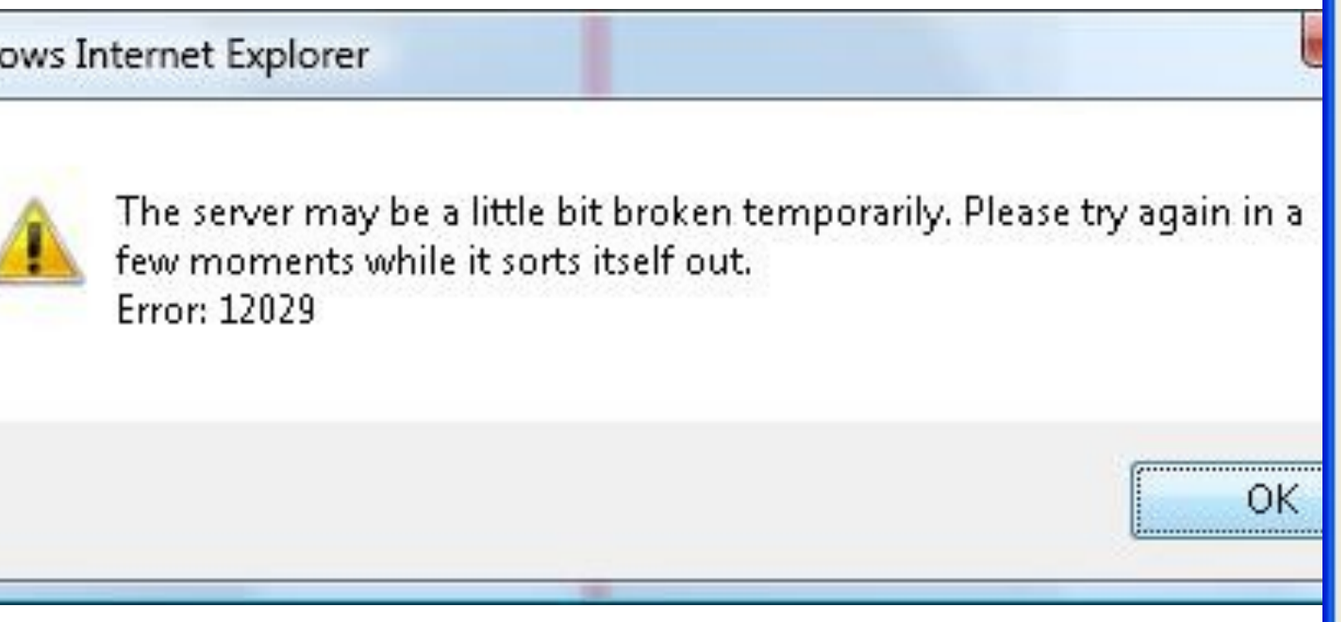
Tackling Errors



- Our intention is to focus the working conference upon techniques that can be easily integrated into existing systems engineering practices. With this in mind, we hope to address a number of different themes:
  - techniques for incident and accident analysis;
  - empirical studies of operator
  - behaviour in safety-critical systems
  - observational studies of safety-critical systems
  - risk assessment techniques for interactive systems
  - safety-related interface design
  - development and testing

About (Human) Errors…

- “If an error is possible, someone will make it” (Norman)
- Human errors may be a starting point to look for design problems
- Design implications
  - Assume all possible errors will be made
  - Minimize the chance to make errors (constraints)
  - Minimize the effect that errors have (is difficult!)
  - Include mechanism to detect errors
  - Make actions reversible



Understanding Errors

- Errors are routinely made
  - Communication and language is used between people to clarify – more often than one imagines
  - Common understanding of goals and intentions between people helps to overcome errors
- Two fundamental categories
  - **Mistakes**
    - overgeneralization
    - wrong conclusions
    - wrong goal
  - **Slips**
    - Result of “automatic” behaviour
    - Appropriate goal but performance/action is wrong

Norman, Chapter 5

Understanding the Types of Slips Users Make

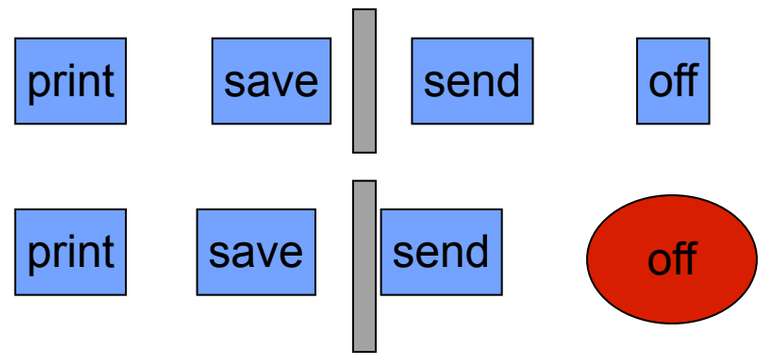
- Capture errors
  - Two actions with common start point, the more familiar one captures the unusual (driving to work on Saturday instead of the supermarket)
- Description errors
  - Performing an action that is close to the action that one wanted to perform (putting the cutlery in the bin instead of the sink)
- Data driven errors
  - Using data that is visible in a particular moment instead of the data that is well-known (calling the room number you see instead of the phone number you know by heart)
- Associate action errors
  - You think of something and that influences your action (e.g. saying come in after picking up the phone)
- Loss-of-Activation error (~ forgetting)
  - In a given environment you decided to do something but when leaving then you forgot what you wanted to do. Going back to the start place helps you remember
- Mode error
  - You forget that you are in a mode that does not allow a certain action or where a action has a different effect

