Vorlesung Mensch-Maschine-Interaktion

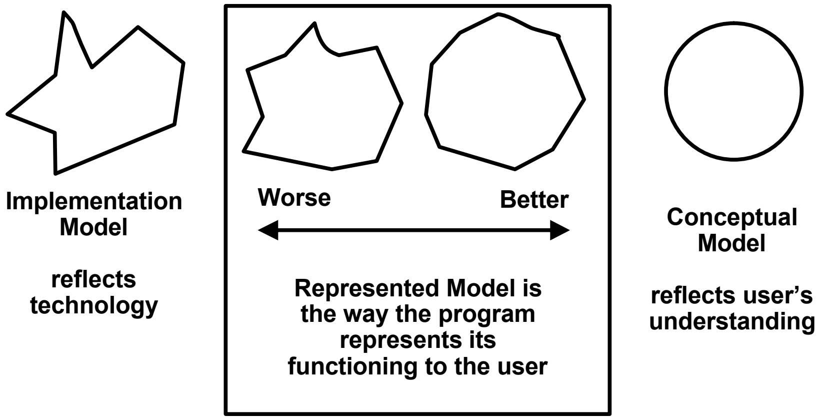
Models and Consistency

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

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

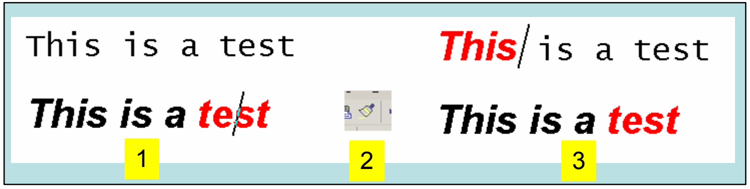
Consistency through Grammars

- 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



- 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

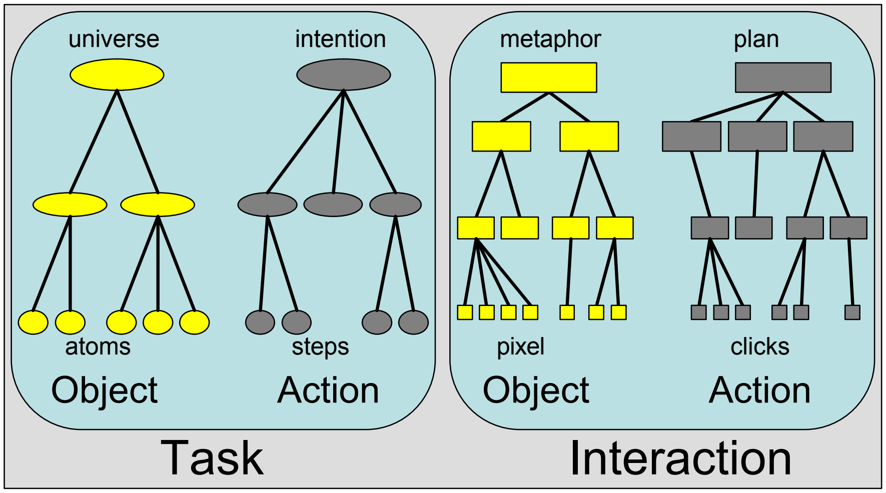
Plans and Situated Actions Distributed Cognition

- 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 (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

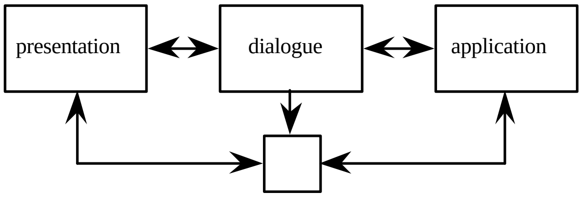
Object-Action Interface Model (OAI)



