Vorlesung
Mensch-Maschine-Interaktion

Models and Users (3)

Ludwig-Maximilians-Universität München
LFE Medieninformatik
Heinrich Hußmann & Albrecht Schmidt
WS2003/2004
http://www.medien.informatik.uni-muenchen.de/

Table of Content
Models and Users (3)

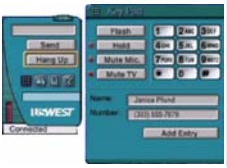
- Digital vs. conventional
- Creating a conceptual model
- Represented Model
- Four levels – Conceptual, Semantic, Syntactic, and Lexical
- Consistency and Inconsistency
- Object-Action Interface Model (AOI)
- Further Models

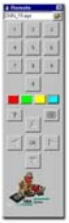
Metaphors and Digital Remakes of Conventional Products

- Limitations of the physical world vs. advantages of the digital
- Sticking close the original (conventional) product in your design may be tempting, but you probably will miss advantages provided by the digital solution
- Basic rule
  - Build on the knowledge that is available from the conventional product
  - Integrate novel concepts offered by the digital solution (short cuts)
- Applies to digital "remakes" and Metaphors

Metaphors and Digital Remakes of Conventional Products - Examples

- Paper forms
  - in a digital form it is possible to eliminate fields that are not applicable based on a previous choice
- Calendars
  - in paper there was one page for each month
  - this limitation is not present in electronic interfaces
  - so why not have additionally scrolling as a further different visualization view in the digital …





- Remote control metaphor
  - E.g. for a software music player
  - constrained of the physical devices – buttons, no display, …
  - why replicate in the digital?

Example Conceptual Model (1) Supporting a Traffic Warden

- Analyse Problem Space
  - Understand and analyse the problem space
  - Approach that leads to ideas
- Understand the User's Goals
  - What is the user (or are the users) trying to achieve
  - Understand the tasks involved
  - Relate the user's goals and tasks to the business model of the envisioned solution
- Which tasks can humans perform better than systems?
- What is the computer and Technology good at?
- What parts are error prone?
- What parts are boring/tedious/dangerous?
- What Technologies exist that can help?

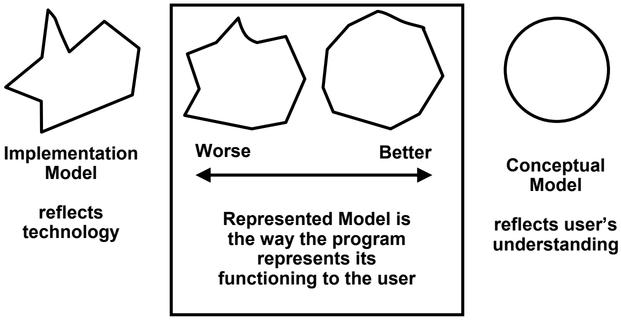
Example Conceptual Model (2) Supporting a Traffic Warden

- Make an explicit model
  - Identify explicitly the design options
  - Keep problem space and user's goals in mind
  - Make the conceptual model explicit (sketches, video, …)
- Activity based model / interaction mode
  - E.g. instructions, conversing, manipulating and navigating, exploring and browsing, or combination
- Interaction style:
  - E.g. command, speech, data-entry, form fill-in, query, graphical, web, pen, augmented reality, gesture, image capture
- Interaction metaphors
  - What objects have a meaning in the domain?
  - What activities are meaningful in the domain?
- Interaction paradigm
  - E.g. desktop, handheld, wearable, pervasive

Represented Model

- Chosen in the user interface design process
- The way in that the functionality of a system is presented to user
- “behavioural face”
- The represented model bridges the gulf between the implementation model and the user’s conceptual model
- The closer the represented model to the users conceptual model the easier is it for the user to operate
- The represented model however must deal with constraints from the implementation model (e.g. remote access, possible error conditions)

Implementation, Represented, Conceptual Model



Implementation Model

reflects technology

Worse ⟷ **Better**

Represented Model is the way the program represents its functioning to the user

Conceptual Model

reflects user’s understanding

From A. Cooper, About Face 2.0

Software is Often Close to the Implementation Model (1)

- If the UI is not designed but created on the fly as the software is implemented this will inevitably reflect the structure of the implementation, e.g.
  - Buttons to call functions
  - Dialog or Window for each module
  - Web page for each transaction step
- The resulting UI may still follow all guidelines, but logics and math (the thinking behind the implementation) is not widely known, e.g.
  - Boolean operators are used differently in computer science and natural language
  - Example: "give me all names of members in London **and** Manchester" → is a **OR** query in the database

Software is Often Close to the Implementation Model (2)

- Technical constraints are represented in the interface – often for no reason – and may have an influence on the metaphors used, e.g.
  - Local disk vs. remote disk
- Assumptions are made that need knowledge of the implementation model
  - Drag & drop in Windows
    on the same drive → move vs. on different drives → copy
  - Saving a file – why do I need to save a file? I have just written it!
  - USB memory – why do I have to stop the device before I remove it physically?