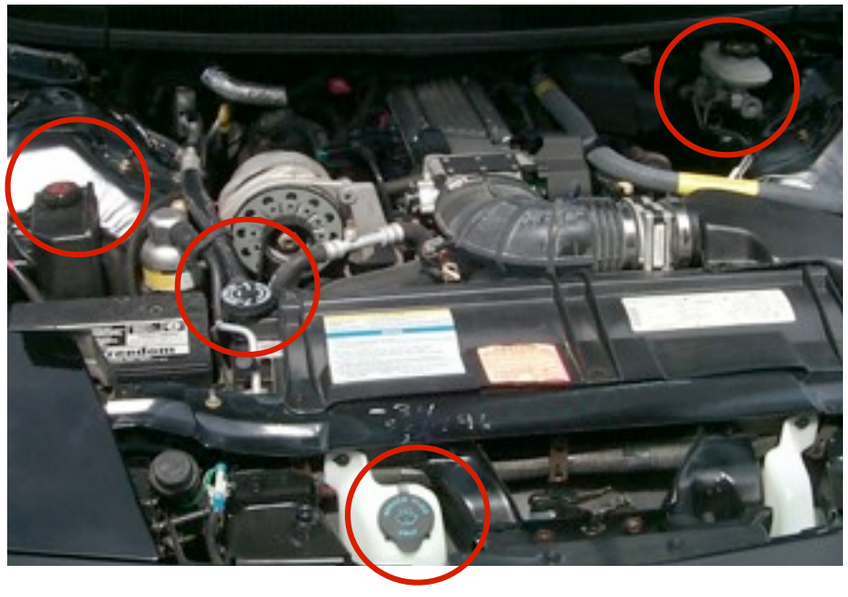
Norman, Chapter 5

Preventing Description Errors

- Related to Gestalt theory

- Example Car
  - Different openings for fluids, e.g. oil, water, break, …
  - Openings differ in
    - Size
    - Position
    - Mechanism to open
    - Color





- Design recommendations
  - Make controls for different actions look different

Preventing Mode Errors

- Why use modes in the first place?
  - User interface trade-off (e.g. number of buttons needed can be reduced, actions within a mode can be speeded up)

- Design recommendations
  - Minimize number of modes
  - Make modes always visible

- Example alarm clock
  - Mode vs. mode free
  - Visualization of mode

Setting time and alarm with mode

Setting time and alarm without mode

Basic HCI Principles

- Users and Developers
- 3 Usability Principles by Dix et al.
- 3 Usability Principles by Shneiderman
- Human Error
- **Background: The Psychology of Everyday Action**
- Constraints, Mappings, Affordances

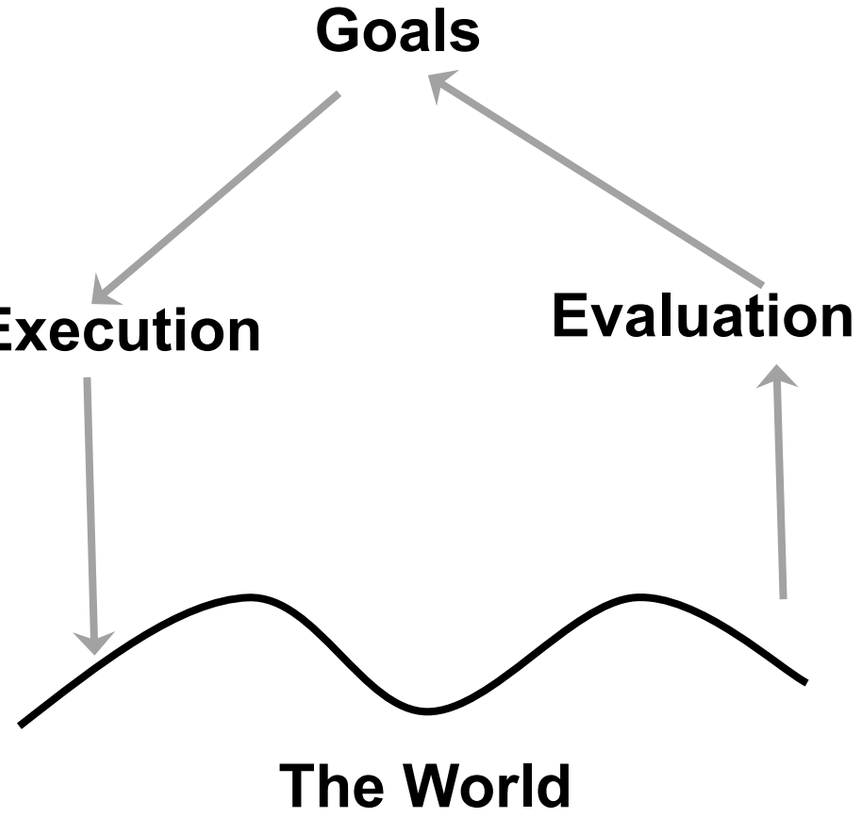
Background: The Psychology of Everyday Action

- 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 needs not be fully correct – it must explain the phenomenon
- People always try to explain actions and results
  - Random coincidence may lead to assumptions about causality