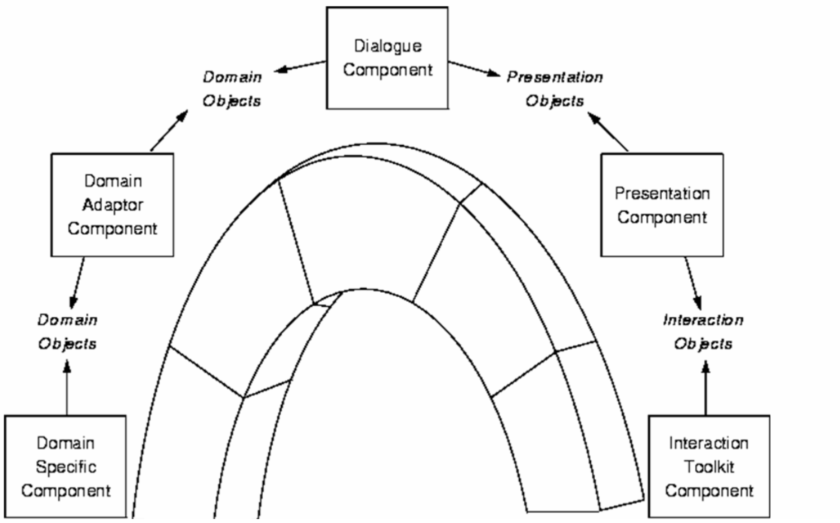
From Shneiderman

Seeheim Model

- Interface from the technical and designers point of view
- 3 basic components/layers with feedback
  - *presentation*, describing the visual interaction objects on a lexical level,
  - *dialogue*, describing the structural elements of the dialogue and the behaviour of the interaction objects on a syntactic level,
  - *application interface*, describing the purpose of the dialogue in the proper application context on a semantic level.
- Difficult to apply to complex GUIs, many extensions proposed



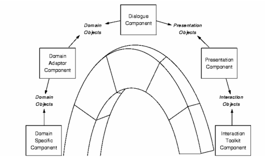
Slinky Metamodel / Arch Model (3)



From Miguel Encarnação

Slinky Metamodel / Arch Model (1)

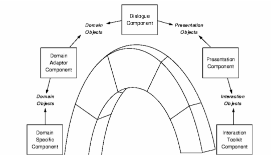
- Modelling data flow
- Minimizing effects of changing technology.
- Deriving system architectures from this metamodel
- Focus on User Interface Management Systems (UIMS)
- Functionalities supported
  - control and re-organization of domain data
  - execution of domain tasks
  - support of task sequencing
  - support of multiple view consistency
  - decisions on appropriate media
  - choice of interaction objects
  - support of physical interaction with the user
  - and conversion between domain formalisms and user-interface formalisms



Slinky Metamodel / Arch Model (2)

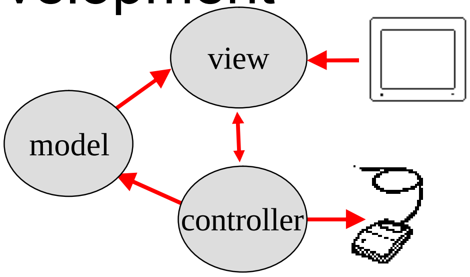
- **Components**
  - Domain-Specific Component controls
  - Interaction Toolkit Component
  - Dialogue Component
  - Presentation Component
  - Domain-Adaptor Component

- **Object** as abstraction for describing information between the components
  - Domain Objects
  - Presentation Objects
  - Interaction Objects



Further Models
UI Development

- MVC
  - Model
  - View
  - Controller



- Language/Protocols
  - XUP - Extensible User Interface Protocol
  - http://www.w3.org/TR/2002/NOTE-xup-20020528/

User Interfaces Beyond the Screen
Ideas and Models

- Tangible Bits
- Ambient Media
- Implicit Interaction

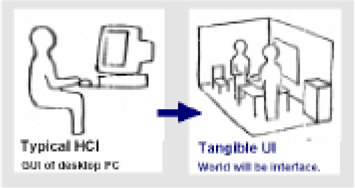
Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms

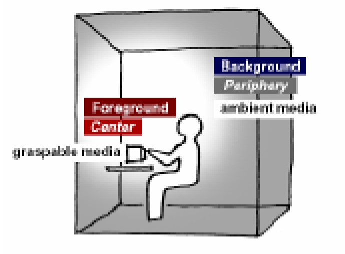
Hiroshi Ishii and Brygg Ullmer

- **Interactive Surfaces**
  Transformation of each surface within architectural space (e.g., walls, desktops,ceilings, doors, windows) into an active interface between the physical and virtual worlds

- **Coupling of Bits and Atoms**
  Seamless coupling of everyday graspable objects (e.g., cards, books, models) with the digital information that pertains to them;

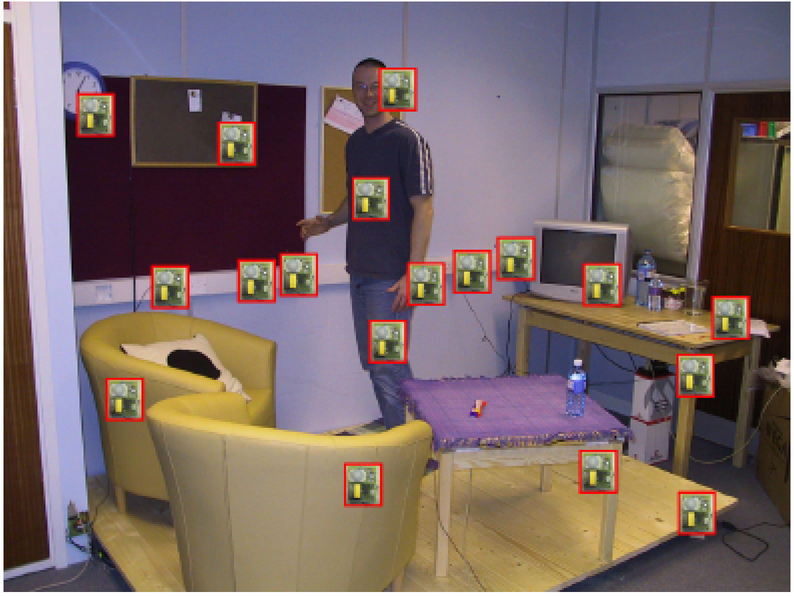
- **Ambient Media**
  Use of ambient media such as sound, light, airflow, and water movement for background interfaces with cyberspace at the periphery of human perception.





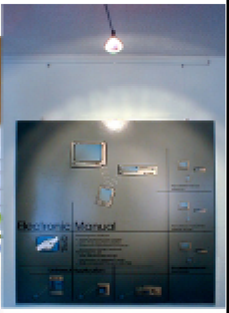
Prototyping the Environment as Interface

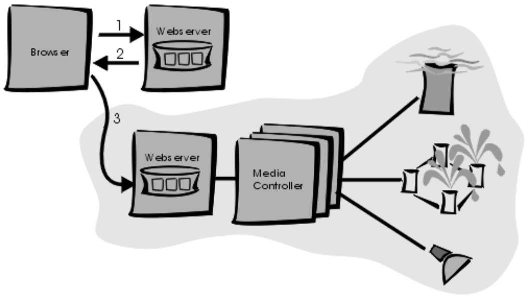
Prototyping the Environment as Interface



Ambient Displays

- Linking web resources with ambient media
- Providing peripheral information





Brows er

1

2

Webs erver

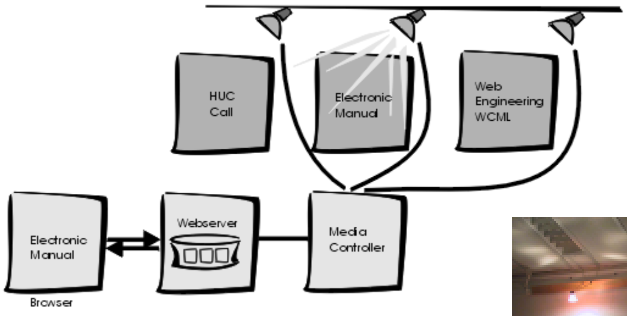
3

Webs erver

Media Controller

- Information is unobtrusive in the environment
- Information is ‘coded’

