Vorlesung Mensch-Maschine-Interaktion

Albrecht Schmidt
Embedded Interaction Research Group
LFE Medieninformatik
Ludwig-Maximilians-Universität München
http://www.hcilab.org/albrecht/

Chapter 2 Basics of HCI and History

Outline of the course

- 1 Introduction
- **2 Basics of HCI and History**
- 3 Designing Systems for Humans
- 4 Analysis
- 5 Designing Interactive Systems
- 6 Implementing Interactive Systems
- 7 Evaluation

Chapter 2
Basics of HCI and History

- 2.1 Motivation
- 2.2 Principles for UI-Design
  - Principle 1: Recognize User Diversity
  - Principle 2: Follow the Eight Golden Rules
  - Principle 3: Prevent Errors
- 2.3 Understanding Errors
- 2.4 Consistency
- 2.5 Basic Models
- 2.6 A Brief History of HCI

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

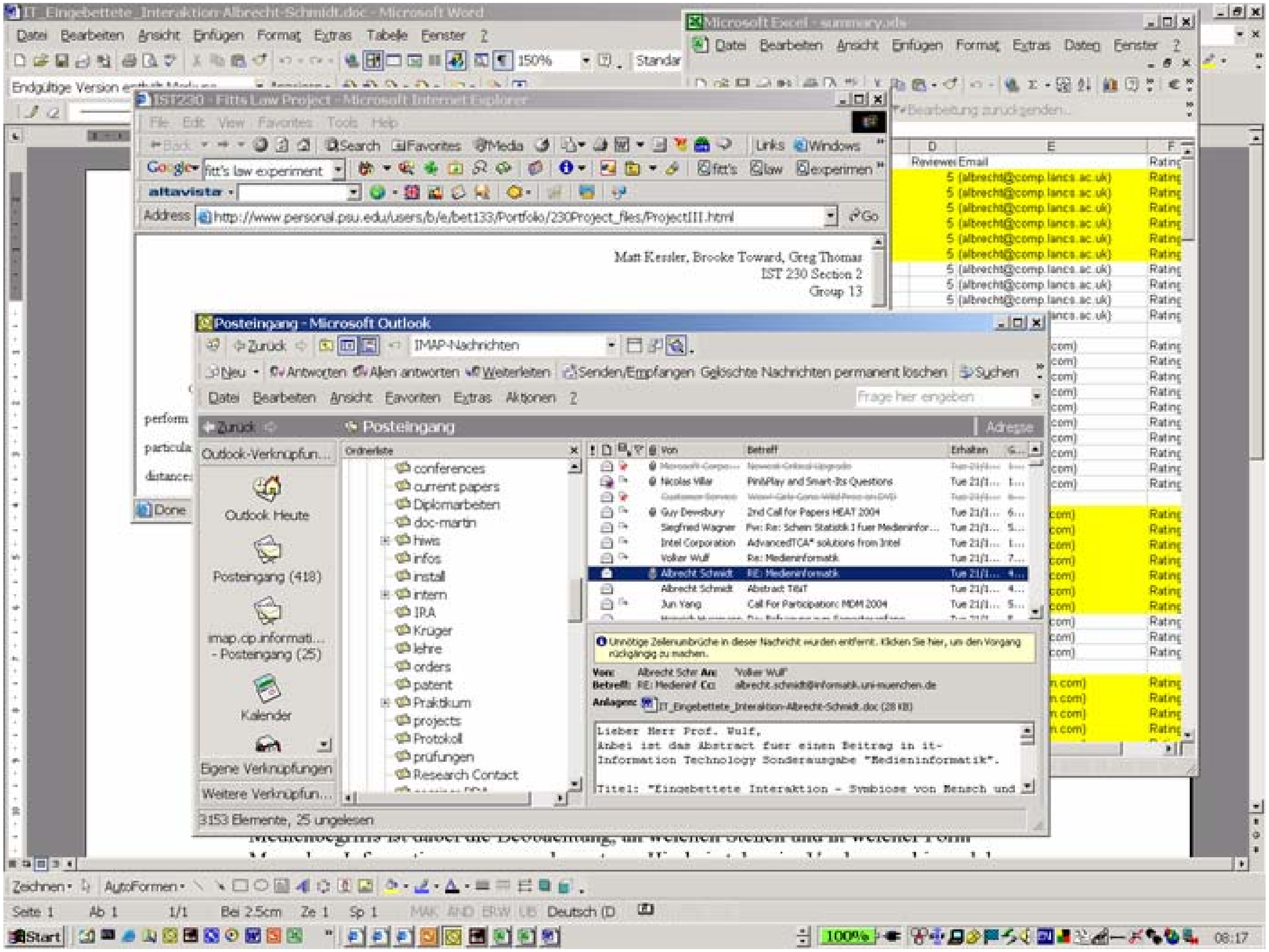
What the User Sees

- Users see only what is visible!

What the Developer Knows

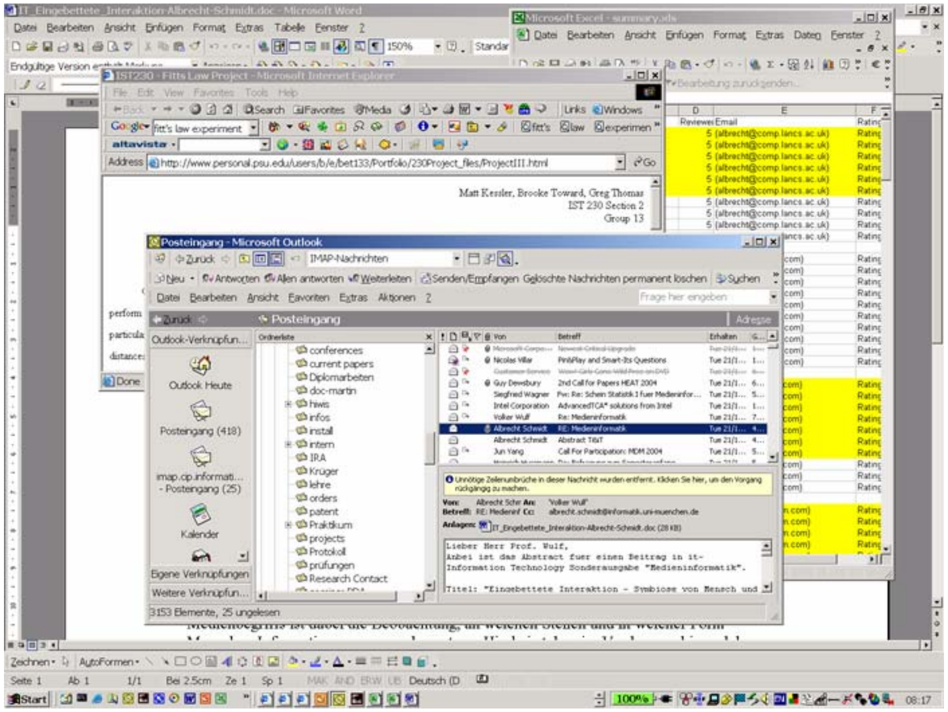
- Users see only what is visible!
- users have little idea about:
  - architecture,
  - state transitions,
  - dependencies
  - application context
  - system restrictions
  - …
- And users often do not want to know about it.