(Norman 2002, Chapter 2)

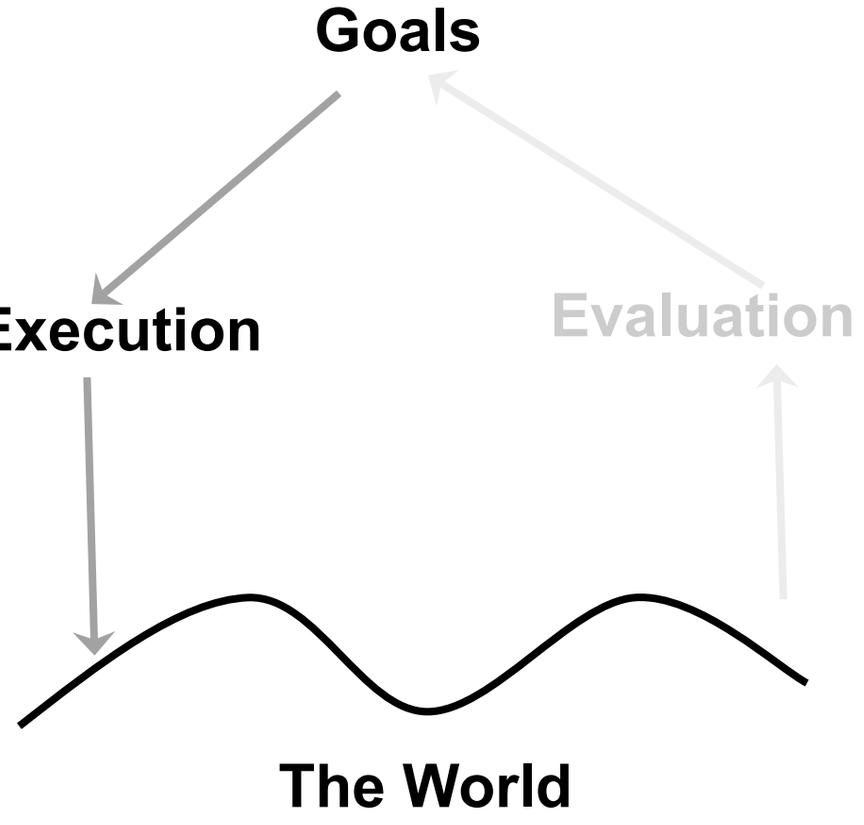
Action Cycle

- The action is goal-directed
  - What do we want to happen?
  - What is the desired state?
- Human action has two major aspects
  - Execution:
    what we do to the world
  - Evaluation:
    compare if what happens is what we want