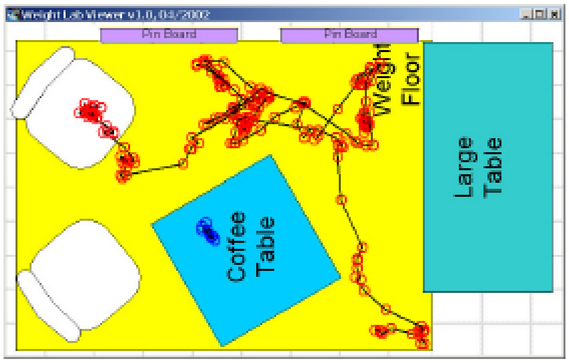
Ambient Displays



- …beyond notification
- Comparison of web access

- Effects: increased motivation and awareness

Load-Sensing as Input

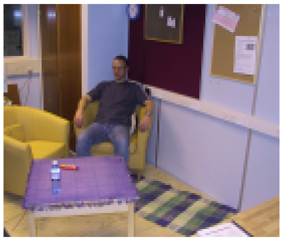


Weight Lab

- Lab environment with load-sensing floor, tables, and shelves
- Common furniture, unobtrusively augmented (wireless)

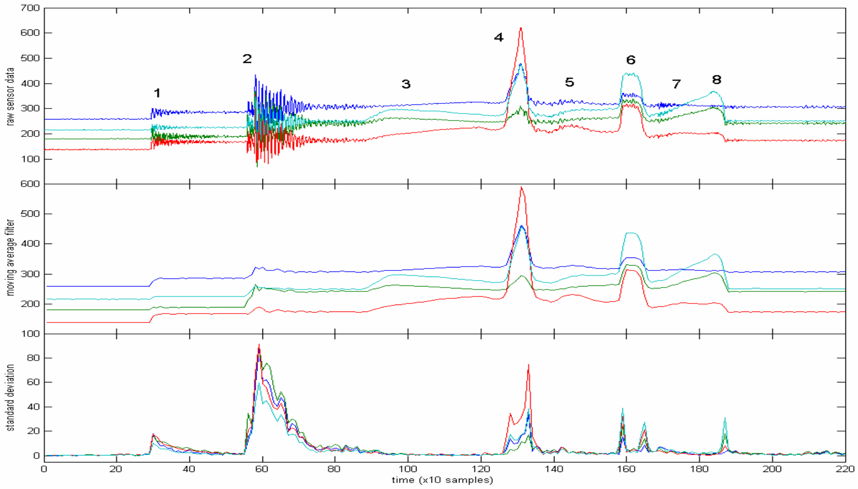
Context Acquistion

- Tracking of people, objects, activities
- In presence of noise (cluttered surfaces)



Load-Sensing Surface

Surfaces as Interaction Device



Implicit Interaction (1)

- 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.

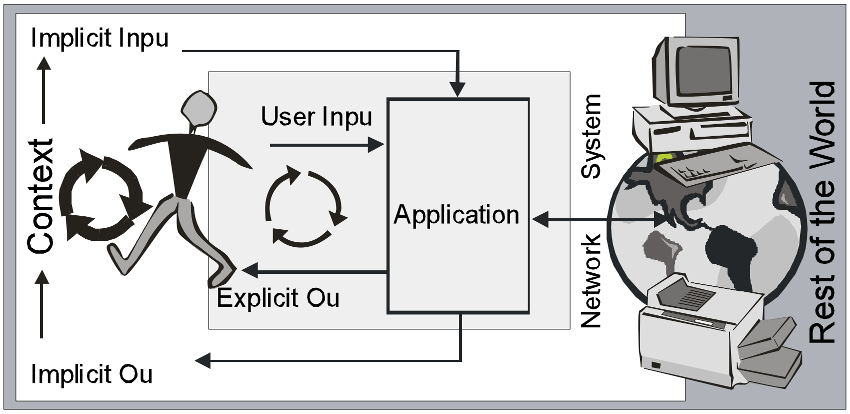
- Implicit Input
  - Implicit input are actions and behaviour of humans, which are done to achieve a goal and are not primarily regarded as interaction with a computer, but captured, recognized and interpret by a computer system as input.