Practical Motivation



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

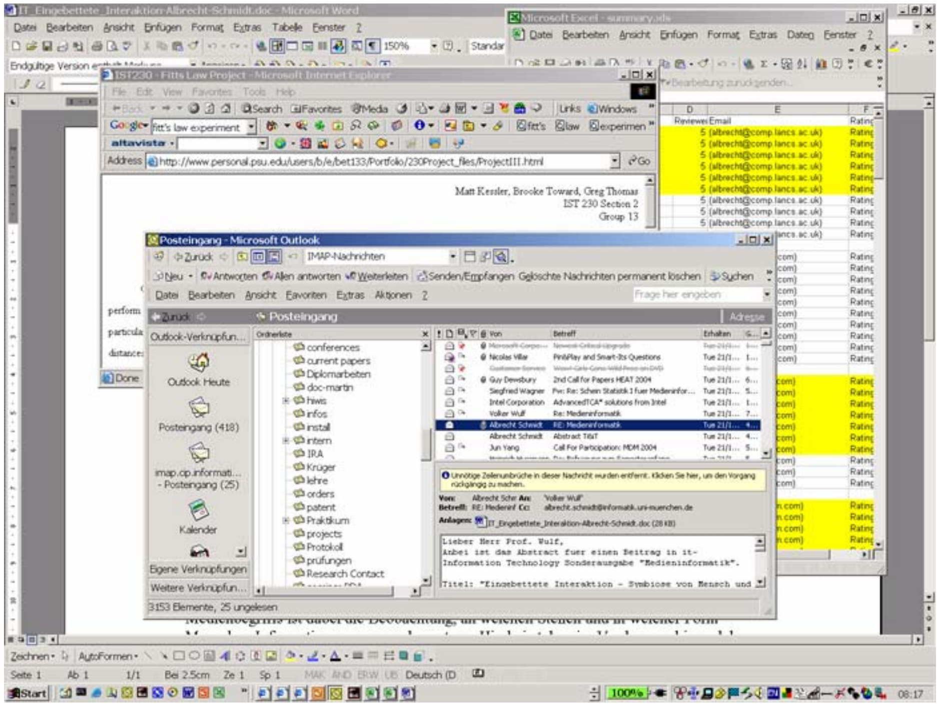
Skilled Computer Users Answers



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

Basic (Naive) Technical Answers

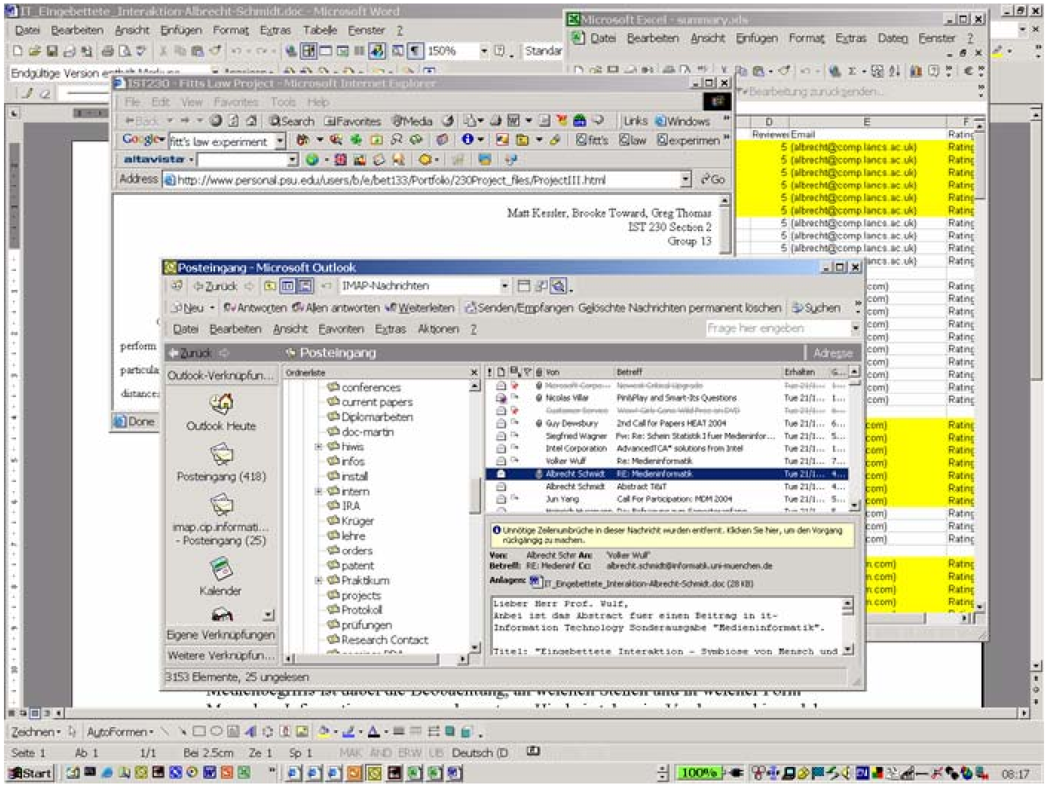
- 2-D surface
- Controllable pixels



- 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

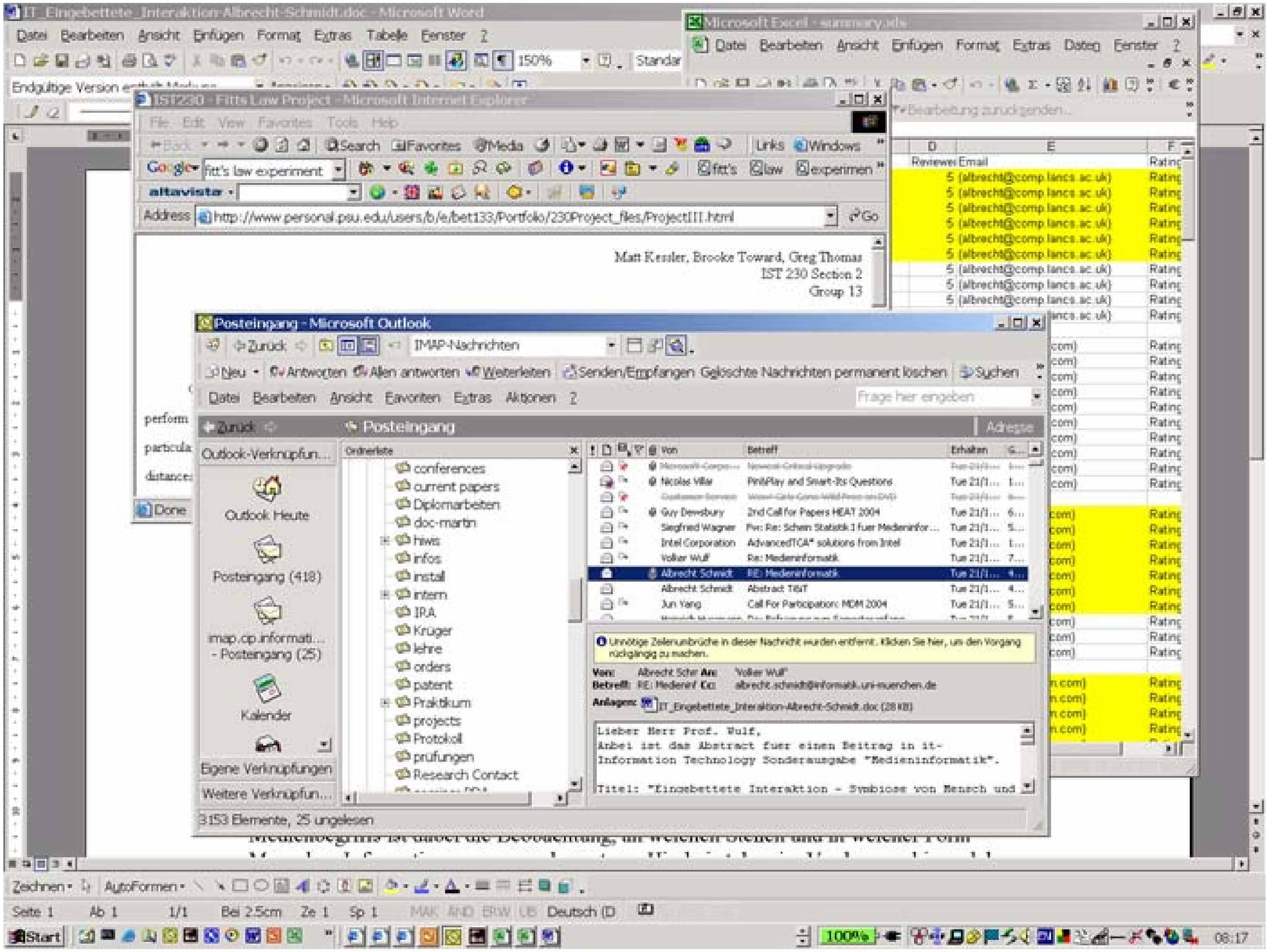
- My work environment



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

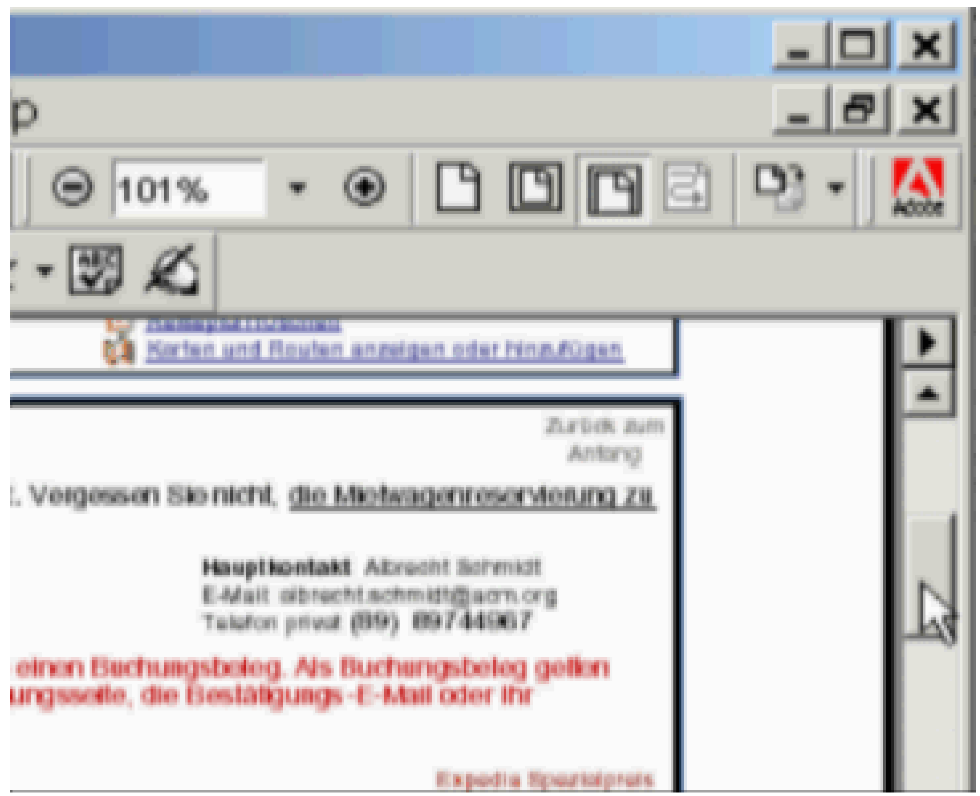
Example I – Overlaying Windows

- What is the meaning that a window is behind another window?
- What is real? What is illusion?
- What does iconizing do?
- Models? Conceptual… Implementation… Represented…



Example II – Scrolling vs. Hand

- Moving up the scroll bar
  Moves down the document

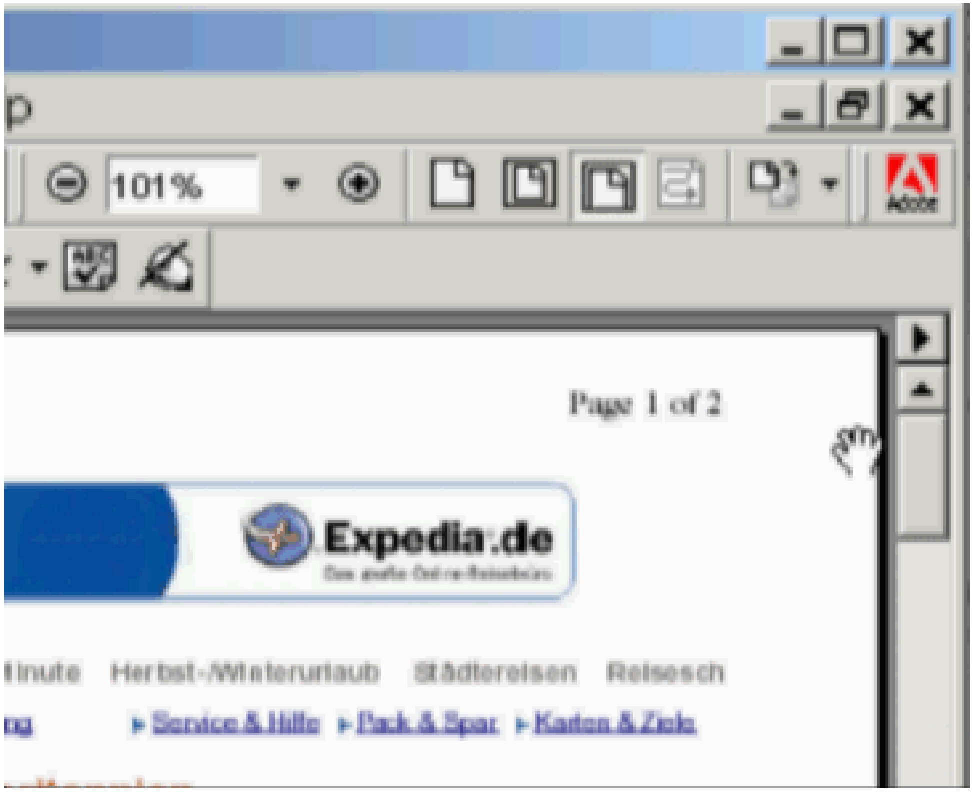


- What happens in reality?
  What do we imagine?
  What is the metaphor?

Example II – Scrolling vs. Hand

- Moving up the hand
  Moves up the document

- What happens in reality?
  What do we imagine?
  What is the metaphor?



Chapter 2
Basics of HCI and History

- 2.1 Motivation
- **2.2 Principles for UI-Design**
  - **Principle 1: Recognize User Diversity**
  - Principle 2: Follow the Eight Golden Rules
  - Principle 3: Prevent Errors
- 2.3 Understanding Errors
- 2.4 Consistency
- 2.5 Basic Models
- 2.6 A Brief History of HCI

Principles for UI design

- Implementation and technology independent
- Shneiderman's principles:
  (see http://media.pearsoncmg.com/aw/aw_shneiderma_dtui_4/chapter2.pdf)
  - **Principle 1 : Recognize User Diversity**
  - **Principle 2 : Follow the Eight Golden Rules**
  - **Principle 3 : Prevent Errors**
- Restated in different variants – basically telling the same story

Principle 1: Recognize User Diversity

- Simple and obvious - nevertheless in reality extremely difficult

- Example: consider a online travel agent
  - Travel agent booking many flights a day – everyday
  - A teacher organizing a field trip (once a year) and making bookings for a large group
  - A business person changing bookings while travelling
  - A family looking for a package holiday

- Basic concepts to structure the problem
  - Usage profiles
  - Task profiles

Usage Profiles “Know Thy User”

- What is the background of the user?
- Different people have different requirements for their interaction with computers.

- Issues to take into account:
  - goals, motivation, personality
  - education, cultural background, training
  - age, gender, physical abilities, …

- Experience:
  - Novice users
  - Knowledgeable intermittent users
  - Expert frequent users

User-Needs and Task Profiles

- Find out what the user is trying to do! The Goal!
- Needs of users, goals and resulting tasks
- Supported tasks should be determined before the design starts
- Functionality should only be added if identified to help solving tasks
  - Temptation: If additional functionality is cheap to include it is often done – this can seriously compromise the user interface concept!
- Frequency of tasks related to user profiles

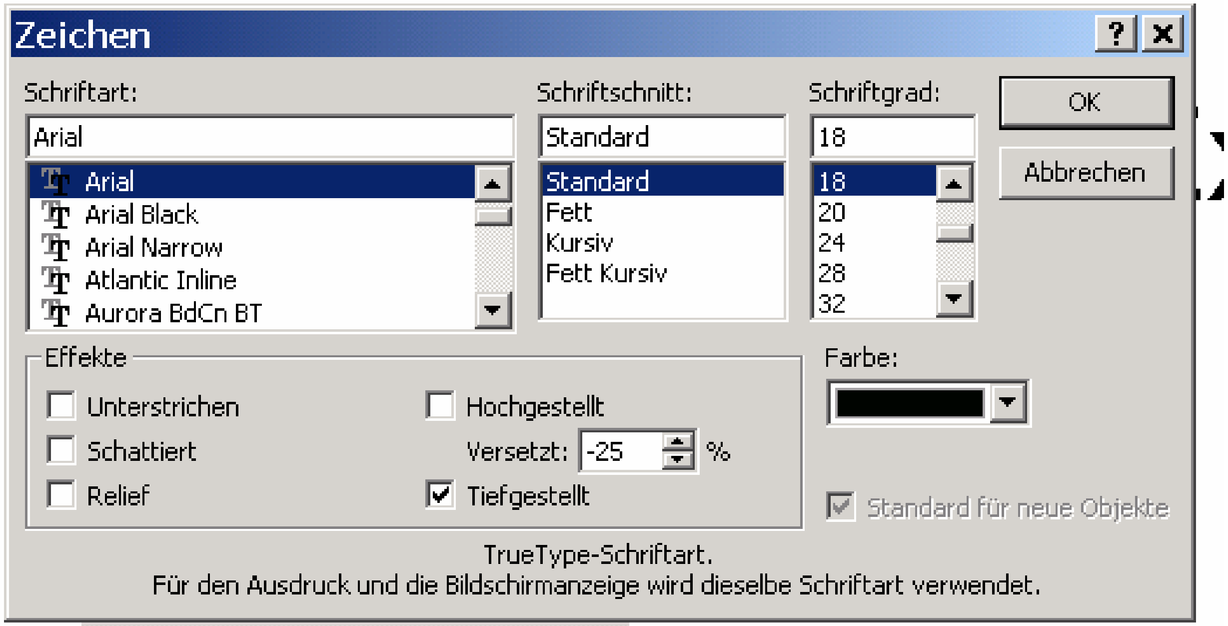
Hypothetical Frequency of Tasks
(Example of a booking system for travel)

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

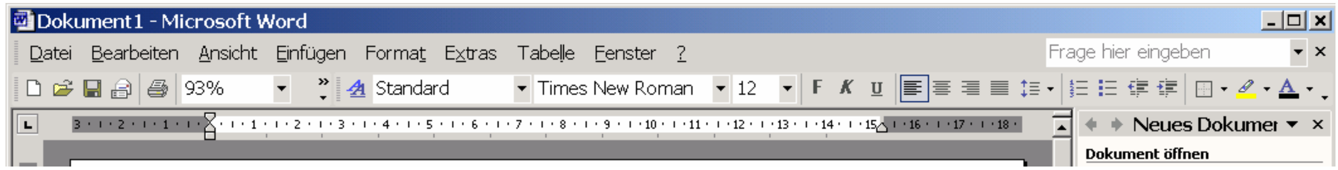
- Helps to shape a menu structure
  - Frequent action should be simple and quick to carry out
  - Infrequent action may take longer
- Example
  - Frequent actions: Toolbar or special key
  - Intermediate frequent actions: Pull-down menu, key combination (Ctrl+S)
  - Infrequent actions: Sequence of menus or dialogs
- Problem – if many (all) actions occur with very similar relative frequency…

