Vorlesung
Mensch-Maschine-Interaktion

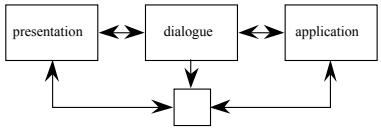
Models, Principles and Guidelines

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

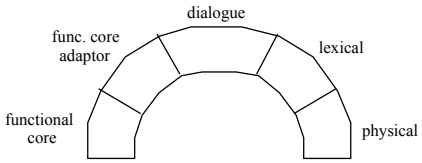
Table of Content
Models, Principles and Guidelines

Further Models from User Interface Management

- Seeheim:

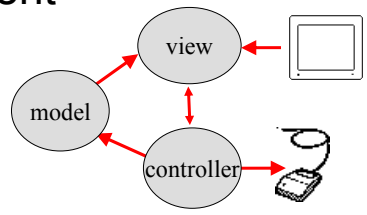


- Arch/Slinky



Further Models UI Development

- MVC
  - Model
  - View
  - Controller



- PAC
  - Presentation
  - Abstraction
  - Control

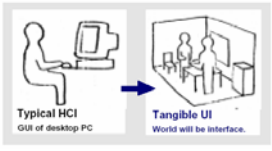
User Interfaces Beyond the Screen Ideas and Models

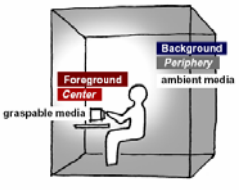
- Tangible Bits
- Ambient Media
- Implicit Interaction

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

Hiroshi Ishii and Brygg Ullmer

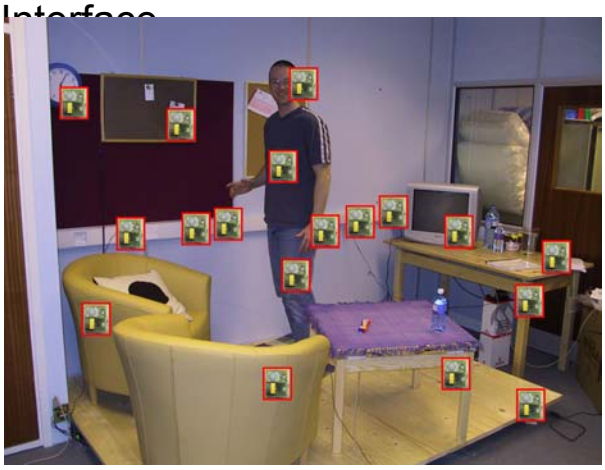
- **Interactive Surfaces**
  Transformation of each surface within architectural space (e.g., walls, desktops,ceilings, doors, windows) into an active interface between the physical and virtual worlds
- **Coupling of Bits and Atoms**
  Seamless coupling of everyday graspable objects (e.g., cards, books, 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.





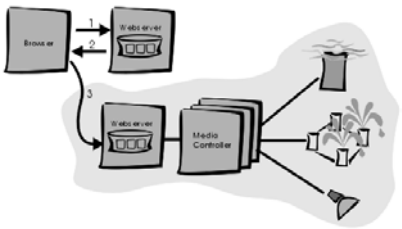
Prototyping the Environment as Interface

Prototyping the Environment as Interface



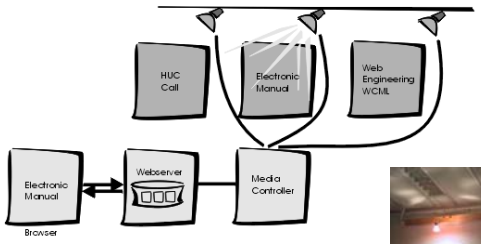
Ambient Displays

- Linking web resources with ambient media
- Providing peripheral information



- Information is unobtrusive in the environment
- Information is 'coded'