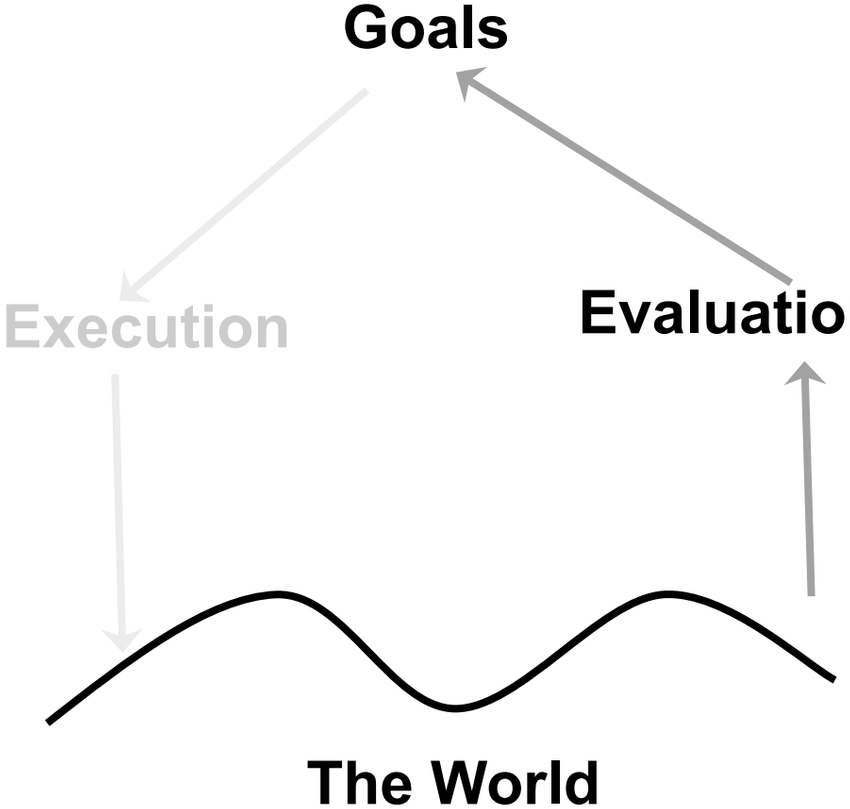
Action Cycle: Stages of Execution

- 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

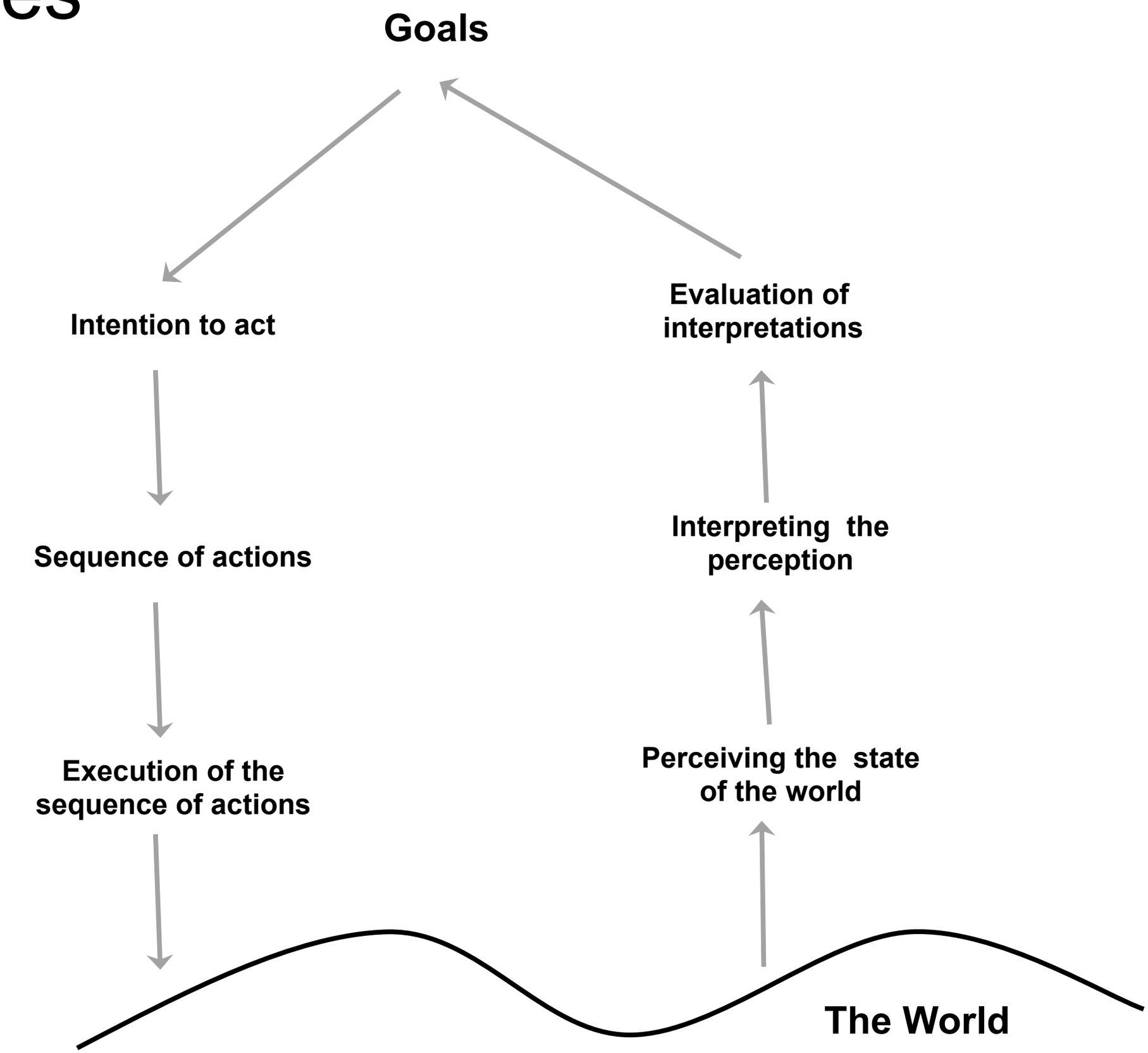


Action Cycle: Stages of Evaluation

- Perceiving the state of the worlds
  - 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