Task Frequency - Examples

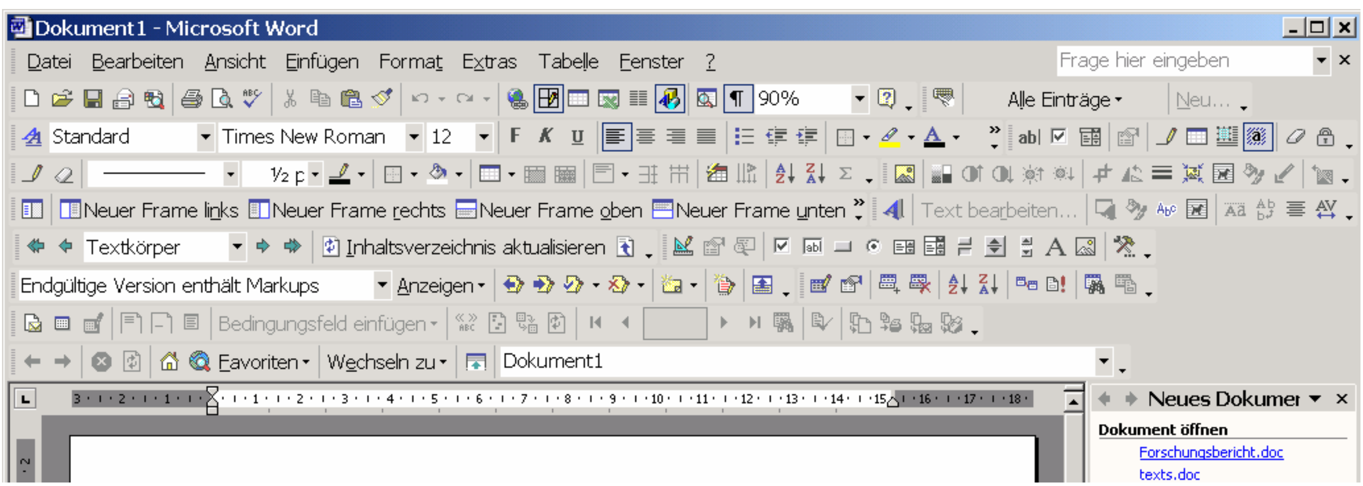


- Bold is available in the toolbar
- Subscript requires menu and dialog
- Assumption for the standard UI is that user needs more often bold than subscript
- For users with different needs the customization is available

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



Albrecht Schmidt
Embedded Interaction Research Group
University of Munich, Germany

Chapter 2
Basics of HCI and History

- 2.1 Motivation
- **2.2 Principles for UI-Design**
  - Principle 1: Recognize User Diversity
  - **Principle 2: Follow the Eight Golden Rules**
  - Principle 3: Prevent Errors
- 2.3 Understanding Errors
- 2.4 Consistency
- 2.5 Basic Models
- 2.6 A Brief History of HCI

Principle 2:
Follow the 8 Golden Rules

- Strive for consistency
- Enable frequent users to use shortcuts
- Offer informative feedback
- Design dialogues to yield closure
- Error prevention/handling
- Permit easy reversal of actions
- Support internal locus of control
- Reduce short-term memory load

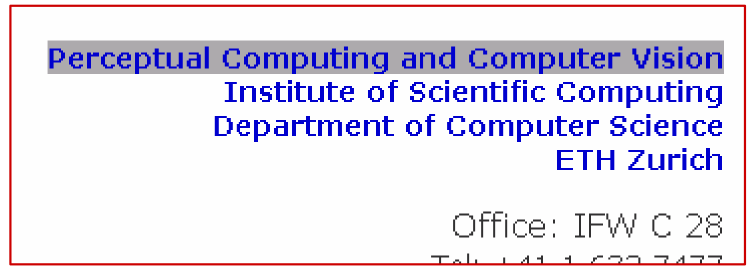
Shneiderman, chapter 2

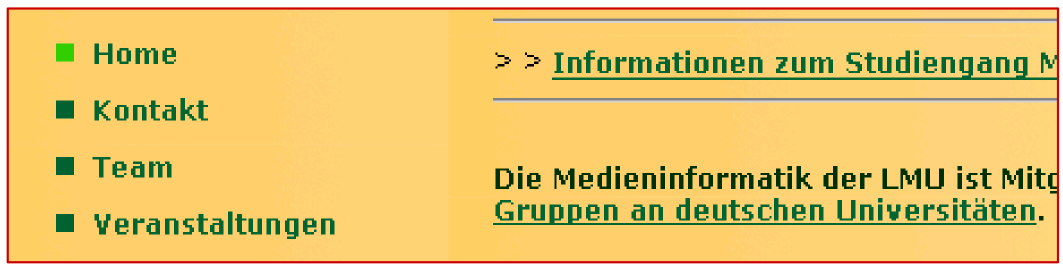
8 Golden Rules - Consistency

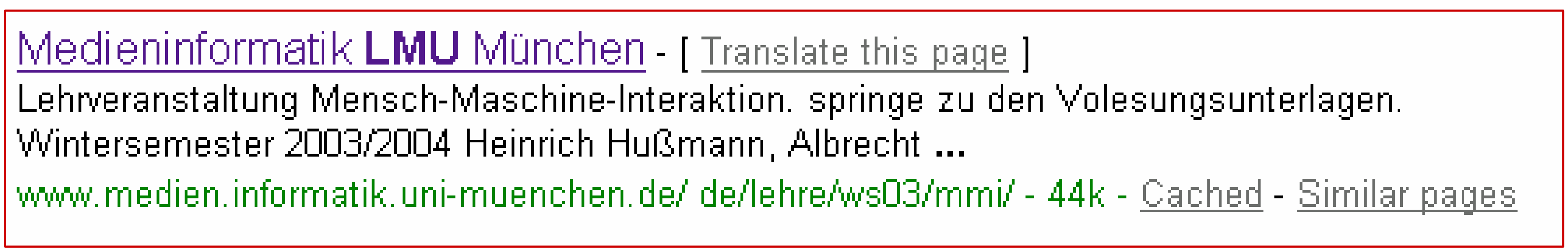
- Within an application it is the developer's job (see earlier slides…but that is the easy part)

- In a specific environment it is defined by guidelines (e.g. for GNOME, for KDE, for Mac OSX, for Win XP, for JAVA Swing)

- In the WWW it gets pretty hard!
  - No real guidelines and no authority
  - How are links represented?
  - Where is the navigation?
  - Styles and "fashion" change quickly…







8 Golden Rules - 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 preset (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 applications for cut, copy and paste
  - However CTRL+S (saving a document) is only implemented in some applications…

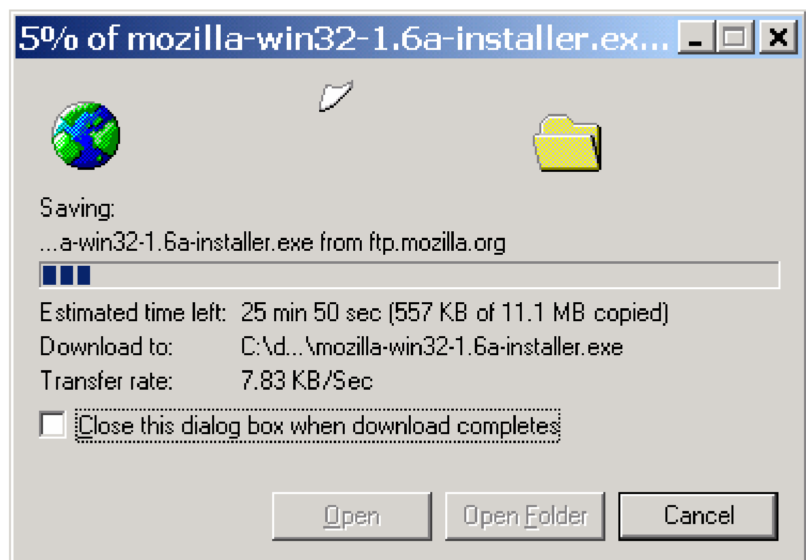
8 Golden Rules - Feedback

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

- For infrequent action is should be more substantial

8 Golden Rules - Closure

- Sequences of actions should have a beginning, middle, and end.
- For non-instantaneous actions
- 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 check-out
  - E.g. in the small: a progress bar



8 Golden Rules – Prevent Errors

- Create UI that make it hard to make errors (e.g. menus instead of commands)
- Detect errors or possible errors
- Is related to “easy reversal of actions”

- Examples
  - Leaving a editor without saving
  - Writing to a file that already exists

- Different options how to handle it:
  - Involve the user (current practice)
  - Prevent the error or its consequences on system level (e.g. create backups/versions when a file is overwritten, keep all files that have been created by the user)

8 Golden Rules – Permit Easy Reversal of Actions

- As a basic rule – all actions should be reversible
- Providing UNDO functions (possibly with infinite depth)
- Allow undo of groups of actions

- Undo is not trivial if user is not going sequential
  - 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 look, … don’t like it, … go back to the old state”

8 Golden Rules - Feeling in Control

- Users should feel to be in control of the system
- User should initiate actions (initiator instead of responder)
- Avoid non-causality
- The system should be predictable

- Some current developments are in contrast:
  - Proactive computing
  - Intelligent agents
- Have to be aware when designing these!

8 Golden Rules – Reduce Short-term Memory Load

- 7 +/- 2 chunks of information
- The system should remember, not the user

- 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
  - Make information that is required visible
  - Use memory aids (visual or audio)

Chapter 2
Basics of HCI and History

- 2.1 Motivation
- **2.2 Principles for UI-Design**
  - Principle 1: Recognize User Diversity
  - Principle 2: Follow the Eight Golden Rules
  - **Principle 3: Prevent Errors**
- 2.3 Understanding Errors
- 2.4 Consistency
- 2.5 Basic Models
- 2.6 A Brief History of HCI

Principle 3: Prevent Errors - Examples

- Correct matching pairs
  - Examples:
    - Making some text <B>**bold**</B> will make too much bold if the </B> is omitted or mistyped
    - IDE often provide {} match checking
- Complete sequences
  - Assistance for the user to complete a sequence of actions to perform a task
  - Example: Wizards
- Command correction
  - Aim: Trying to prevent users entering incorrect commands
  - Examples:
    - File completion on Unix
    - Helpful error messages

Chapter 2
Basics of HCI and History

Human Error

Top News

Bombardier 'Stands Down' Against Human Error

Wed, 26 Oct '05

Event Aims To Reduce Pilot Mistakes

More than 420 pilots, crewmembers, safety specialists, industry officials and media representatives have gathered at Bombardier's 9th Annual Safety Standdown in Wichita, KS. The event, billed as the industry's foremost safety event, is being held Oct. 25-27.

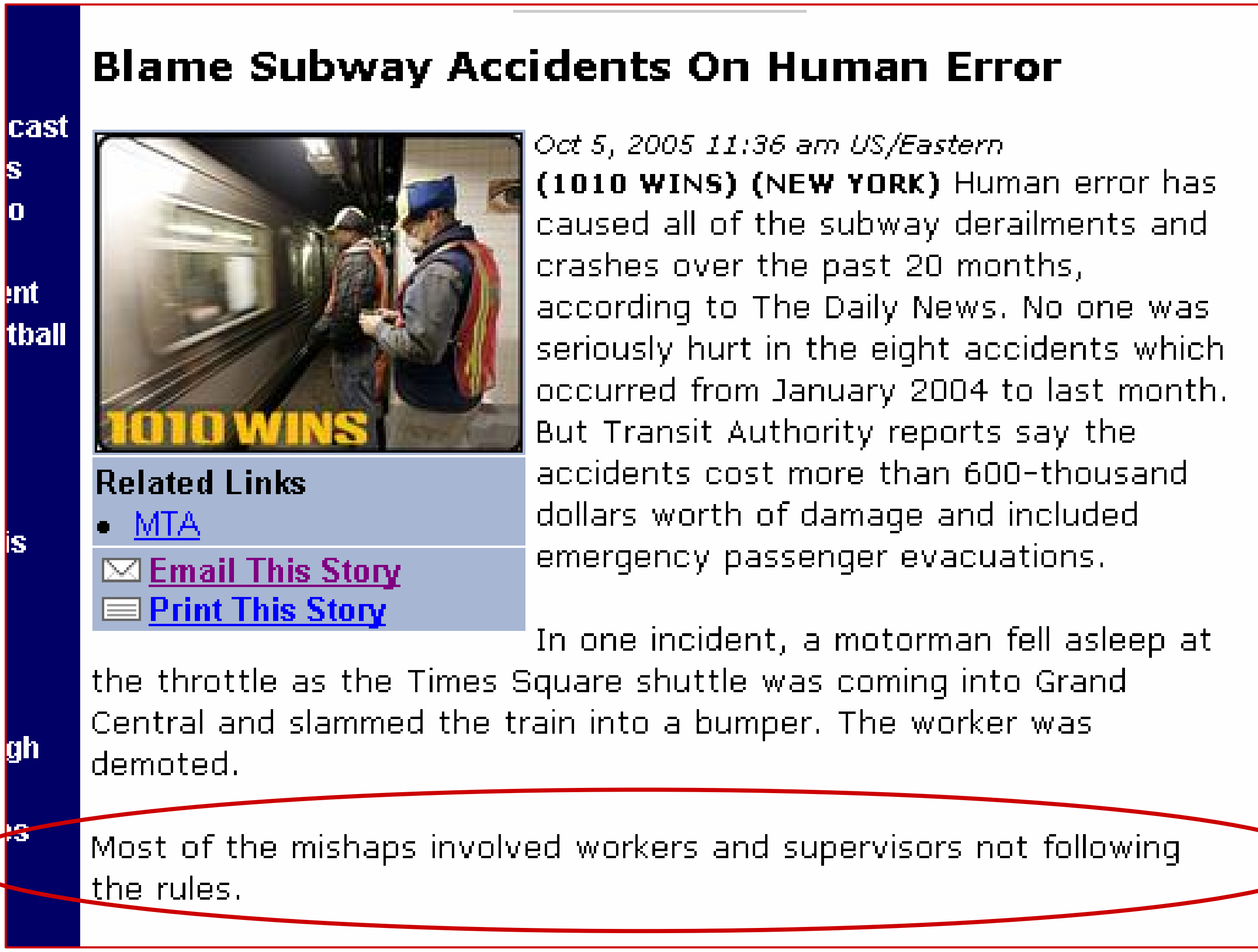
The only safety seminar of its kind to be offered by a civil aircraft manufacturer, Bombardier's Safety Standdown is taking clear aim at the cause of 78 percent of all accidents in aviation -- human error.