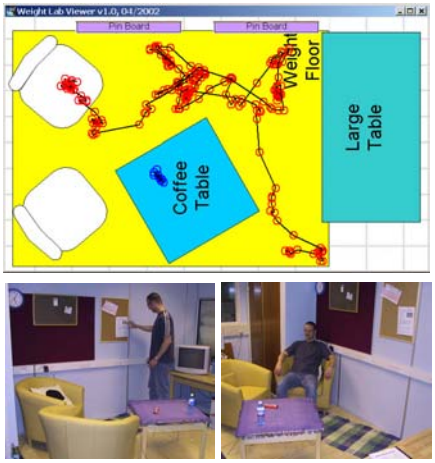
Ambient Displays



- …beyond notification
- Comparison of web access

- Effects: increased motivation and awareness

Load-Sensing as Input



Weight Lab

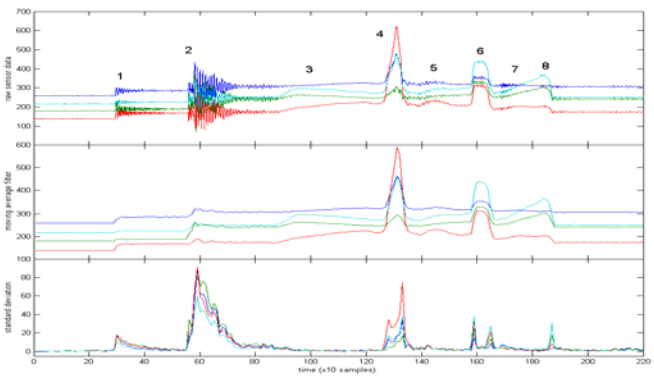
- Lab environment with load-sensing floor, tables, and shelves
- Common furniture, unobtrusively augmented (wireless)

Context Acquistion

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

Load-Sensing Surface

Surfaces as Interaction Device



Implicit Interaction (1)

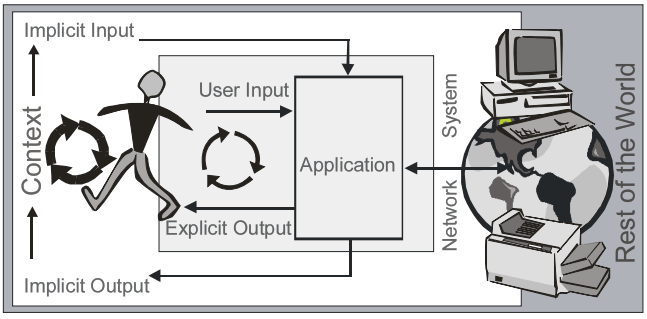
- Implicit Human-Computer Interaction (iHCI)
  - iHCI is the interaction of a human with the environment and with artefacts which is aimed to accomplish a goal. Within this process the system acquires *implicit inputs* from the user and may present *implicit output* to the user.
- Implicit Input
  - 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



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

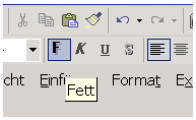
(hypothetical example from Shneiderman)

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



- 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

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

8 Golden Rules - Consistency

- Within an application it is with the developer (see last week…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…

Medieninformatik LMU München - [Translate this page]
Lehrveranstaltung Mensch-Maschine-Interaktion. springe zu den Volesungsunterlagen.
Wintersemester 2003/2004 Heinrich Hußmann, Albrecht ...
www.medien.informatik.uni-muenchen.de/ de/lehre/ws03/mmi/ - 44k - Cached - Similar pages

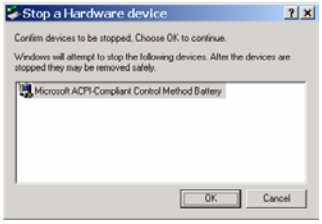
8 Golden Rules - Shortcuts

- 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

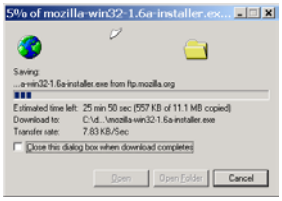
- For **any** action performed the user should have appropriate and informative feedback
- For frequent actions it should be modest, peripheral

- For infrequent action is should be more substantial



8 Golden Rules - Closure

- Sequences of actions should have a beginning, middle, and end.
- For non-instanteous 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
  - write to a file that already exists

