Vorlesung
Mensch-Maschine-Interaktion

Limitations for the Design of Interactive Systems

Ludwig-Maximilians-Universität München
LFE Medieninformatik
Andreas Butz & Albrecht Schmidt
WS2004/2005
http://www.medien.informatik.uni-muenchen.de/

Limitations on Interactive System

especially relevant for systems beyond the desktop

- Computation bound
  - Time to complete a computation, systems keeps the user waiting
- Storage bound
  - Limitations to the amount of data that can be stored
- Storage channel bound
  - transferring data between different types of memory (RAM – harddrive)
- Graphics bound
  - Limitations to displaying information
- Media capture bound
  - Limitations to for acquiring information / capturing
- Network bound
  - Access to networked resources

Computation

- Unlimited processing for applications?
  - Standard office task on a PC (e.g. writing email) – processor is often idle
- Not really…
  - Media intensive applications (e.g. video editing)
  - Scientific computing (e.g. simulations)
  - Mobile devices (e.g. mobile phone)
  - Embedded systems (e.g. heating controller)
- Strategies for interactive applications
  - Give interactive tasks high priority
  - Do calculations/processing before the user asks for them (while the system is idle) – this is difficult as it is often hard to predict what the user wants…
  - If computational results are not provided immediately indicate the duration and an option to pause or stop
  - Don't block the systems while doing computations (e.g. allow the user to interact while long term computations are done)

Storage

- Storage bound
  - Nearly unlimited storage for stationary systems available
  - More difficult for mobile systems, especially for media capture (e.g. a digital video camera)
- Storage channel bound
  - Transfer speed differs, RAM, Harddrive, DVD, DVD-RW, network, …
- Strategies for interactive systems
  - Make use of the storage available
    - Speed up interaction (e.g. store multiple index to a database, store search queries)
    - To allow reversing user action (even over multiple sessions)
    - Free users short term and long term memory (store what the user told the software once – only ask again if there is reason)
  - Minimize transfer between different storage types
  - Don't cache data that is written to a device that can be removed at any time (e.g. USB memory)

Use of storage – Example I

- Basic concept
  - Ticket machine for public transport
  - User pays by cash-card / credit card
  - System provides short cuts to last travels done
- Storage
  - Data can be stored locally in the ticket machine – as the last travel is particular meaningful in this context
  - Minimal amount of data – id (computed from the card) and a list of travels (start-point, end-point, type of ticket)

Use of storage – future speculation

- Digital VCR
- Much effort goes by now into how the user selects what should be stored!
- User is probably more interested what to watch…
- As more storage becomes available (and multiple receiver are cheap enough) the system can record everything and the interface is only concerned with selecting what to watch?

- 24h x 50 channels x 7 days = 8400 hours of Video
- How long will it take? … 10 years?
- Will TV change before that?

Graphics Bound

- No problem for office applications
- Even multiple screens are no problem

- Gaming and Entertainment
  - Graphics and rendering are a limiting factor
  - High resolution video (digital cinema)

- CAD
  - Resolution and screen size a limiting factors

- Mobile devices
  - Inherent trade-off between device size and screen area
  - Rendering performance a limiting factor (e.g. mobile phone)

- Strategies
  - Use the maximal display size available in the context
  - Use graphics hardware

Network Bound

- Different types of networks
  - Local wired networks (e.g. 1GBit/s)
  - Local wireless networks (e.g. WiFi 54MBit/s, Bluetooth < 1MBit/s)
  - Global wired network (Internet)
  - “global” wireless networks (small bandwidth, e.g. GSM, UMTS)

- Issues for interactive applications
  - Bandwidth, throughput
  - Jitter
  - Delay
  - Reliability

- Strategies
  - Design the system and interaction to fit the underlying network, e.g.
    - Unreliable network → allow offline use
    - Low bandwidth network → minimize data that is transferred (compression)
    - Network with long delay → keep interaction local
  - some network short comings can be compensated by storage use

Vorlesung
Mensch-Maschine-Interaktion

Principles and Guidelines

Ludwig-Maximilians-Universität München
LFE Medieninformatik
Andreas Butz & Albrecht Schmidt
WS2004/2005
http://www.medien.informatik.uni-muenchen.de/

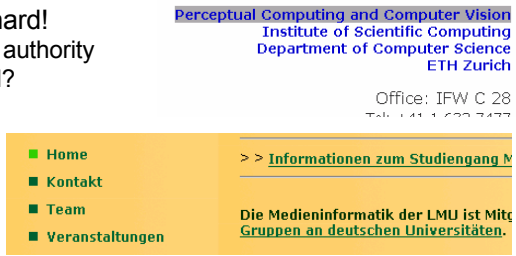
Principle 2:
Follow the 8 Golden Rules

- 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 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…



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

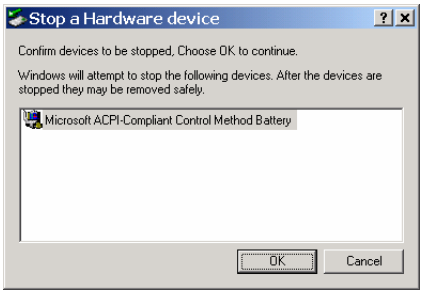
8 Golden Rules - Shortcuts

- 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

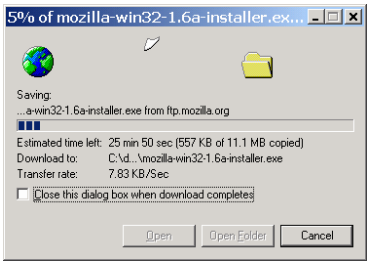
- For **any** action performed the user should have appropriate and informative feedback
- For frequent actions it should be modest, peripheral

- For infrequent action is should be more substantial



8 Golden Rules - Closure

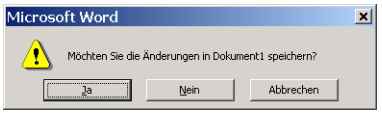
- Sequences of actions should have a beginning, middle, and end.
- For non-instanteous actions
- On different levels –
  - E.g. in the large: Web shop - it should be clear when I am in the shop, and when I have successfully check-out
  - E.g. in the small: a progress bar



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)

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
  - …

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 (examples in the PDF page 44,56,65,131,183,194)
- 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???
  http://www.webstyleguide.com/ … and many others!