"The intent of Safety Standdown is to reduce accidents caused by human failure across the aviation industry as a whole, whether they occur during corporate, commercial or military missions," stated Bob Agostino, director, flight operations, Bombardier Business Aircraft. "While we believe current training programs using simulators and other training devices are excellent, we also recognize that accident prevention requires more than simply perfecting technical skills."

This year's event will focus on "Winning The War On Error," enabling aviation professionals to better understand why and how crucial mistakes occur by providing in-depth, knowledge-based training in areas such as fatigue, nutrition and psychological factors.

- http://www.aero-news.net/index.cfm?ContentBlockID=cda9332e-b872-4d41-960a-2352e5f47744

Human Error



Blame Subway Accidents On Human Error

Oct 5, 2005 11:36 am US/Eastern

(1010 WINS) (NEW YORK) Human error has caused all of the subway derailments and crashes over the past 20 months, according to The Daily News. No one was seriously hurt in the eight accidents which occurred from January 2004 to last month. But Transit Authority reports say the accidents cost more than 600-thousand dollars worth of damage and included emergency passenger evacuations.

In one incident, a motorman fell asleep at the throttle as the Times Square shuttle was coming into Grand Central and slammed the train into a bumper. The worker was demoted.

Most of the mishaps involved workers and supervisors not following the rules.

Related Links

- MTA

Email This Story

Print This Story

- http://1010wins.com/topstories/local_story_278071424.html

Human Error

PITTSBURGH TRIBUNE-REVIEW

Back to headlines

Larger Text 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.

McKesson

Founded: Healthcar Sean McD student a University

Acquired Healthcar by drug d McKesso million.

Headqua

Presiden Souerwin

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

more (Human) Errors…

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.

About (Human) Errors…

- "If an error is possible, someone will make it" (Norman)
- Human Error may also 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
  - Attempt to 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 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

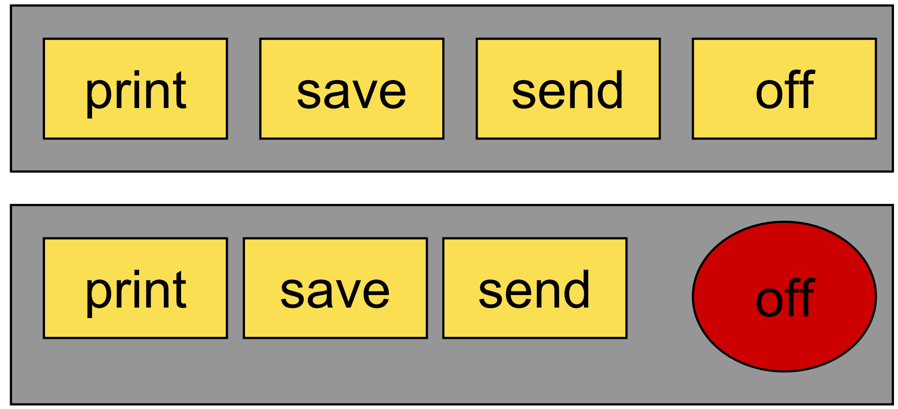
Confirmation is unlikely to prevent Errors

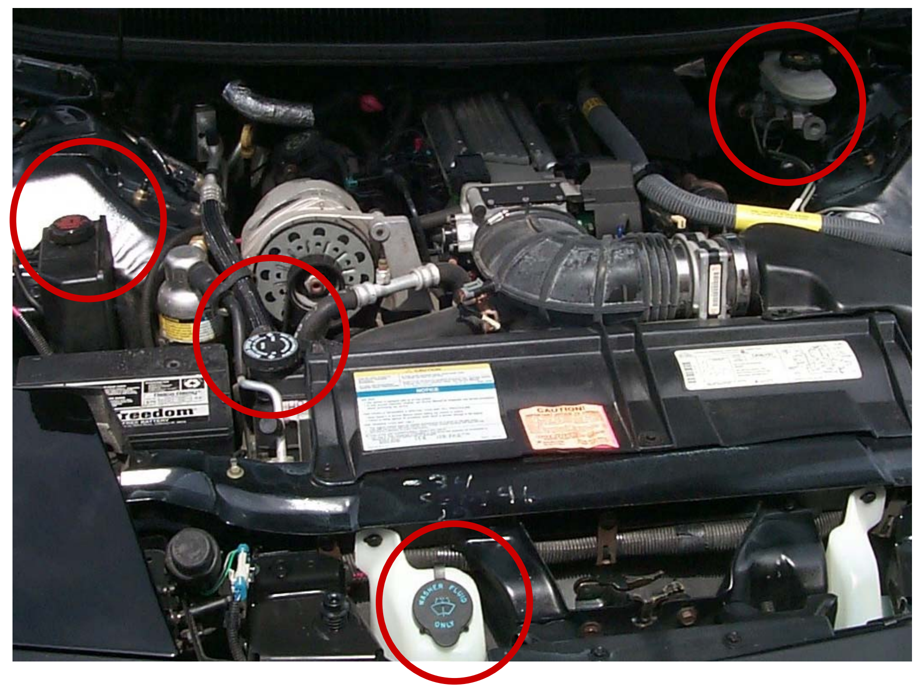
- Example
  - User: "remove the file '**`most-important-work.txt`**'"
  - computer: "are you sure that you want to remove the file '**`most-important-work.txt`**'?"
  - User: "yes"
  - Computer: "are you certain?"
  - User: "yes of course"
  - Computer: "the file '**`most-important-work.txt`**' has been removed"
  - User: Oops, damm
- The user is not reconsidering the overall action – it only prompts to think about the immediate action (clicking)
- A solution is to make the action reversible

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
- What is your solution?
  - Draw the control elements
  - Provide labels

Setting time and alarm with mode?

Setting time and alarm without mode?

Making things reversible

- Is a great solution –
  but where is the problem with it?

- What is the cost?

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

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

- 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

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)

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”

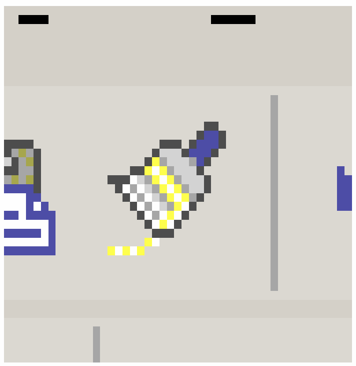
- Applicability
  - to command line user interfaces
  - Keyboard short cuts
  - Speech interfaces
  - Tool bars
  - Menus
  - Selection operation
  - Gestures

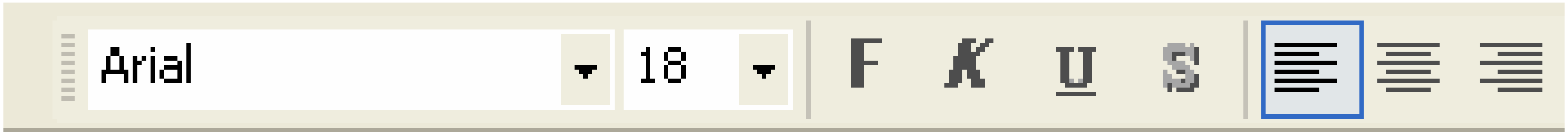
Consistency through Grammars

- 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

How does the Format Brush work?

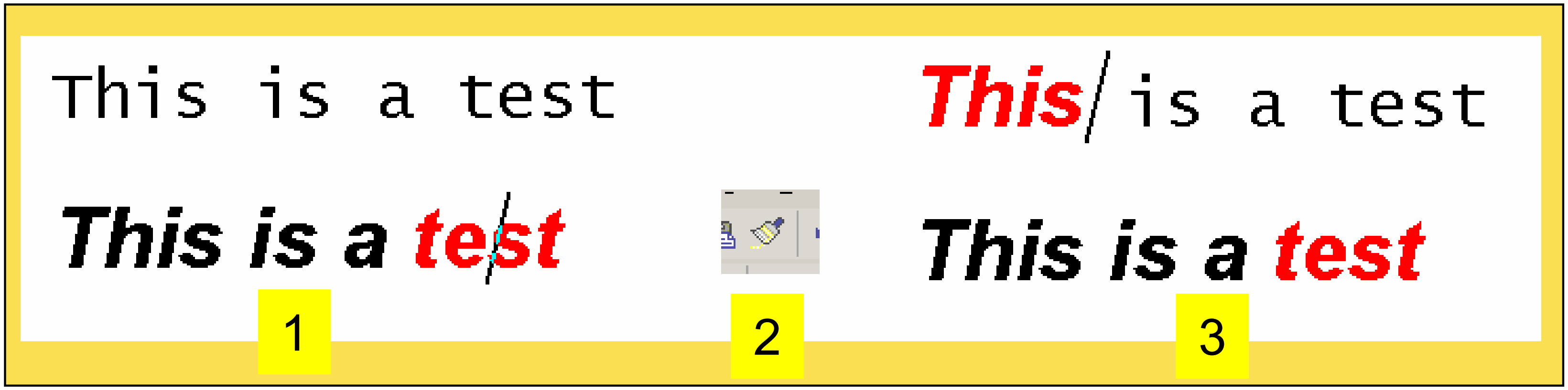




- compare it to bold, italic, underline, …

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
- Bold face font (2)
  1. Switch bold face font on (Click the toolbar button for bold)
  2. Write text
  3. Switch it of when ready

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

Chapter 2
Basics of HCI and History

Models & Theories

- What are models and theories used for?
  - explanatory
  - **predictive**
  - descriptive/taxonomy
- Models on different levels
  - concept
  - human action
  - …
  - dialog
  - keystroke
- What is modelled?
  - user
  - task
  - dialogs
  - transitions
  - software
  - input/output
  - system
  - interaction
  - behaviour
  - …
  - combination of these

Example Motivation - Prediction

http://www.xe.com/ucc/

- 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

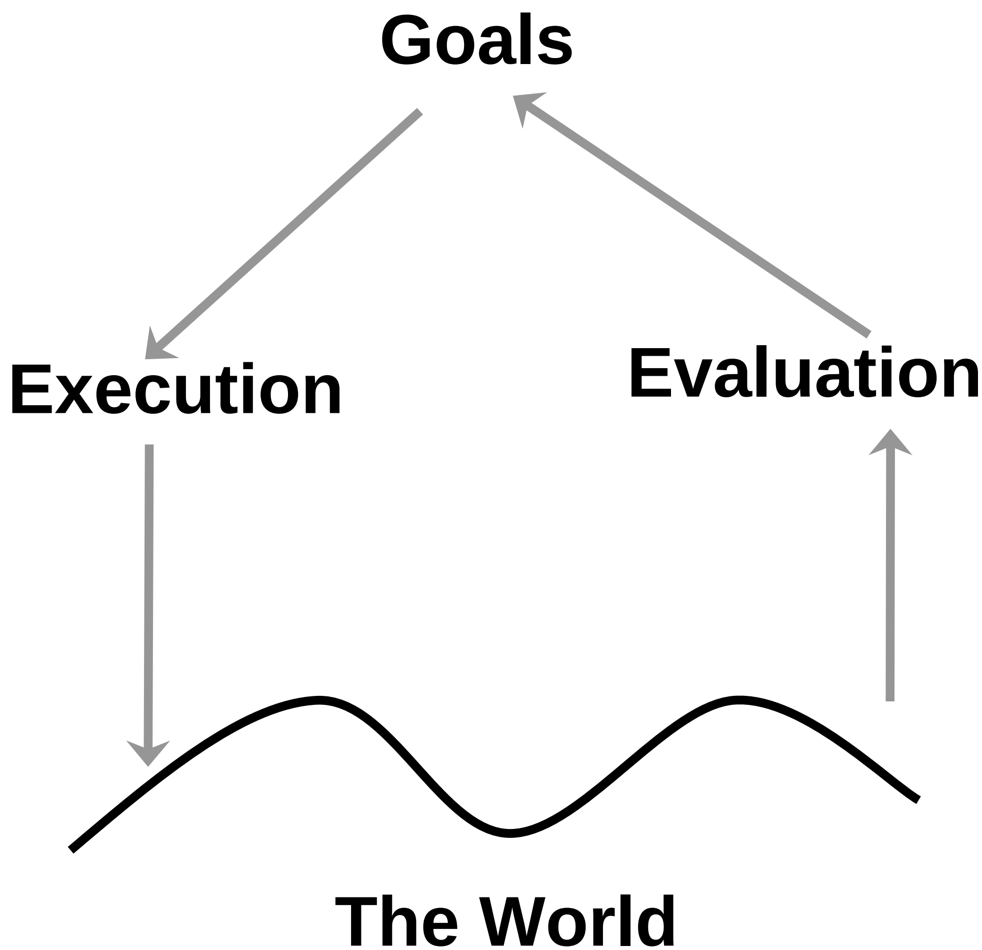
This questions many of the modeling approaches…