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)

traditional explicit human computer interaction and implicit interaction with the context



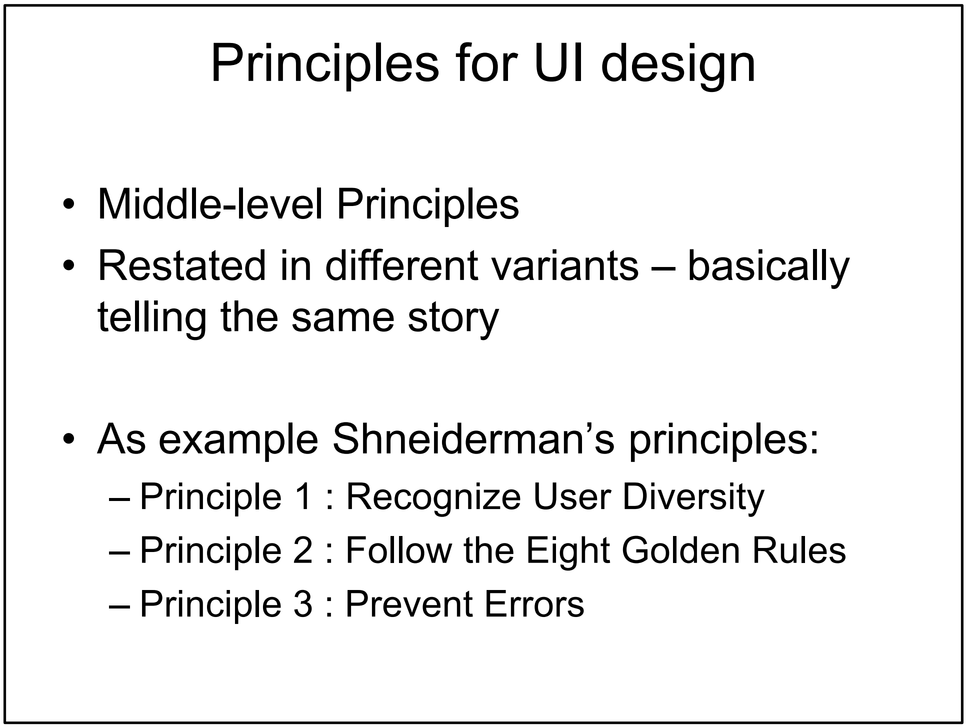
References

- 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

Vorlesung
Mensch-Maschine-Interaktion

Principles and Guidelines

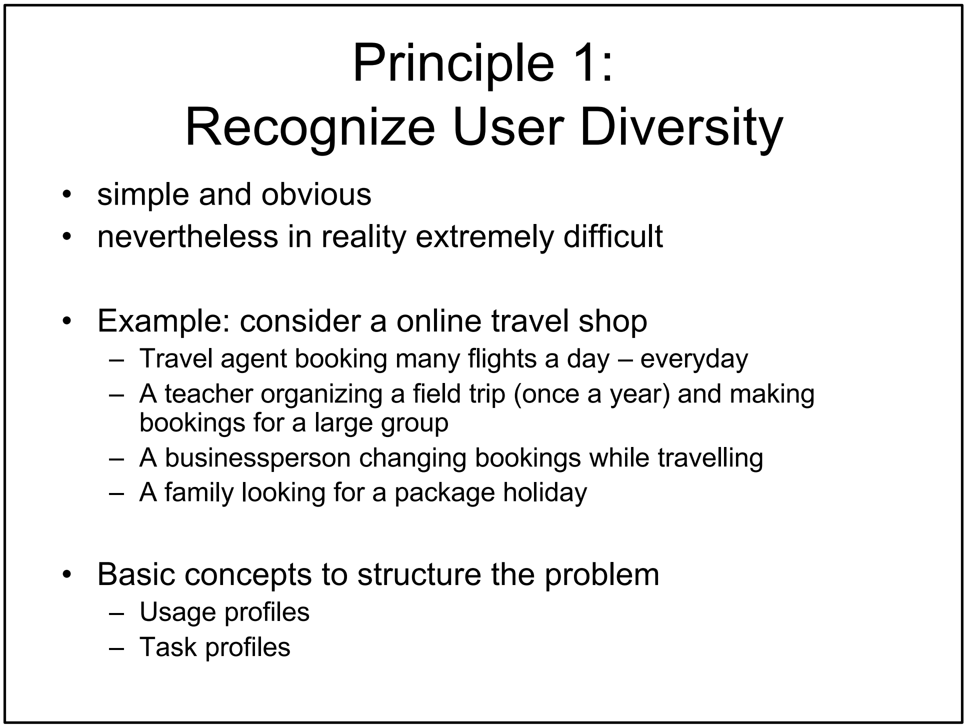
Ludwig-Maximilians-Universität München
LFE Medieninformatik
Heinrich Hußmann & Albrecht Schmidt
WS2003/2004
http://www.medien.informatik.uni-muenchen.de/