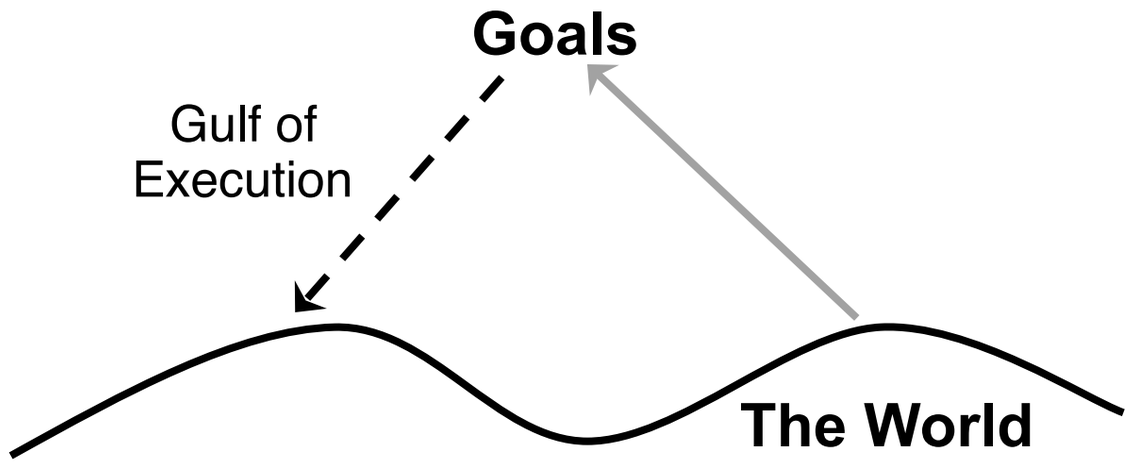


Seven Stages of Action



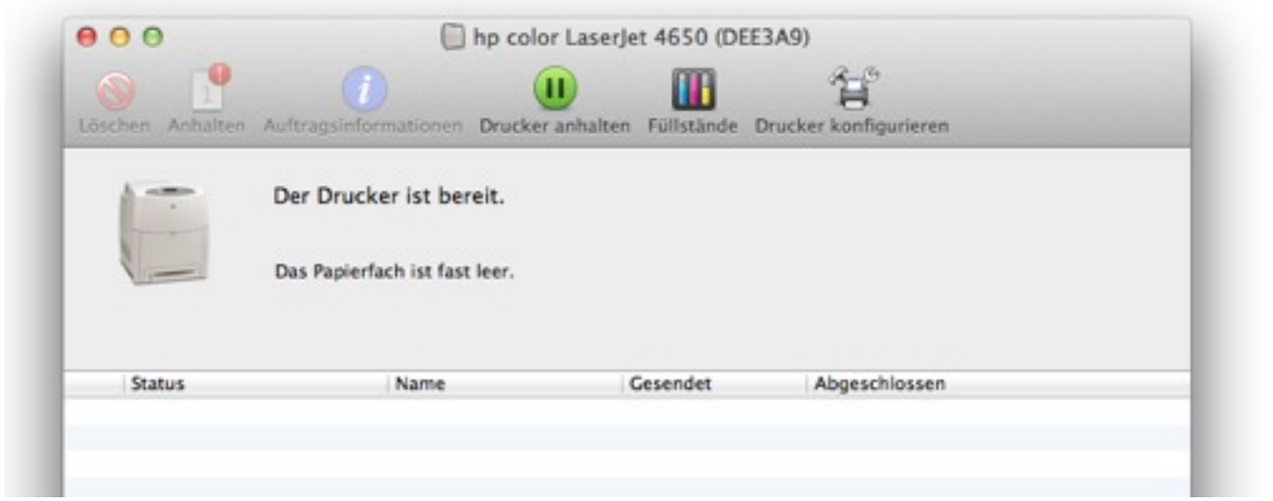
Gulf of Execution

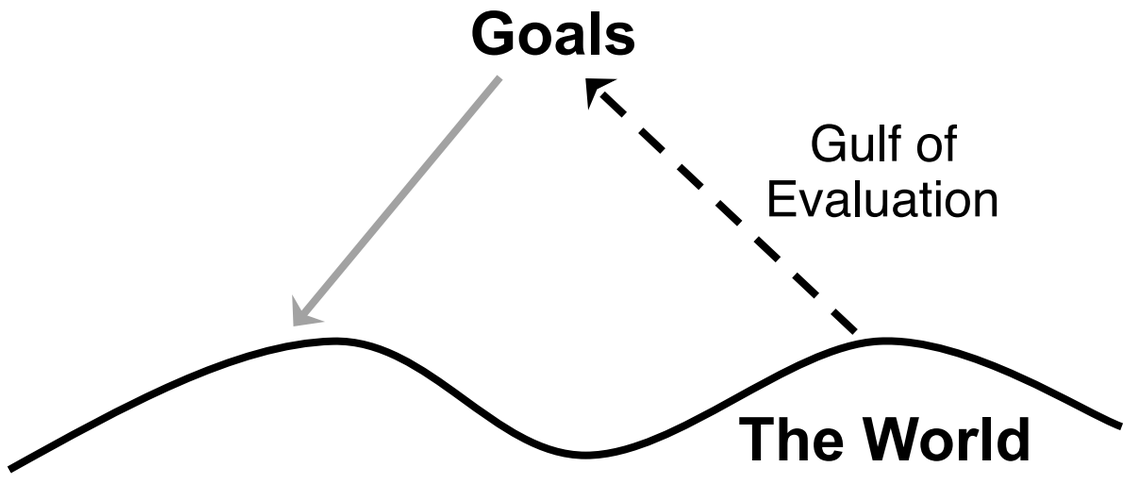
- The difference between the intentions and the allowable actions is the Gulf of Execution
  - How directly can the actions be accomplished?
  - Do the actions that can be taken in the system match the actions intended by the person?
- Example:
  - The user wants a document written on the system in paper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



Gulf of Evaluation

- The Gulf of Evaluation reflects the amount of effort needed to interpret the state of the system and 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 the process observable? Are intermediate steps visible?
- Good design minimizes the Gulf of Evaluation





Evaluation and Design Questions

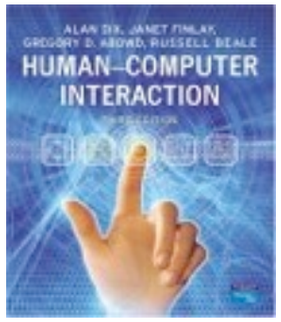
- **Execution**
  - Can the user tell what actions are possible?
  - Does the interface help with mapping from intention to physical movement?
  - Does the device easily support required actions?
- **Evaluation**
  - Can the user tell if the system is in the desired state?
  - Can the user map from the system state to an interpretation?
  - Can the user tell what state the system is in?