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

- 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

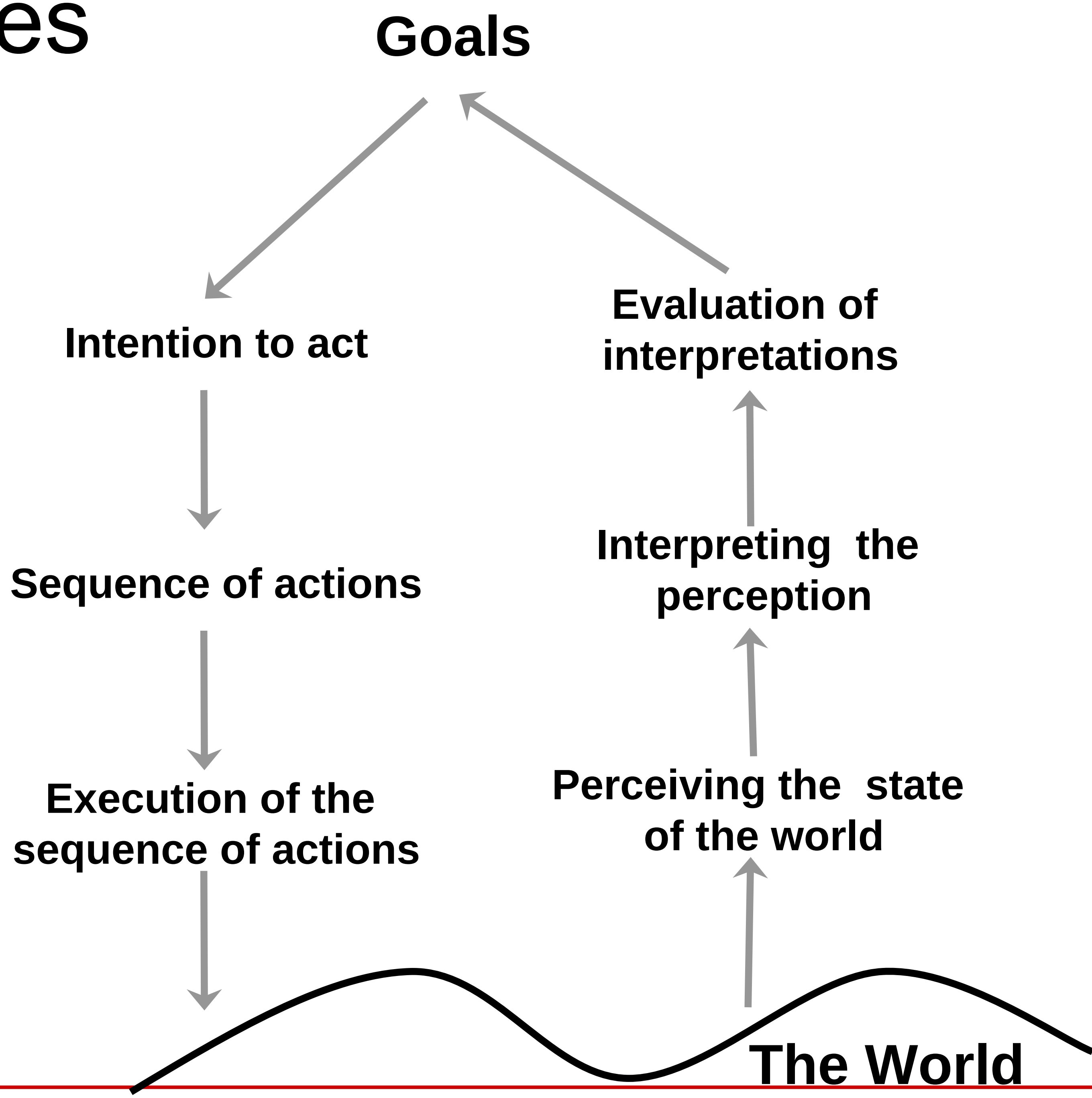
  followed by
- Interpreting the perception according to our expectations

  followed by
- Evaluation of the interpretations with what we expected to happen (original intentions)

  followed by
- Goal

Seven Stages of Action

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



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

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 intensions?

- 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

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 many not be able to specify / execute the desired action
  - Inappropriate / mismatching feedback

Fitts' Law
Predicting Movement Time (MT)

- $MT = a + b \;\log2(2A / W)$
  - 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.
  - Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology, 47*, 381-391.

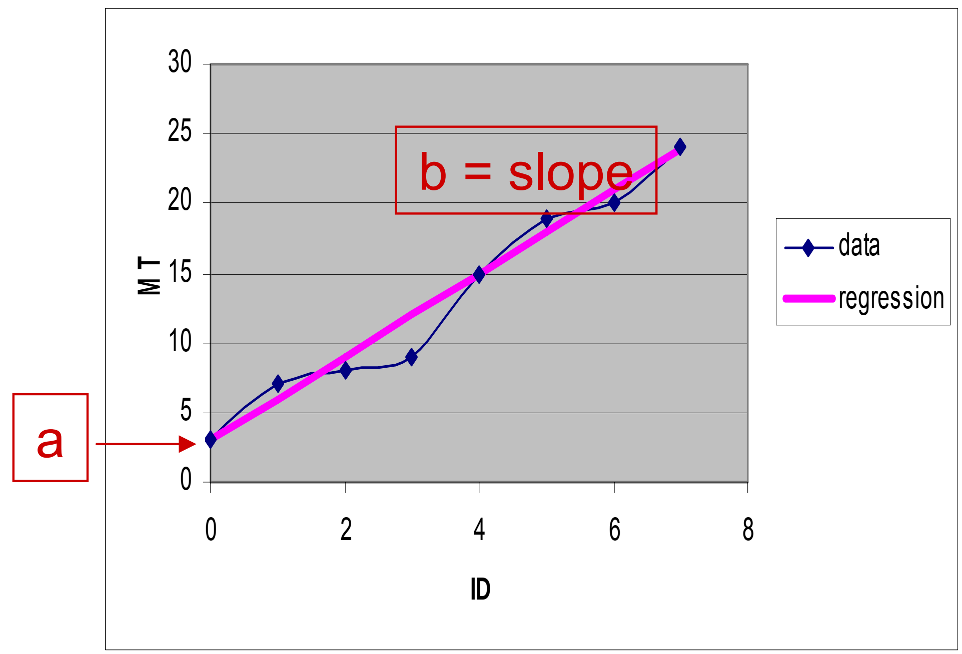
- $MT = a + b \;\log2(A / W + 1)$
  - improvement of the original fitts' law
  - MacKenzie, I. S. (1989). A note on the information-theoretic basis for Fitts' law. *Journal of Motor Behavior*, *21*, 323-330.

http://www.billbuxton.com/fitts91.html

Fitts' Law – index of difficulty

- How difficult the motor pointing task is
- ID=Index of Difficulty
- ID=log2(A/W + 1)
- ID has the unit bits

- *MT* = *a* + *b* ID
- a has the unit s
- b has the unit s/bits
- Collect data set and calculate a and b
- a can be negative



linear regression model

Fitts' law in practice

- $MT = a + b \log2((A/W) + 1)$
- 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

- Jef Raskin, The Humane Interface, ACM Press 2000, p93-94

Experimental data for pointing devices

MT = *a* + *b* *ID*, where *ID* = log2(*A*/*W* + 1).

Regression Coefficients

Device	r^a	Intercept, *a* (ms)	Slope, *b* (ms/bit>	*IP* (bits/s)[b]
		*** Pointing ***		
Mouse	.990	-107	223	4.5
Tablet	.988	-55	204	4.9
Trackball	.981	75	300	3.3
		*** Dragging ***		
Mouse	.992	135	249	4.0
Tablet	.992	-27	276	3.6
Trackball	.923	-349	688	1.5

[a] $n = 16$, $p < .001$

[b] *IP* (index of performance) = 1/*b*

- From http://www.billbuxton.com/fitts91.html
  MacKenzie, I. S., Sellen, A., & Buxton, W. (1991). A comparison of input devices in elemental pointing and dragging tasks. *Proceedings of the CHI `91 Conference on Human Factors in Computing Systems*, pp. 161-166. New York: ACM.

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 H
- n alternatives of equal probability
  H = log2(n + 1).
- Alternatives of unequal probability
  pi = the probability of alternative i
  H = Σ pi log2(1/pi + 1).
- Common practical values: b=150 ms/bit
- http://www.usabilityfirst.com

Hick's law does not apply if it requires linear search (e.g. a randomly ordered list of commands in a menu). It applies if the user can search by sub-division

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

http://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/oai.html

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)

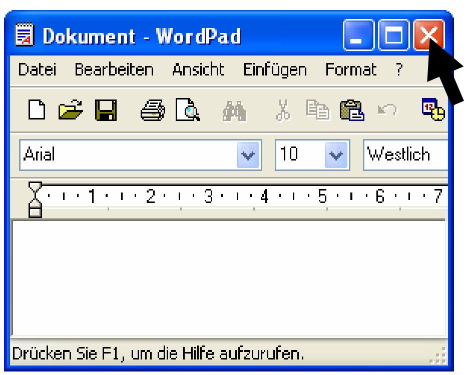
- John, B. & Kieras, D. (1996). Using GOMS for user interface design and evaluation: which technique? *ACM Transactions on Computer-Human Interaction, 3,* 287-319.

Example (adapted from Dix 2004, p. 423):
Close the window that has the focus (Windows XP)

- Compare three options:

ALT + F4 Key-shortcut

Context-menu



Close-button

GOAL: CLOSE-WINDOW
. [select GOAL: USE-KEY-SHORTCUT
. . hold-ALT-key
. . press-F4-key
. GOAL: USE-CONTEX-MENU
. . Move-mouse-win-head
. . Open-menu (right click)
. . Left-click-close
. GOAL: USE-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: LOCATE-ARTICLE
. GOAL: COPY-PAGE repeat until no more pages
. GOAL: ORIENT-PAGE
. . OPEN-COVER
. . SELECT-PAGE
. . POSITION-PAGE
. . CLOSE-COVER
. GOAL: PRESS-COPY
. GOAL: VERIFY-COPY
. . LOCATE OUTPUT
. . EXAMINE COPY
. GOAL: COLLECT-COPY
. . LOCATE OUTPUT
. . REMOVE-COPY
(outer goal satisfied!)
. GOAL: RETRIEVE-ORIGINAL
. . OPEN-COVER
. . 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..