Bridging the gap between Conceptual and Implementation Model

- Educating the user about the implementation model
  - Traditional approach of training people to use a software system
  - In many cases there is no alternative
  - For new media applications education the user is difficult
  - In some cases it may be possible to educate the user “on the fly”
- Providing a represented Model that is close to the conceptual model
  - Knowingly using a design/representation that is not related to the implementation model
  - Creating systems that mediate between the conceptual and implementation model
- Design and model the user interface explicitly
- Record the mapping and relationship to the implementation

Four-level model (1)

- A way of thinking of different aspects of the interface
- Levels
  - Conceptual level
  - Semantic level
  - Syntactic level
  - Lexical level
- Designers are to work from top to bottom
- Mappings between levels are recorded

Four-level model (2)

- Conceptual level
  - The user’s mental model of the interactive system.
  - Examples
    - line editors vs. screen editors
    - Pixel based drawing vs. vector based drawing
- Semantic level
  - The meanings conveyed by the user’s input and by the computer’s output
  - Example
    - the meaning of the delete paragraph command
    - the meanings of the copy and paste commands

Four-level model (3)

- Syntactic level
  - How the units/words that convey the semantics are assembled into a term order to instruct the computer to perform a task
  - Example
    - the command format: first keyword type (ls), then parameter (/tmp)
      - `ls /tmp`
    - first the user selects the paragraph to copy, then issues the copy command, then selects the location for the paste operation, then issues the paste command
- Lexical level
  - The precise mechanisms with which the user specifies the syntactic level.
  - Example
    - Control-D means backspace
    - clicking within the form places the curser in the form
    - select an object by placing the cursor over the object and dragging across the object.

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)

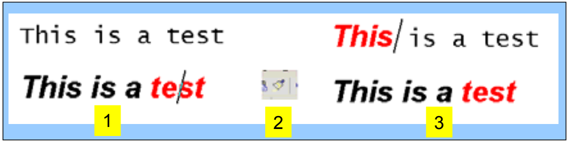
- Semantic Consistency
- Global commands always available
  - Help
  - Abort (command underway)
  - Undo (completed command)
- Operations valid on all reasonable objects
  - if object of class “X” can be deleted, so can object of class “Y”
- Applicability - roots
  - to command line user interfaces
  - Keyboard short cuts
  - Speech interfaces
- Applicability – additionally
  - 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

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



This is a test

This is a *test*

This is a test

This is a *test*

- 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 careful!
- 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

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

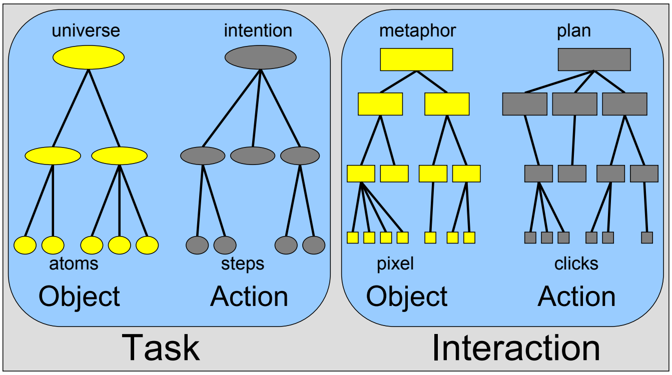
- Suchman, 1990
  - human plans are often not orderly executed
  - plans are often adapted or changed
  - user’s actions are situated in time and place
  - user’s actions are responsive to the environment
  - distributed cognition – knowledge is not just in the user’s head it is in the environment

Object-Action Interface Model (AOI)

- 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

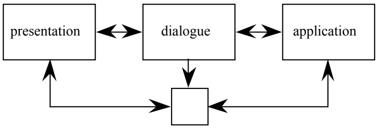
Object-Action Interface Model (AOI)



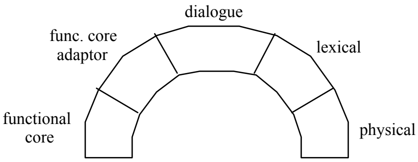
From Shneiderman

Further Models from User Interface Management

- Seeheim:

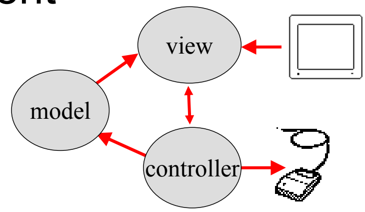


- Arch/Slinky



Further Models UI Development

- MVC
  - Model
  - View
  - Controller



- PAC
  - Presentation
  - Abstraction
  - Control

Implicit Interaction (1)

(Schmidt, et al., 2001)

- Implicit Human-Computer Interaction (iHCI)
  - iHCI is the interaction of a human with the environment and with artefacts which is aimed to accomplish a goal. Within this process the system acquires *implicit inputs* from the user and may present *implicit output* to the user.
- Implicit Input
  - Implicit input are actions and behaviour of humans, which are done to achieve a goal and are not primarily regarded as interaction with a computer, but captured, recognized and interpret by a computer system as input.

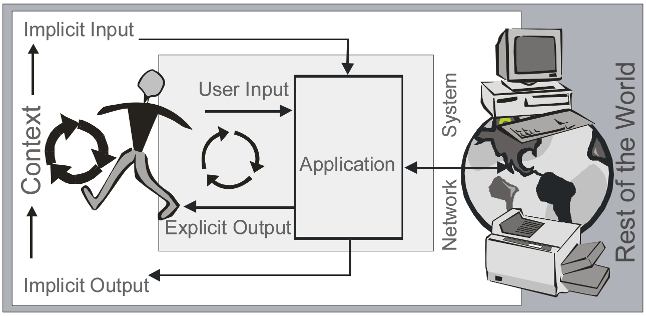
Implicit Output

- Output of a computer that is not directly related to an explicit input and which is seamlessly integrated with the environment and the task of the user.

Implicit Interaction (2)

(Schmidt, et al., 2001)

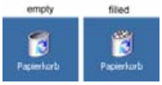
traditional explicit human computer interaction and implicit interaction with the context



References

- B. Shneiderman. Designing the User Interface: Strategies for Effective Human-Computer Interaction , Third Edition. 1997. ISBN: 0201694972
- Alan Cooper, Robert M. Reimann. (2003) About Face 2.0: The Essentials of Interaction Design; ISBN: 0764526413
- 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)
- Andreas Holzinger. (2001) Basiswissen Multimedia. Band 3: Design; ISBN: 3802318587
- Suchman, L.A. (1990). Plans and situated actions: the problem of human-machine communication. Cambridge, NY: Cambridge University Press.
- A. Schmidt, H.-W. Gellersen. Nutzung von Kontext in ubiquitären Informationssystemen. it+ti - Informationstechnik und technische Informatik, Sonderheft "Ubiquitous Computing - der allgegenwärtige Computer". 2001, Volume 43, Issue 02, p. 83-90. http://www.comp.lancs.ac.uk/~albrecht/pubs/pdf/gellersen_it_ti_02-2001.pdf

Exercise 2: Dynamic Icons (optional)



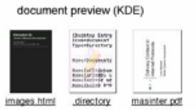
<u>Task A: Design a set of new folder icons</u>

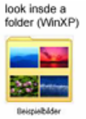
- Design a set of new folder icons that can convey information about:
  - disk size used by the folder
  - further folders contained in the folder
  - number of documents contained in the folder
  - security/privacy settings and ownership of the folder

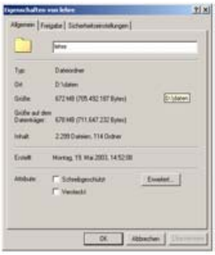
<u>Task B: How could you test the design?</u>

- Think how you could test your design.
  - What and when would it help the user?
  - What tasks could be sped up?
  - Which tasks could be easier?
  - What would be an appropriate way to do an experiment?
  - How would you implement the experiment?

- See web page for the exercise sheet







lehre

Typ: Dateiordner

Ort: D:\daten

Größe: 672 MB (705.492.187 Bytes)

Größe auf dem Datenträger: 678 MB (711.647.232 Bytes)

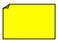
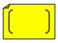
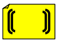
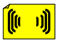
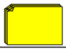
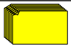
Inhalt: 2.299 Dateien, 114 Ordner

Erstellt: Montag, 19. Mai 2003, 14:52:08

Attribute: □ Schreibgeschützt □ Versteckt

Ideas for Exercise 2

- think of (better) icons that represent folders containing…

	<2 MB	2…200MB	200MB…2GB	>2 GB
<20 files				
20…200 files		…	…	…
> 200 files		…	…	…

- Ideas for Tasks
  - Search for a movie or other document
  - Remember structure of file
- Ideas for implementation
  - Photoshop and Javascript/Flash/Java
  - Implementing it using the real api calls (that is difficult!)