- Different options how to handle it:
  - Involve the user (current practise)
  - Prevent the error or its consequences on system level (e.g. create a 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 acausality
- 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)

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

Understanding Errors (Norman)

- 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

Understanding the types of Slips Users Make (Norman)

- Capture errors
  - Two actions with common start point, the more familiar one captures the unusual (driving to work on Saturday instead to 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

Confirmation is unlikely to prevent Errors (Norman)

- 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
- A solution is to make the action reversible

Hix and Hartson’s guidelines

1. User centered design
2. Know the user
3. Involve the user
4. Prevent user errors
5. Optimize user operation
6. Keep control with the user
7. Help the user to get started
8. Give a task-based mental model
9. Be consistent
10. Keep it simple
11. Design for memory limitations
12. Use recognition rather recall
13. Use cognitive directness
14. Draw on real world analogies

Hix and Hartson guidelines (2)

15. Use informative feedback
16. Give status indicators
17. Use user-centred wording
18. Use non-threatening wording
19. Use specific constructive advice
20. Make the system take the blame
21. Do not anthropomorphise

- Use modes cautiously
- Make user action reversible
- Get attention judiciously
- Maintain display inertia
- Organize screen to manage complexity
- Accommodate individual difference

(Hix and Hartson, Developing User Interfaces, Wiley, 1993)

Specific Guidelines for Operating Systems, Window Managers, and the WWW Some Examples:

- Introduction to the Apple Human Interface Guidelines
  http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/index.html
- KDE User Interface Guidelines
  http://developer.kde.org/documentation/design/ui/
  http://developer.kde.org/documentation/standards/kde/style/basics/
- Palm OS® User Interface Guidelines
  http://www.palmos.com/dev/support/docs/ui/UIGuide_Front.html
- MSDN - User Interface Design and Development
  http://msdn.microsoft.com
- GNOME Human Interface Guidelines (1.1 - DRAFT)
  http://developer.gnome.org/projects/gup/hig/draft_hig_new/
- Web Guidelines???
  … many!

GNOME Guideline

- 1. Usability Principles
  - Design for People
  - Don't Limit Your User Base
  - Accessibility
  - Internationalization and Localization
  - Create a Match Between Your Application and the Real World
  - Make Your Application Consistent
  - Keep the User Informed
  - Keep It Simple and Pretty
  - Put the User in Control
  - Forgive the User
  - Provide Direct Manipulation
- 2. Desktop Integration
  - Placing Entries in the Applications Menu
  - Menu Item Names
  - …
- 3. Windows
  - Titles
  - …
  - Layout
  - Common Dialogs
- 4. Menus
  - The Menubar
  - Types of Menu
  - Drop-down Menus
  - …
  - Help
- 5. Toolbars
  - Appearance and Content
  - …
- 6. Controls
  - …
  - Sliders
  - Buttons
  - Check Boxes
  - …

GNOME Guideline

- 7. Feedback
  - Characteristics of Responsive Applications
  - Acceptable Response Times
  - Responding to User Requests
  - Types of Visual Feedback
  - …
- 8. Visual Design
  - Color
  - Palette
  - Hue, Brightness, Contrast
  - …
- 9. Testing Icons
  - …
  - Kinds of Icons
  - Document Icons
  - Application Icons
  - Toolbar Icons
  - …
- 10. User Input
  - Mouse Interaction
  - Buttons
  - Selecting Objects
  - Drag and Drop
  - …
- 11. Language
  - …
- 12. Checklists
  - Things You Can Do Yourself
  - Before You Start
  - …
- …

Workshop: Benutzerschnittstellen und Bedienkonzepte für Leseschwache

- Gemeinsam mit Sonderpädagogik
- 8. Dezember 2003, 18 Uhr c.t.
- Amalienstr. 17, 506 (oder 105-106)
- Domain Analyse & Prototyping
- Beispiel E-Mailanwendung & WWW Portal

References

- B. Shneiderman. Designing the User Interface: Strategies for Effective Human-Computer Interaction , Third Edition. 1997. ISBN: 0201694972
- 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