Principles for UI design

- Middle-level Principles
- Restated in different variants – basically telling the same story

- As example Shneiderman's principles:
  - Principle 1 : Recognize User Diversity
  - Principle 2 : Follow the Eight Golden Rules
  - Principle 3 : Prevent Errors

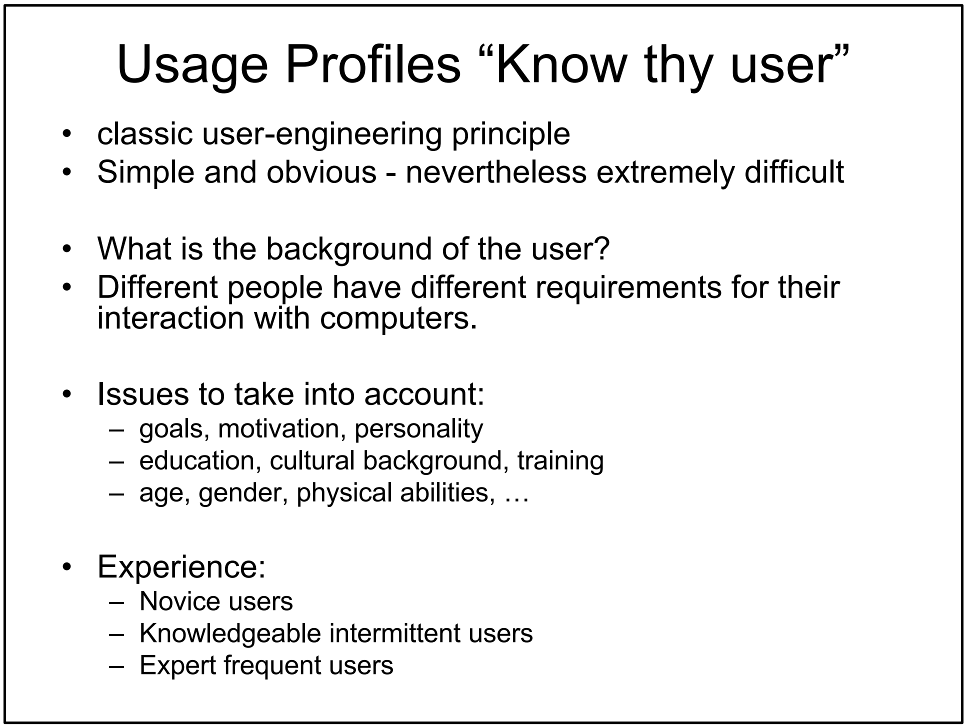


Principle 1: Recognize User Diversity

- simple and obvious
- nevertheless in reality extremely difficult

- Example: consider a online travel shop
  - 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 businessperson 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”

- classic user-engineering principle
- Simple and obvious - nevertheless extremely difficult

- 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

Frequency of Task by Job Title

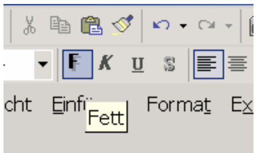
(hypothetical example from Shneiderman)