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

- Design to keep your card..

```
GOAL: GET-MONEY
.   GOAL: USE-CASH-MACHINE
.   .     INSERT-CARD
.   .     ENTER-PIN
.   .     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 ~~each of~~ the components of the model (goals, operators, methods, and selection rules), the concept of level of detail, and the different computational forms that GOMS models take. In this section, we will define each of these concepts; in subsequent sections we will categorize existing GOMS models according to these concepts.

Figure 1. The example task: editing a marked-up manuscript.

- From: John, Bonnie and Kieras, David E., The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast, *ACM Transactions on Computer-Human Interaction 3,4* (December 1996b), 320-351

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 an keyboard (H)
  - mental operators (M) – placed by heuristics
  - Drawing (D)
  - System response (R)

- Card, S. K., Moran, T. P., and Newell, A. 1980. The keystroke-level model for user performance time with interactive systems. *Commun. ACM* 23, 7 (Jul. 1980), 396-410.

Times for basic operators

Operator	Description and Remarks	Time (sec)
K	Keystroke or button press. Pressing the SHIFT or CONTROL key counts as a separate K operation. Time varies with the typing skill of the user; the following shows the range of typical values:	
	Best typist (135 wpm)	.08[a]
	Good typist (90 wpm)	.12[a]
	Average skilled typist (55 wpm)	.20[a]
	Average non-secretary typist (40 wpm)	.28[b]
	Typing random letters	.50[a]
	Typing complex codes	.75[a]
	Worst typist (unfamiliar with keyboard)	1.20[a]
P	Pointing to a target on a display with a mouse. The time to point varies with distance and target size according to Fitts's Law. The time ranges from .8 to 1.5 sec, with 1.1 being an average time. This operator does *not* include the button press that often follows (.2 sec).	1.10[c]
H	Homing the hand(s) on the keyboard or other device.	.40[d]
$D(n_D,l_D)$	Drawing (manually) n_D straight-line segments having a total length of l_D cm. This is a very restricted operator; it assumes that drawing is done with the mouse on a system that constrains all lines to fall on a square .56 cm grid. Users vary in their drawing skill; the time given is an average value.	$.9n_D + .16l_D$[e]
M	Mentally preparing for executing physical actions.	1.35[f]
$R(t)$	Response of t sec *by the system*. This takes different times for different commands in the system. These times must be input to the model. The response time counts only if it causes the user to wait.	t

- Experimentally measured
- From: Card, S. K., Moran, T. P., and Newell, A. 1980. The keystroke-level model for user performance time with interactive systems. *Commun. ACM* 23, 7 (Jul. 1980), 396-410.

Basic time estimation

Operator	Remarks	Time(sec)
K	Press Key	
	good typist(90wpm)	0.12
	poor typist(40wpm)	0.28
	non-typist	1.20
