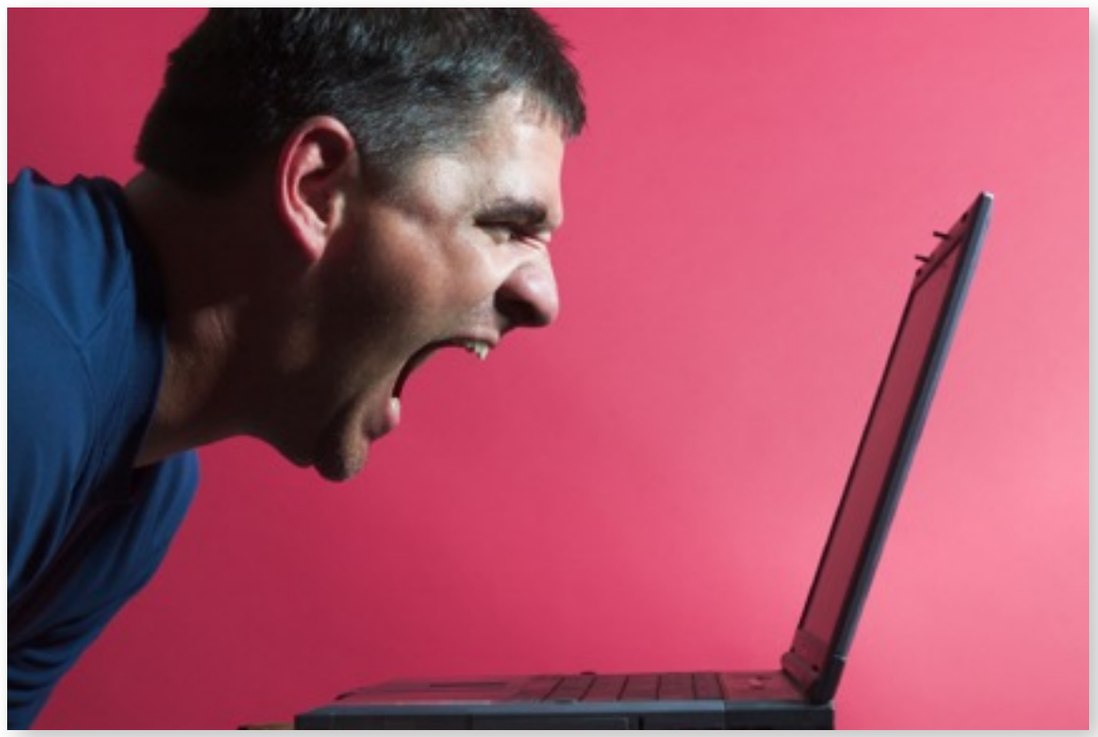
Human-computer interaction: usability & cognition

Christian Kaiser



http://www.thedigitalconsultant.co.uk/blog/wp-content/uploads/man-angry-at-computer.jpg