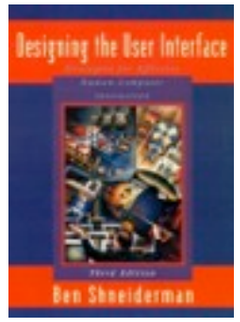
Implications on Design

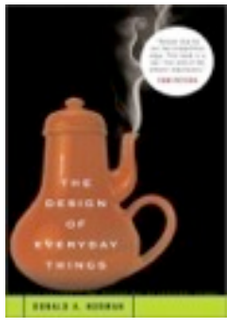
- Principles of good design (Norman)
  - Stage and action alternatives should be always visible
  - Good conceptual model with a consistent system image
  - Interfaces should include good mappings that show the relationship between stages
  - Continuous feedback to the user

- Critical points/failures
  - Inadequate goal formed by the user
  - User does not find the correct interface / interaction object
  - User many not be able to specify / execute the desired action
  - Inappropriate / mismatching feedback

References







- D. A. Norman. The Design of Everyday Things. Basic Books. 2002. ISBN: 978-0465067107
- B. Shneiderman. Designing the User Interface: Strategies for Effective Human-Computer Interaction , 5th Edition. 2009. ISBN: 978-0321537355
- L. Suchman, Plans and Situated Action:- The Problem of Human-Machine Communication. 1987, ISBN 978-0521337397
- Alan Dix, Janet Finlay, Gregory Abowd and Russell Beale. (2003) Human Computer, Interaction (3rd edition), Prentice Hall, ISBN 978-0130461094

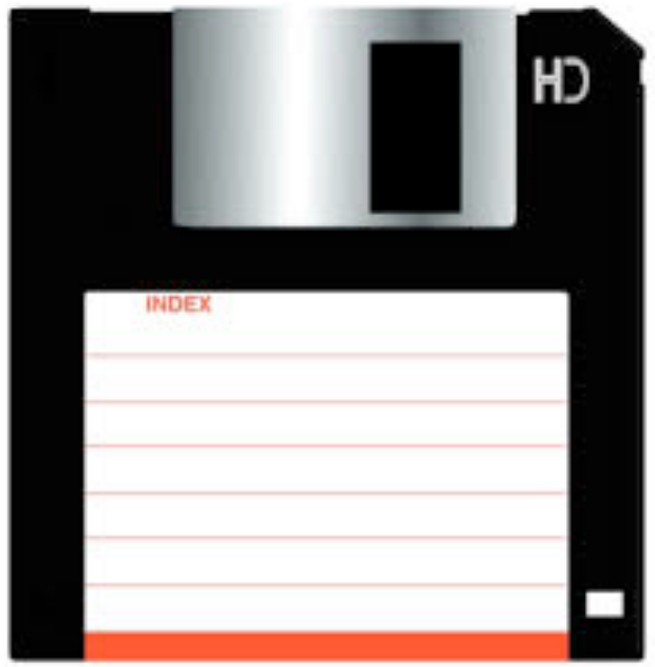
Basic HCI Principles

- Users and Developers
- 3 Usability Principles by Dix et al.
- 3 Usability Principles by Shneiderman
- Human Error
- Background: The Psychology of Everyday Action
- **Constraints, Mappings, Affordances**

Constraints - helpful restrictions

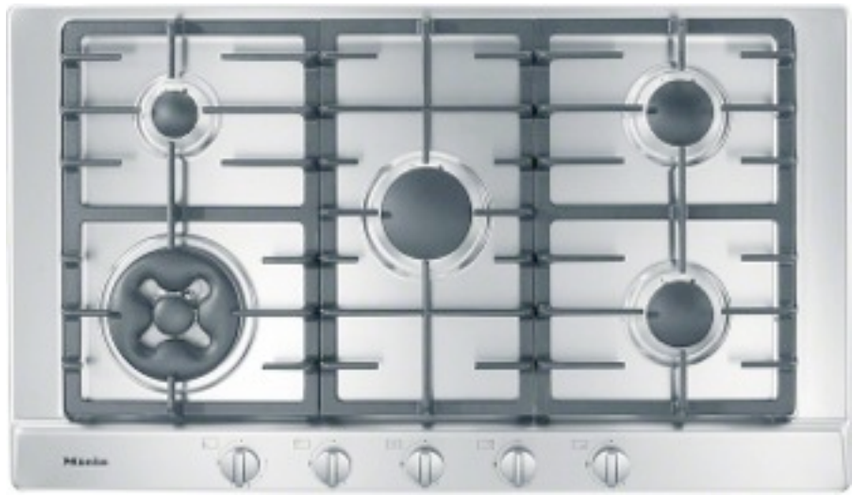
- Physical constraints
  - Basic physical limitations
- Semantic constraints
  - Assumption to create something meaningful
- Cultural constraints
  - Borders and context provided by cultural conventions
- Logical constraints
  - Restrictions due to reasoning
- Applying constraints is a design decision!
  - Practical way to realise the principle “prevent errors”