B	Mouse button press	
	down or up	0.10
	click	0.20
P	Point with mouse	
	Fitts's law	0.1lg(D/S +0.5)
	Average movement	1.10
H	Home hands to and from keyboard	0.40
D	Drawing- domain-dependent	__
M	Mentally prepare	1.35
R	Response from sytem - measure	__

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

Calculate overall time required

- $T_{task} = T_{acquire} + T_{execute}$
- $T_{execute} = T_K + T_B + T_P + T_H + T_D + T_M + T_R$
  - 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[c]	0,28s
K[m]	0,28s
K[d]	0,28s
K[return]	0,28s
	5,47s

- T = 2*P+2*B+4*K+H+M

KLM - Example

http://www.xe.com/ucc/

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

KLM – Example result

- P[to input field]
- B[click]
- H[to keyboard]
- M[consider number]
- 4K[BSP-7-1-2]
- H[to mouse]
- M[consider currency]
- P[to GBP]
- B[click]
- M[consider currency]
- P[to EUR]
- B[click]
- P[to convert]
- B[click]
- R[show page with result]

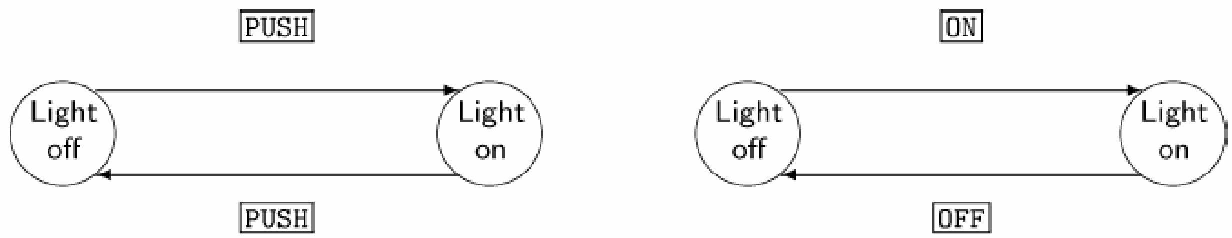
- 4*P = 4,40s
- 4*B = 0,80s
- 2*H = 0,80s
- 3*M = 4,05s
- 4*K = 1,12s
- 1*R = 1,00s

- Summe= 12,17s

Further reading
User Interface Design With Matrix Algebra
Harold Thimbleby

ACM Transactions on Computer-Human Interaction, Vol. 11, No. 2, June 2004, Pages 181–236.

- Algebra analysis of interactive systems
- Proving properties of interactive systems



a. Light with push-on/push-off action.   b. Light with separate on/off actions.

Finite state machines (FSMs)

<u>States as vectors</u>:
on (1 0)
off (0 1)

<u>Actions as Matrix</u>:

$$\boxed{\text{PUSH}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

<u>Press the button when off results in on</u>

$$\begin{aligned} \textbf{off}\,\boxed{\text{PUSH}} &= (0\ 1)\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ &= (1\ 0) \\ &= \textbf{on} \end{aligned}$$

<u>Press the button twice does not alter the state</u>

$$\begin{aligned} \boxed{\text{PUSH}}\,\boxed{\text{PUSH}} &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &= I \end{aligned}$$

Chapter 2
Basics of HCI and History

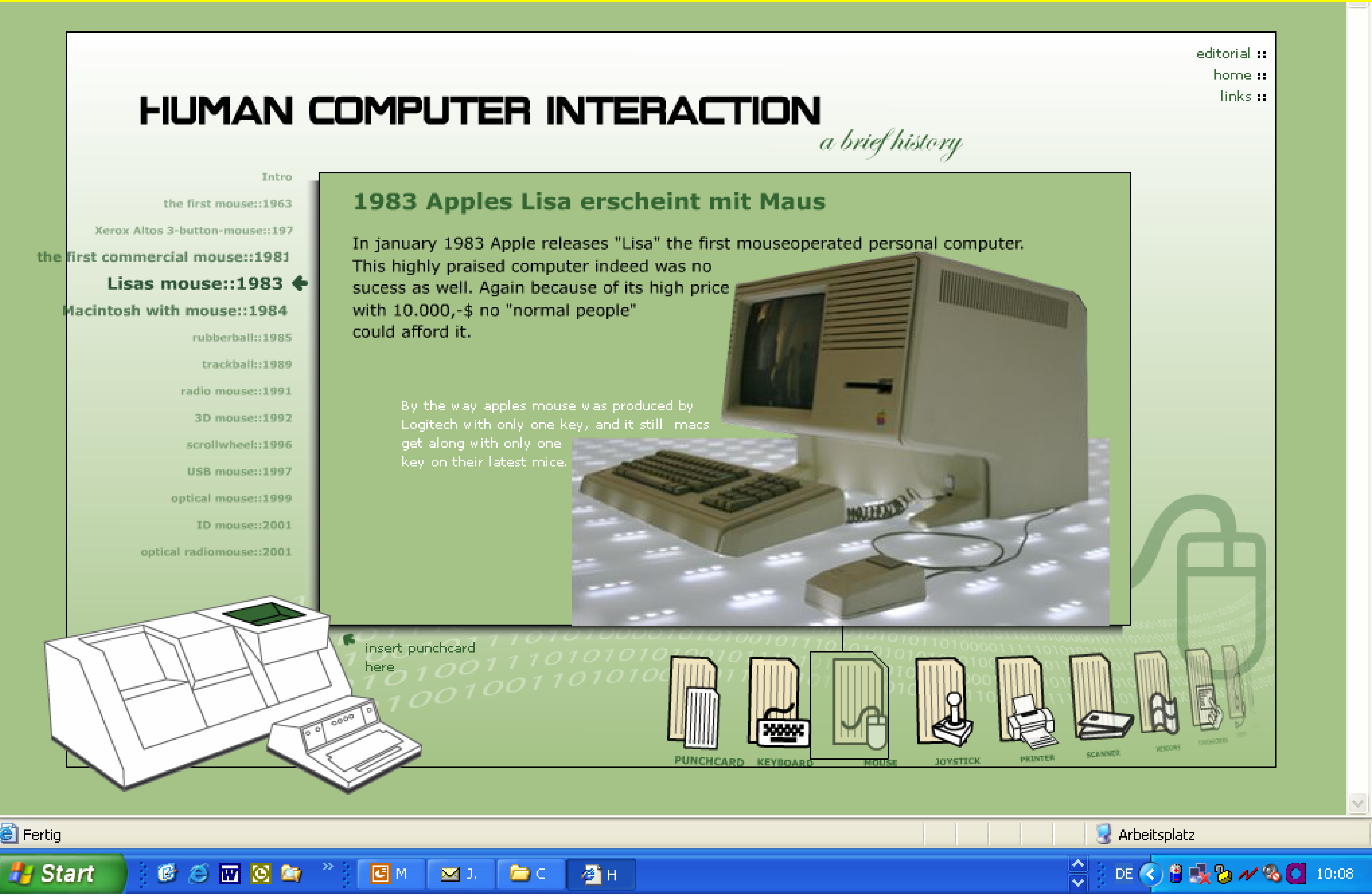
- 2.1 Motivation
- 2.2 Principles for UI-Design
- 2.3 Understanding Errors
- 2.4 Consistency
- 2.5 Basic Models
- **2.6 A Brief History of HCI**

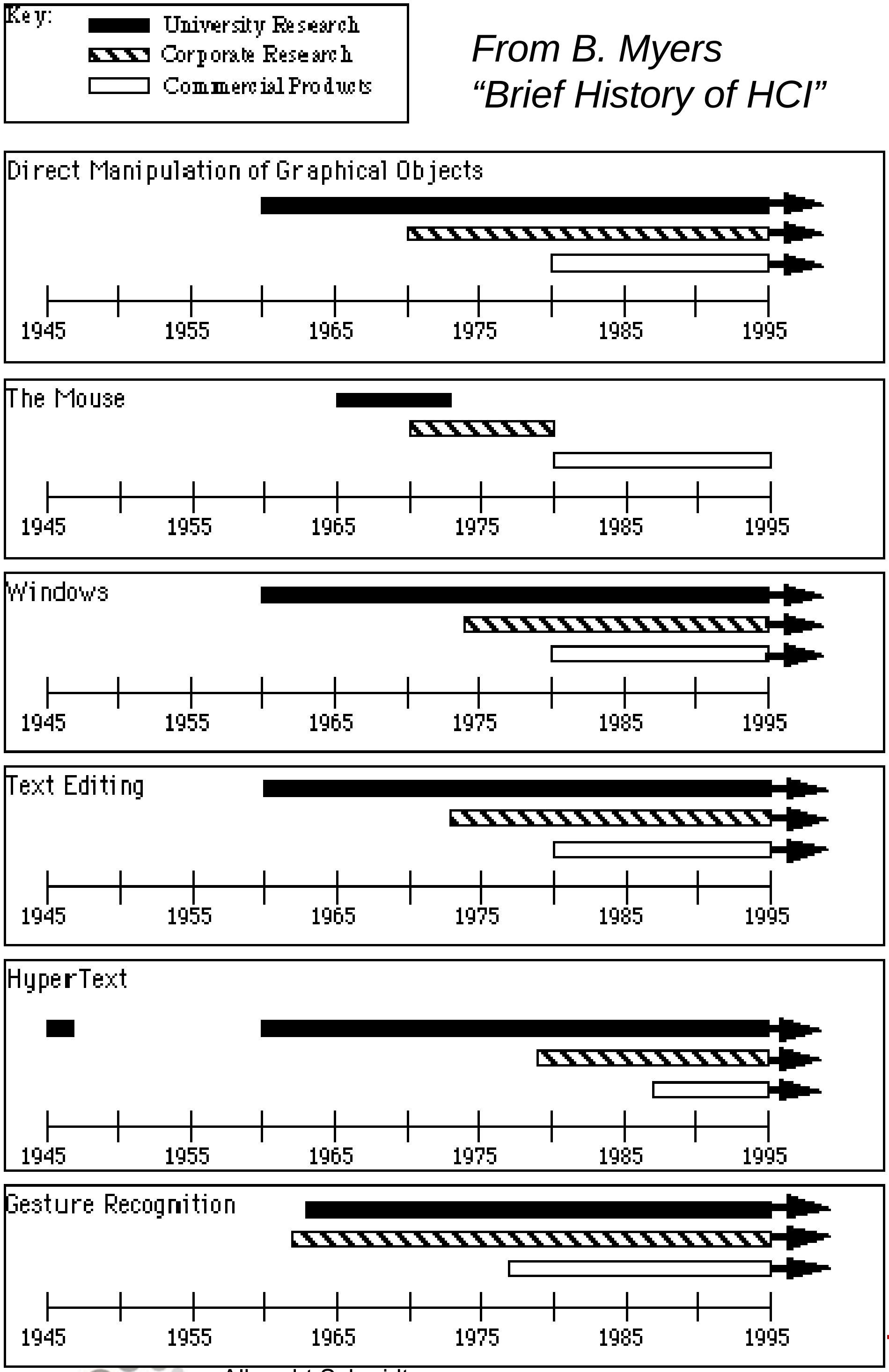
Evolution of HCI ‘interfaces’

- **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

Student Project

http://www.hcilab.org/projects/historybook/



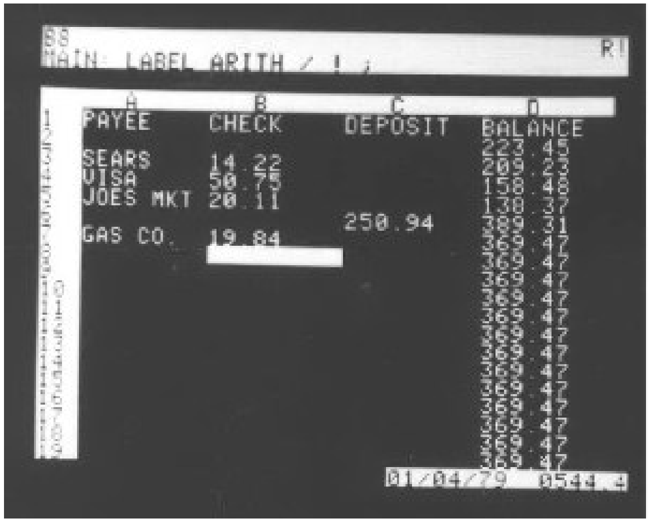


From B. Myers
"Brief History of HCI"

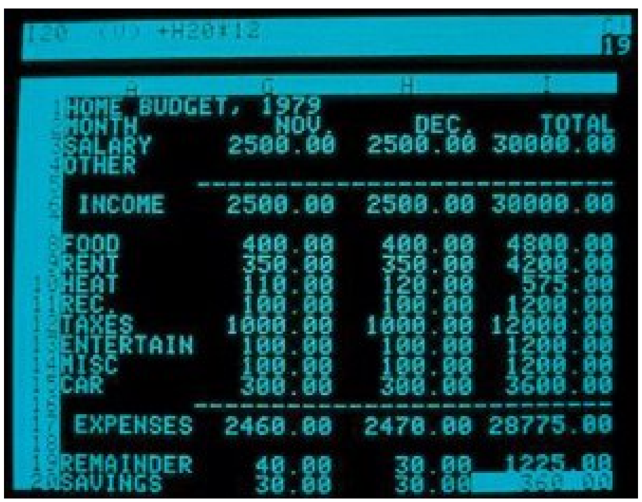
A Brief History of HCI

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

VisiCalc - Widespread use of an Interactive Application



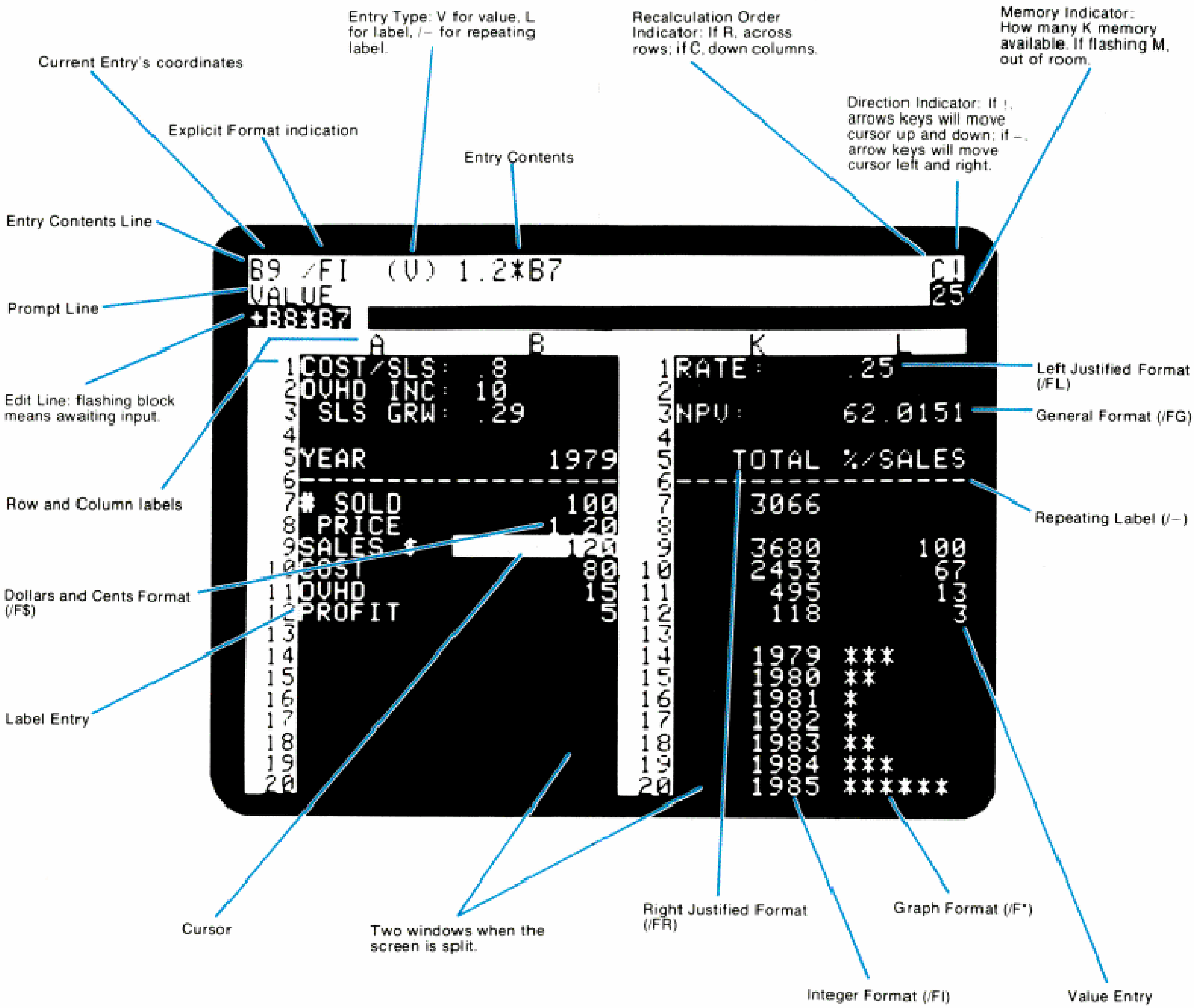
VisiCalc Screen, early Alpha 1/4/79



First version of VisiCalc screenshot

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

A VISICALC™ Screen:



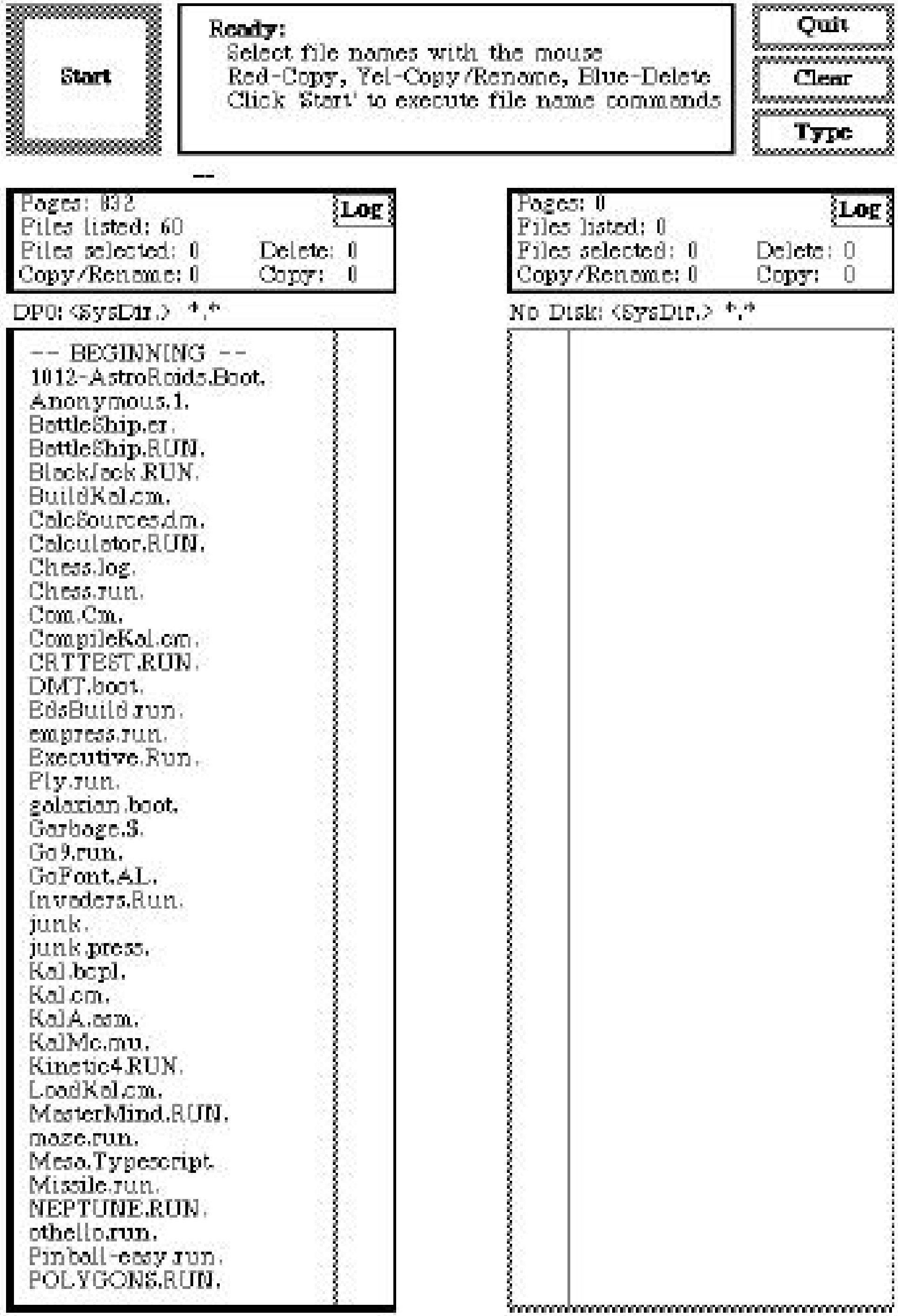
Changing Interaction Paradigms