Interactive systems

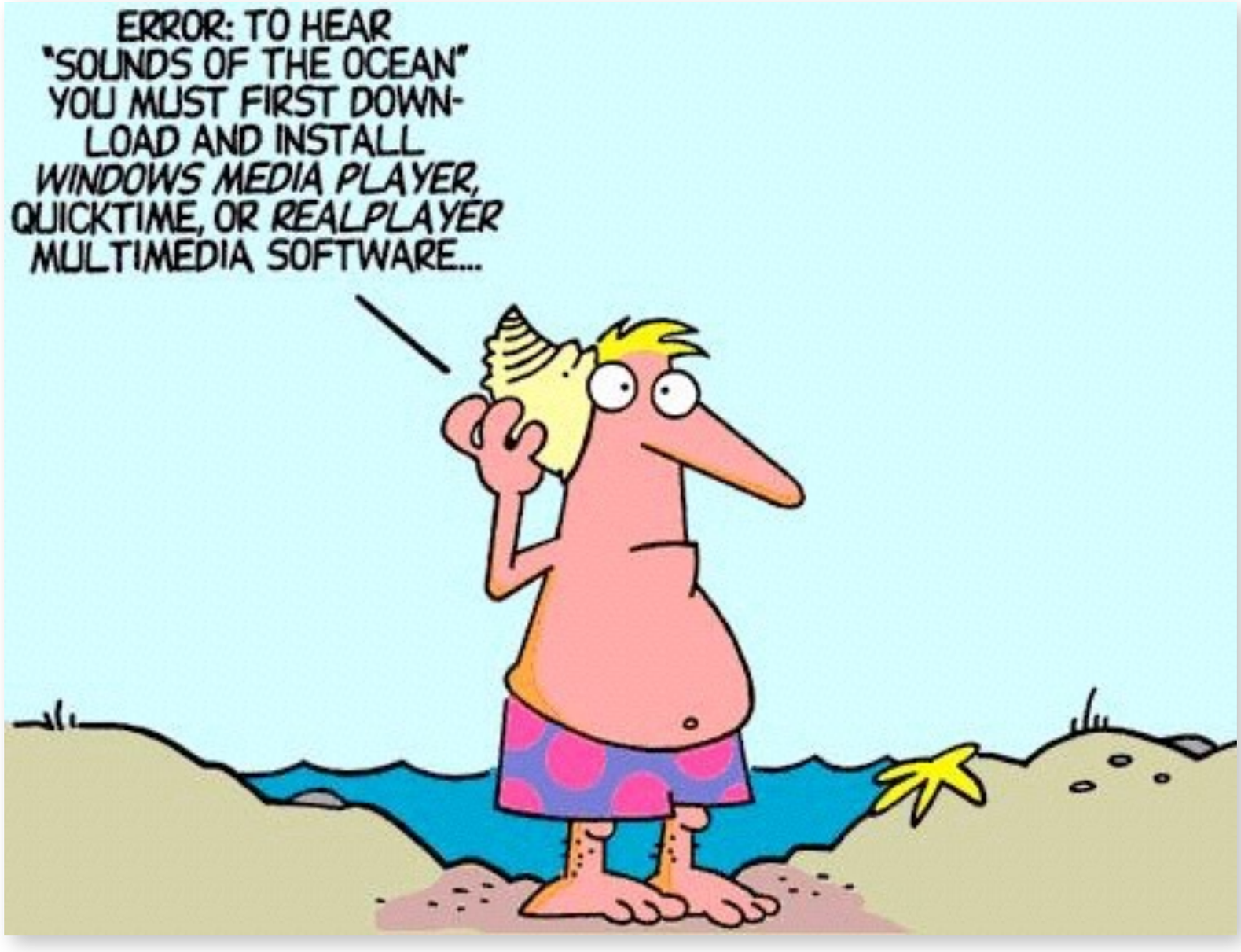
- Technological system requiring interaction with users in order to accomplish the task it has been designed for
- Human Computer Interaction (**HCI**) focuses on how to best design interactive systems
  - Focus **is not** on adding more and more **features**
  - Focus is on what people **can actually do** with the technology
- **Ease of use** is important

Useful, Usable and Accessible

- An **efficient system** is useful, usable and accessible.
- **Useful:** the user **can actually achieve** the task he wants to do. The system supports the user objectives and purpose.
  - If the user wants to write a text and uses Microsoft Word, then Microsoft Word is *useful*.
- **Usable**: the user **achieves a task easily**.
  - If the user wants to buy a train ticket for the first time at the machine and is able to do so without any difficulty, then the ticket machine was *usable*.
- **Accessible:** the system can be used by the full range of intended users.
  - Google gives a simple, very interactive user design that everybody can use. Google is in this case *accessible*.