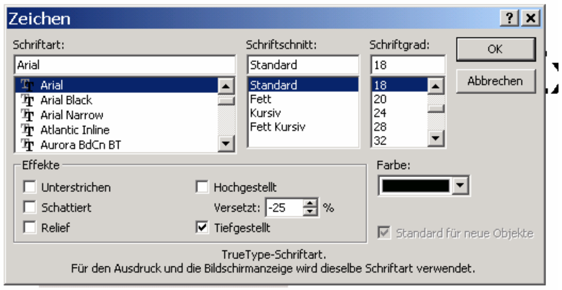
Task *Job*	Query by Patient	Update Data	Query across Patients	Change Database	Evaluate System
Nurse	0.14	0.11			
Supervisor	0.01	0.01	0.04		
Appointment Personnel	0.26				
Clinical researcher			0.08		
Database Programmer			0.02	0.02	0.05

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
  - Intermediately frequent actions: pull-down menu, key combination (Ctrl+S)
  - Infrequent actions: sequence of menus or dialogs
- Problem – when many (all) actions are with very similar relative frequency…

Task Frequency - Examples




Zeichen

Schriftart: Arial
Arial
Arial Black
Arial Narrow
Atlantic Inline
Aurora BdCn BT

Schriftschnitt: Standard
Standard
Fett
Kursiv
Fett Kursiv

Schriftgrad: 18
18
20
24
28
32

OK
Abbrechen

Effekte
□ Unterstrichen
□ Schattiert
□ Relief
□ Hochgestellt
Versetzt: -25 %
☑ Tiefgestellt

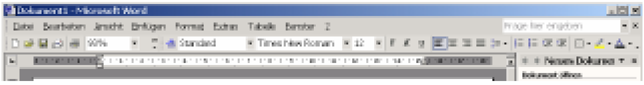
Farbe:

☑ Standard für neue Objekte

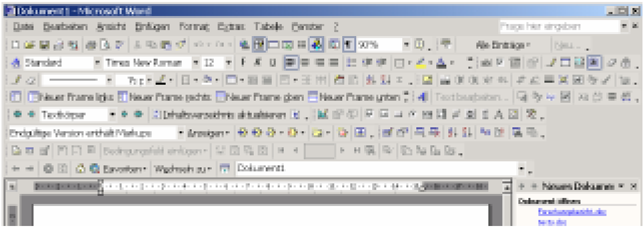
TrueType-Schriftart.
Für den Ausdruck und die Bildschirmanzeige wird dieselbe Schriftart verwendet.

- 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


Dokument1 - Microsoft Word
Datei Bearbeiten Ansicht Einfügen Format Extras Tabelle Fenster ?
Standard
Times New Roman
12

- 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


Dokument1 - Microsoft Word
Datei Bearbeiten Ansicht Einfügen Format Extras Tabelle Fenster ?
Standard
Times New Roman
12
Neuer Frame links
Neuer Frame rechts
Neuer Frame oben
Neuer Frame unten
Textkörper
Inhaltsverzeichnis aktualisieren
Endgültige Version enthält Markups
Anzeigen
Favoriten
Wechseln zu
Dokument1
Neues Dokument

References

- B. Shneiderman. Designing the User Interface: Strategies for Effective Human-Computer Interaction , Third Edition. 1997. ISBN: 0201694972
- L. Miguel Encarnação. Concept and realization of intelligent user support in interactive graphics applications. http://www.crcg.edu/company/staff/mencarna/publs/diss/node11.html
- Hiroshi Ishii and Brygg Ullmer. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms
  http://tangible.media.mit.edu/courses/ti02/ishii-chi97-tangbits.pdf
- H.-W. Gellersen, A. Schmidt. Look who's visiting: supporting visitor awareness in the web. http://www.comp.lancs.ac.uk/~albrecht/pubs/pdf/gellersen_ijhcs_2001.pdf
- 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)
- D. A. Norman. The Design of Everyday Things. Basic Books 2002. ISBN: 0465067107
- GNOME Human Interface Guidelines (1.0) by The GNOME Usability Project http://developer.gnome.org/projects/gup/hig/1.0/hig-1.0.pdf