- 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

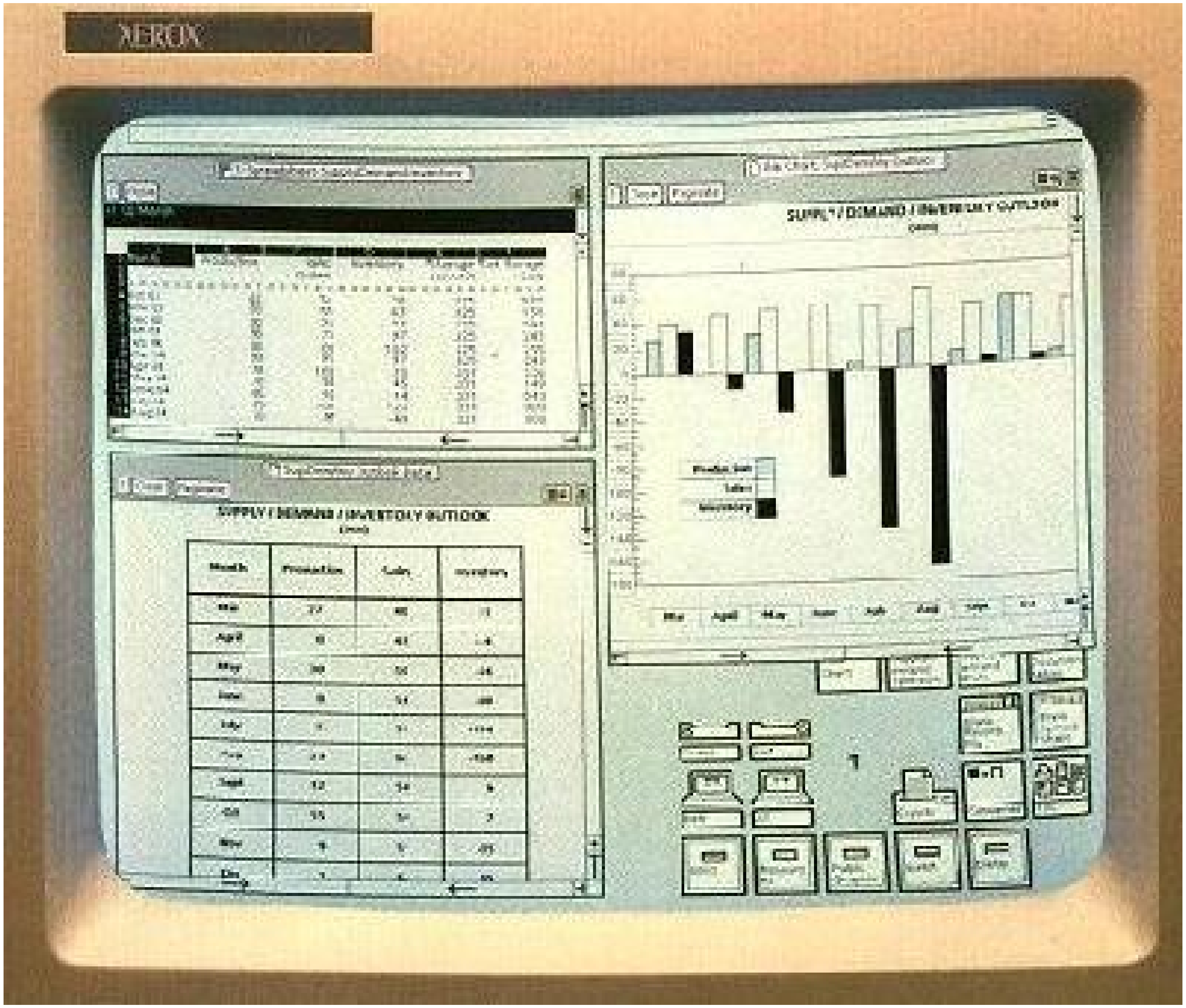
- Origins of direct manipulation an graphical user interfaces
  - Ivan Sutherland's Sketchpad, 1963, object manipulation with a light pen (grabbing, moving, resizing)
  - Douglas C. Engelbart, 1968, Mouse, NLS
  - XEROX ALTO (50 units at Universities in 1978)
  - XEROX Star (1981)
  - Apple Macintosh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

XEROX ALTO

Photos from
http://members.fortunecity.com/pcmuseum/alto.html

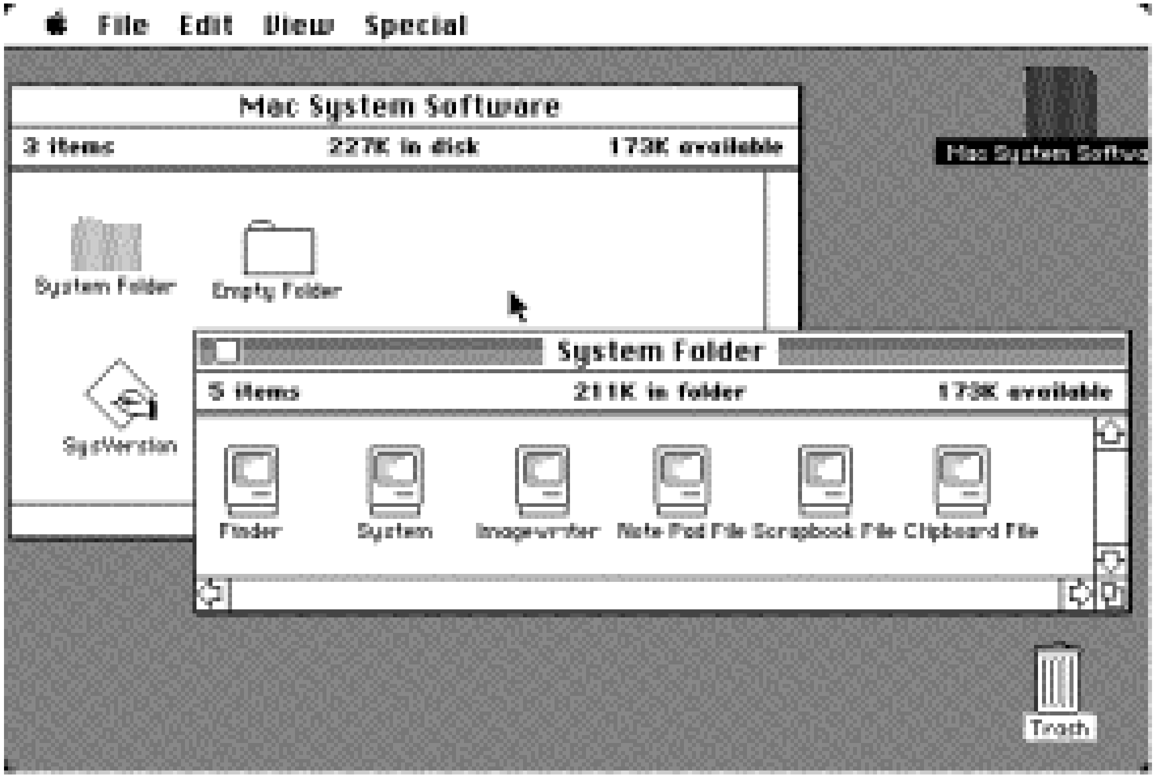


XEROX Star



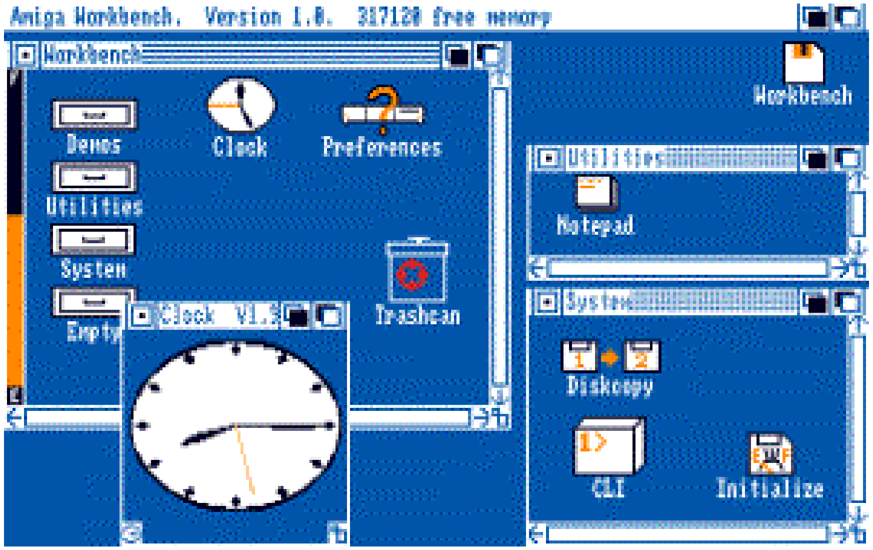
Photos from http://members.fortunecity.com/pcmuseum/alto.html

Apple Macintosh

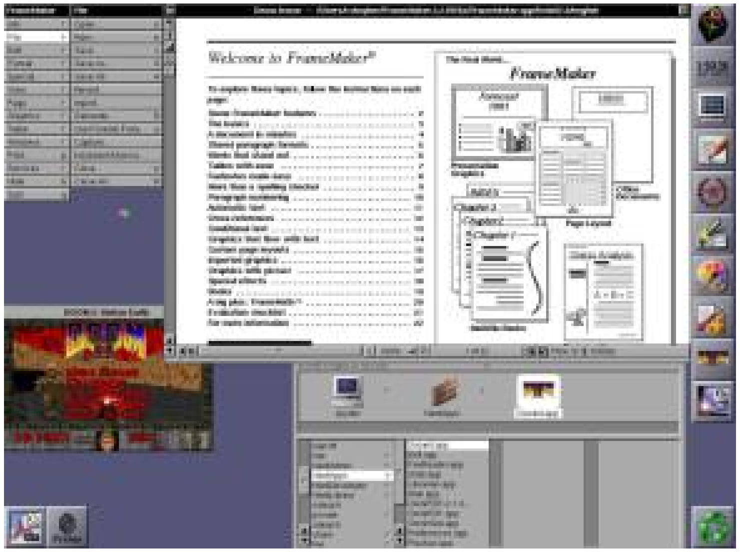


1984 – commercially successful GUI

More GUIs



Amiga 1985



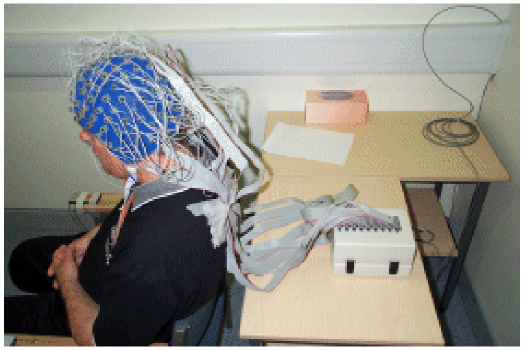
NextStep 1989

Win 3.11 1992

OS/2 1992

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 can not hide the user interface - good ideas spread out
- The first to come out with a new user interface is not necessarily the most successful

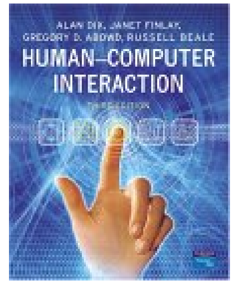


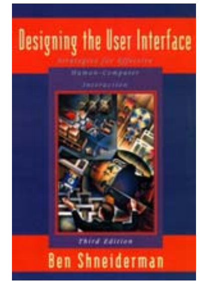
- Technologies to look out for?
  - Eye gaze detection
  - Speech and gesture recognition
  - EEG, ECG, EMG interfaces (e.g. http://www.biosemi.com/products.htm) ElectroEncephaloGraphy, ElectroCardioGraphy, ElectroMyoGraphy

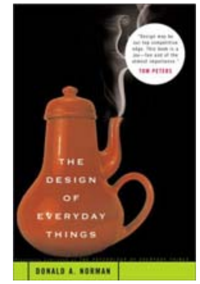
Brain Ball

http://smart.tii.se/smart/projects/brainball/index_en.html

References







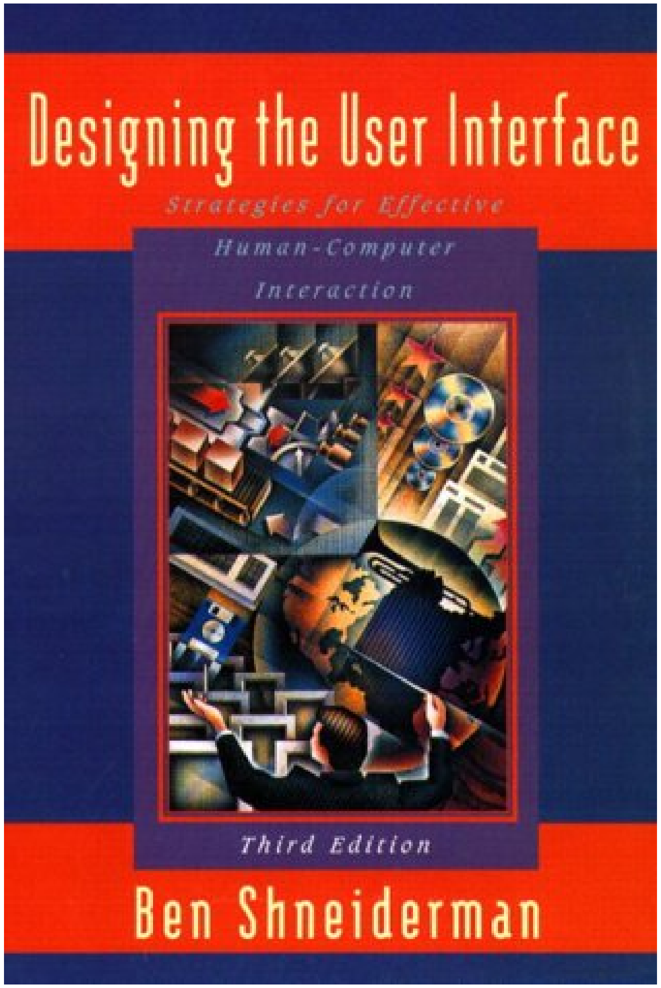
- D. A. Norman. The Design of Everyday Things. Basic Books 2002. ISBN: 0465067107
- B. Shneiderman. Designing the User Interface: Strategies for Effective Human-Computer Interaction , Third Edition. 1997. ISBN: 0201694972
- L. Suchman, Plans and situated action. 1990
- Alan Dix, Janet Finlay, Gregory Abowd and Russell Beale. (1998) Human Computer, Interaction (second edition), Prentice Hall, ISBN 0132398648 (new Edition announced for October 2003)
- Jef Raskin, The Humane Interface, ACM Press 2000
- Brad A. Myers. "A Brief History of Human Computer Interaction Technology." *ACM interactions*. Vol. 5, no. 2, March, 1998. pp. 44-54. http://www-2.cs.cmu.edu/~amulet/papers/uihistory.tr.html
- Software Arts and VisiCalc http://www.bricklin.com/history/intro.htm
- A. Cooper. About Face 2.0: Chapter 1 - Goal-Directed Design http://media.wiley.com/product_data/excerpt/13/07645264/0764526413.pdf

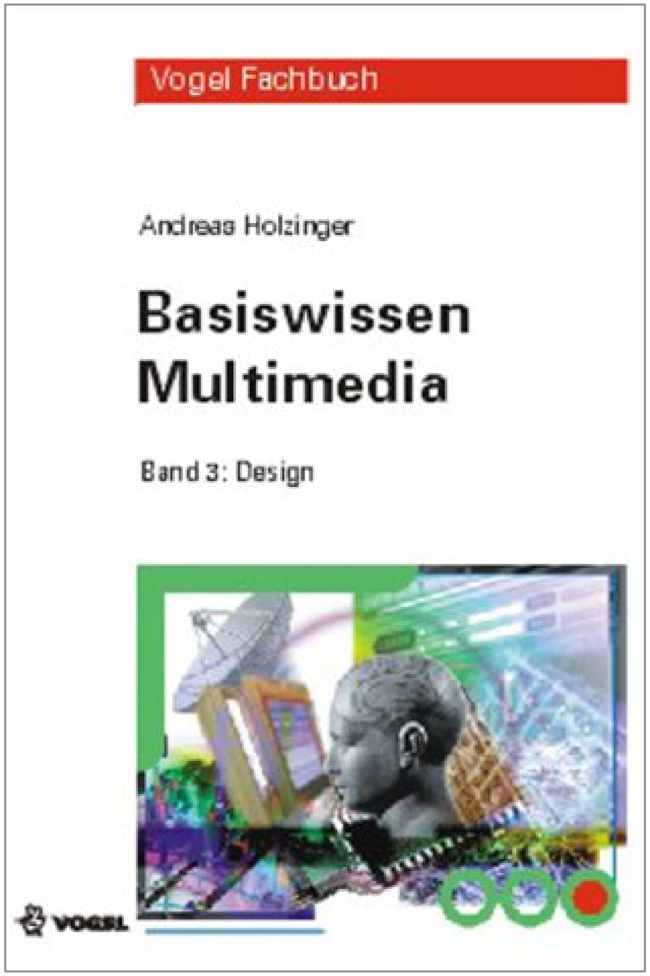
Meet the Authors

5. November 2005
16.00 Uhr
AudiMax der LMU

Medieninformatiktreffen and der LMU

Es sprechen
Ben Shneiderman
und **Andreas Holzinger**





- Ben Shneiderman. (1998) Designing the User Interface, 3rd Ed., Addison Wesley; ISBN: 0201694972
- Andreas Holzinger. (2001) Basiswissen Multimedia. Band 3: Design; ISBN: 3802318587

Ben Shneiderman in Munich Saturday, Nov. 5th 16-20Uhr

- Nassi-Shneiderman diagram (1972)
- Split menu (1992/1994)