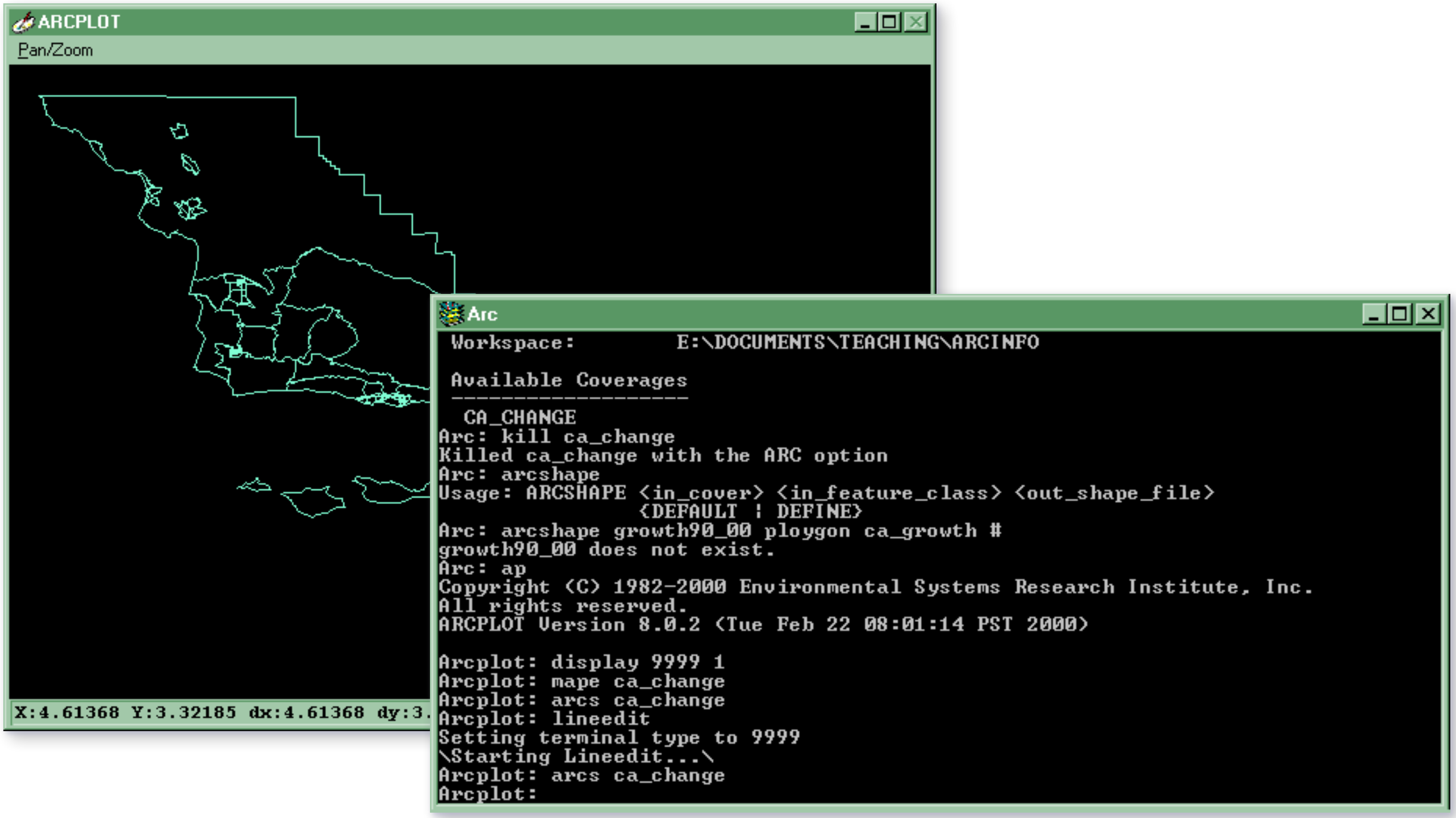
Example…

- Useful?
- Usable?
- Accessible?