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

Vorlesung
Mensch-Maschine-Interaktion

Process, Methods & Tools

Ludwig-Maximilians-Universität München
LFE Medieninformatik
Andreas Butz & Albrecht Schmidt
WS2004/2005
http://www.medien.informatik.uni-muenchen.de/

Table of Content

Development Process

Separation between interaction design and technical design

- For interactive applications a separation into a two stage process is often advisable
- 1st – Interaction design (iterative)
  - concept
  - Interaction analysis
  - Prototypes
  - Evaluation
  - Stable and tested design
- 2nd – technical realization
  - Technical analysis
  - Technical specification (e.g. architecture, platform)
  - Implementation
  - Evaluation and Quality management

Development Process

Logical User Centered Interactive development Methodology (LUCID)
http://www.cognetics.com/lucid/index.html

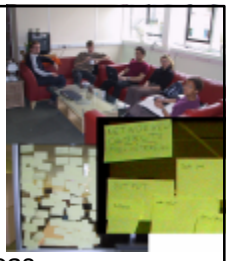
- Stage 1: **Envision**
  - Develop UI Roadmap which defines the product concept, rationale, constraints and design objectives.
- Stage 2: **Analyze**
  - Analyze the user needs and develop requirements.
- Stage 3: **Design**
  - Create a design concept and implement a key screen prototype.
- Stage 4: **Refine**
  - Test the prototype for design problems and iteratively refine and expand the design.
- Stage 5: **Implement**
  - Support implementation of the product making late stage design changes where required. Develop user support components.
- Stage 6: **Support**
  - Provide roll-out support as the product is deployed and gather data for next version.

Brainstorming Sessions I

- Collect as many ideas/issues as possible
- Rules
  - During brainstorming NO criticism is allowed
  - Developers must not say “this can’t be implemented”
  - Graphics designers are not to comment on drawing styles
- Do a selection in a second step

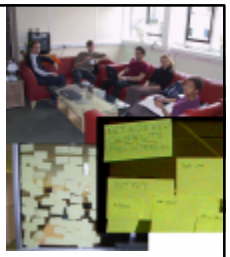
(Pin&Play Meeting, July 2002, Lancaster)

Brainstorming Sessions II



- Some hints
  - Get a mixed set of people (developer, manager, admin, writer, students, sales, customer)
  - Allow people to have freaky / crazy / unrealistic ideas
  - Use low technology (e.g. paper, pens, post-its, posters)
  - Do not allow to fetch / lookup additional material during the session
  - Go to a neutral / different / inspiring place (e.g. meeting room in another building, meeting room in a hotel at the Starnberger See, a hut in the mountains)
- If you get stuck?
  - Ignore boundaries – assume there is a little magic available
  - Assume there is a human brain insight
  - Get another person to help (e.g. get another person and explain where you are stuck)
  - Go for a walk

Brainstorming Sessions III



- Organize the ideas
  - Involve everybody
  - Identify concepts and themes
  - Group ideas that express the same concept or belong to a common theme
  - Identify conflicting ideas
  - Identify parallel ideas
  - Identify ideas that exclude each other
- Document the results!!!
  - Capture the raw material (usually you won't need it but it is no effort…)
  - Extract the design/product concepts
  - In the best case you have several competing concepts that can evaluated

Ethnographic Observation in HCI

- Traditional ethnographers immerse into other cultures over a extended period (weeks, month, years) and thereby study and understand the culture
- Ethnographic observations in HCI are a means of data collection
- Usually observing potential users (typical users) over a period of hours, days, or weeks. Include critical times (e.g. shift change)
- Goal
  - Acquire information that is required to create user interfaces and interaction mechanisms suitable
- Risk
  - Misinterpretation of observations (often due to a luck of insight)
  - Changing peoples behavior, disrupt processes
  - Overlooking / missing important facts
- Some problems occur infrequently – if you can not observe them conduct interviews

Guidelines for Ethnographic Observation in HCI (Shneiderman, chapter 3)

- Preparation
  - Understand the current system in the context of the organization and culture – don't be ignorant!
  - Describe the goals of the observation and prepare questions
  - Get permissions for observations and interviews
- Field Study
  - Establish contact, talk to people
  - Observe, interview, and collected data in situ
  - Document observations
- Analysis
  - Compile data, summaries and quantify
  - Provide interpretation of the data
  - Refine the goals and record issues about the process
- Reporting
  - Describe findings – possibly for different audiences

Ethnographic Observation in HCI Video Observation

- Capture work practices on video (consider legal and ethical issues)
- Different view points simultaneously
  - Camera overlooking the workplace
  - Camera looking from the screen to the user
  - Camera capturing what the user sees (e.g. camera mounted to glasses)
- User's view often provides significant insight
- Asking user's to talk (to describe) while doing a task provides generally a lot of useful information
- Raw material alone is of little value – need for analysis
- Analyzing video observations is hard and time consuming!
- Can be very useful
  - Multiple people interact (and observation of an individual and the whole group is of interest)
  - for tasks that are done very quickly or hard to observe
  - where observation is not possible (e.g. for safety or security reasons)
- Users may not like it! If they agree a person observing them they still may disagree to be videoed