Constraints: brief discussion

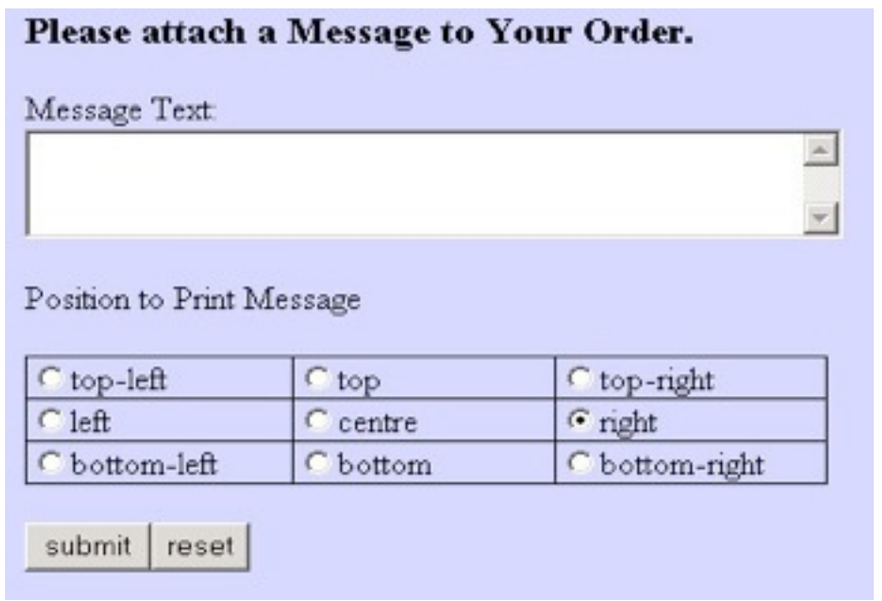


Mappings

- Relationship between controls and action
- Mappings should be
  - **Understandable**
    (e.g. moving the mouse up also moves the slider up)
  - **Consistent**
  - **Recognizable** or at least quickly learnable and easy to recall
  - **Natural**, i.e. consistent with knowledge the user already has



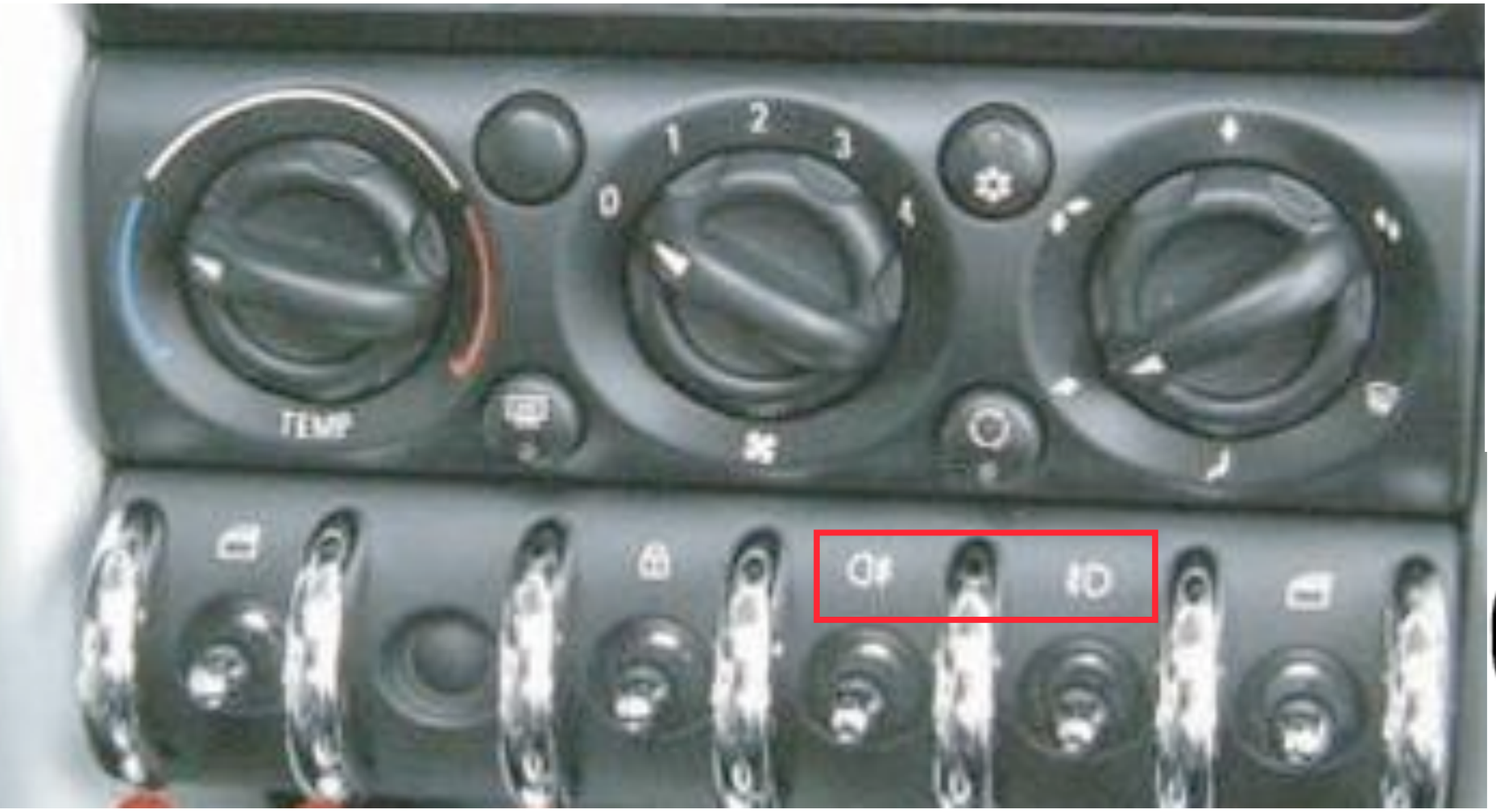
Mappings: Examples



Mapping & Gulf of Execution

- Switch row on dashboard of a car:

ISO 2575



- Right row: The upper symbol is no 4.21 for front fog lamp, the second is no. 4.22 for rear fog lamp

Newer isn‘t always better! ;-)

Informatik, Univ. des Saarlandes

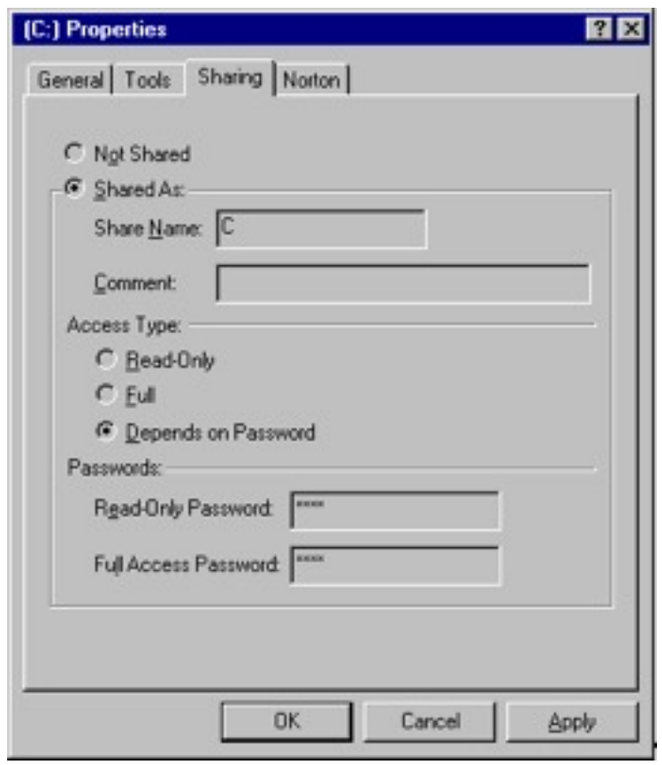
DFKI Saarbrücken

No comment!

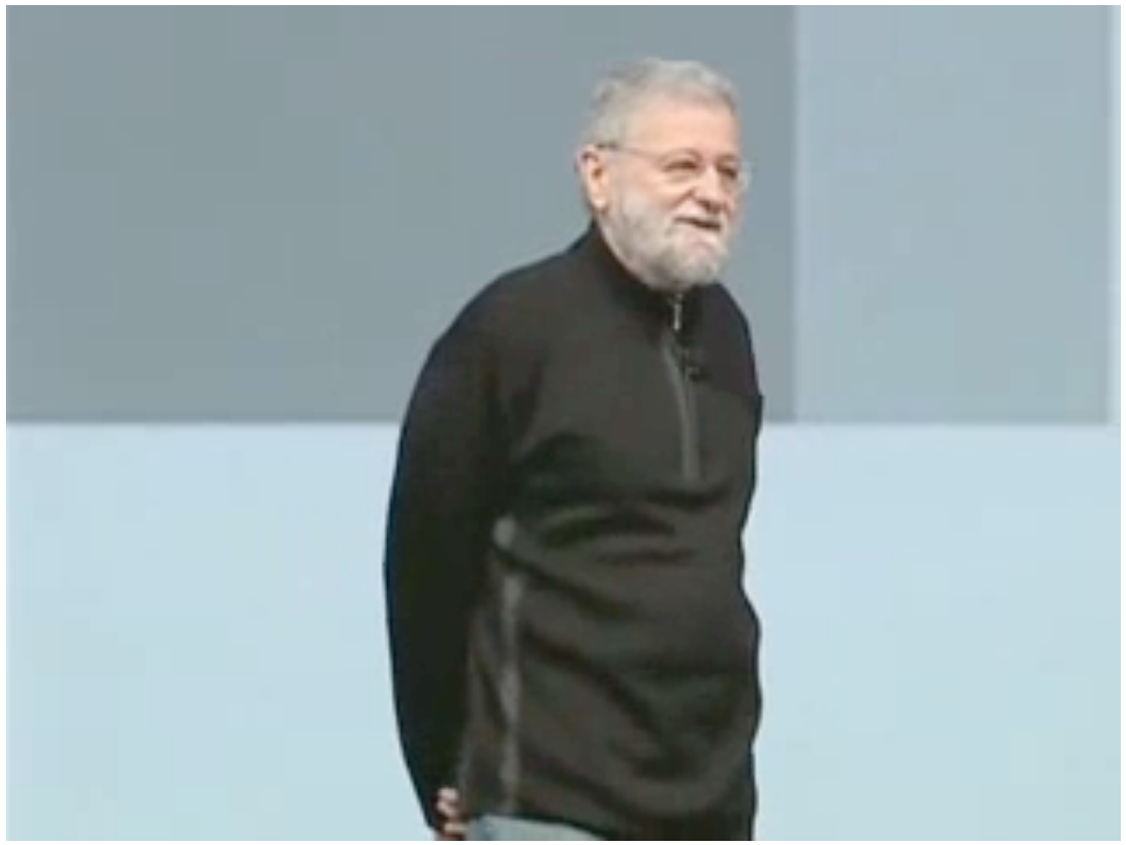
Affordances

- Concept from cognitive psychology
  - brought into HCI by Don Norman
  - newer term: „signifier“
- Objects tell us by their shape how they can be used

- also works in the digital world



The user as the ultima ratio...



Donald Norman