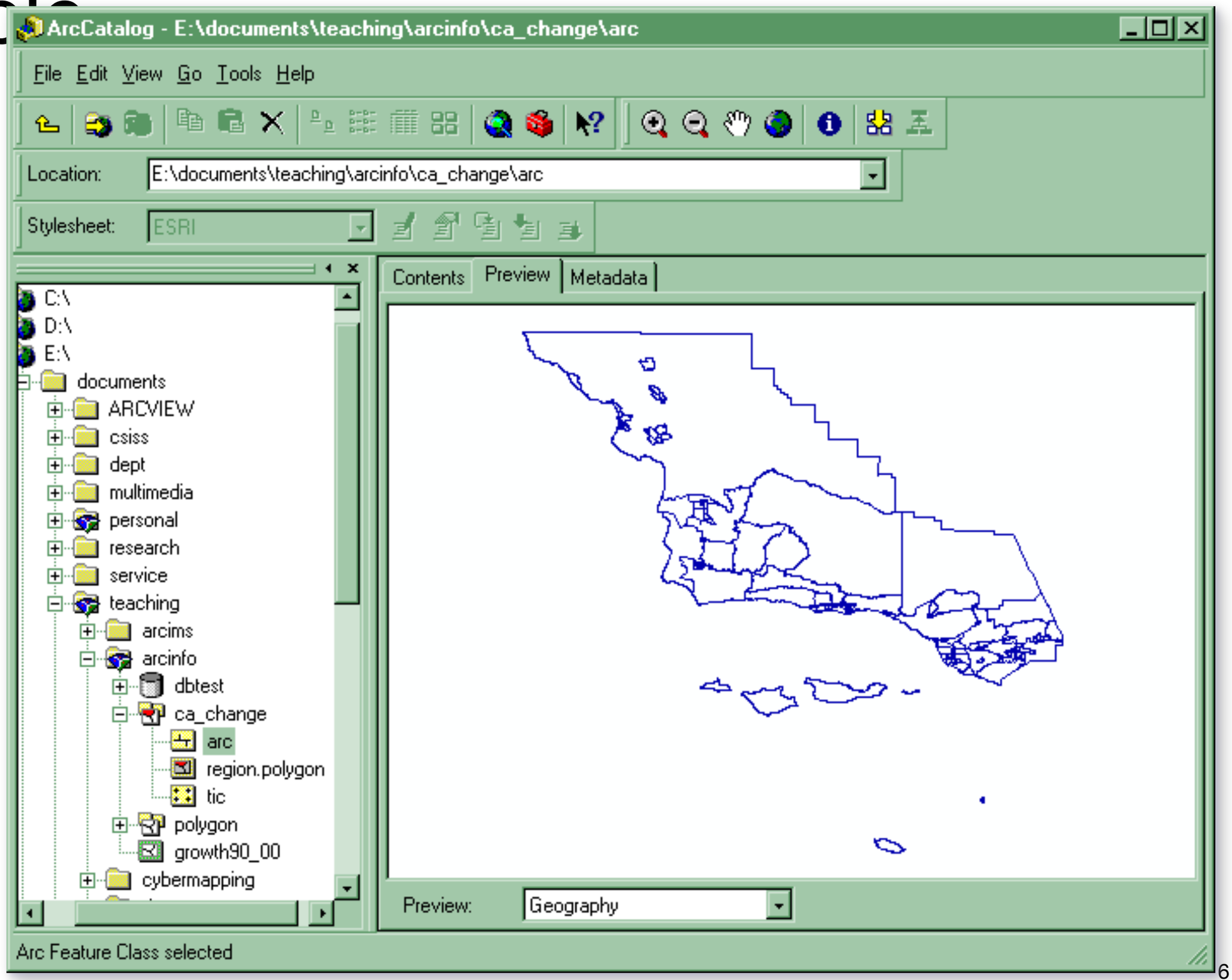


http://2.bp.blogspot.com/-b37meWYHqgM/Th-mvLvB13I/AAAAAAAAAKQ/B6badRWewf0/s1600/hci.JPG

Example…


ARCPLOT
Pan/Zoom
X:4.61368 Y:3.32185 dx:4.61368 dy:3.
Arc
Workspace: E:\DOCUMENTS\TEACHING\ARCINFO
Available Coverages
CA_CHANGE
Arc: kill ca_change
Killed ca_change with the ARC option
Arc: arcshape
Usage: ARCSHAPE <in_cover> <in_feature_class> <out_shape_file>
<DEFAULT | DEFINE>
Arc: arcshape growth90_00 ploygon ca_growth #
growth90_00 does not exist.
Arc: ap
Copyright (C) 1982-2000 Environmental Systems Research Institute, Inc.
All rights reserved.
ARCPLOT Version 8.0.2 (Tue Feb 22 08:01:14 PST 2000)
Arcplot: display 9999 1
Arcplot: mape ca_change
Arcplot: arcs ca_change
Arcplot: lineedit
Setting terminal type to 9999
\Starting Lineedit...\
Arcplot: arcs ca_change
Arcplot:

Example

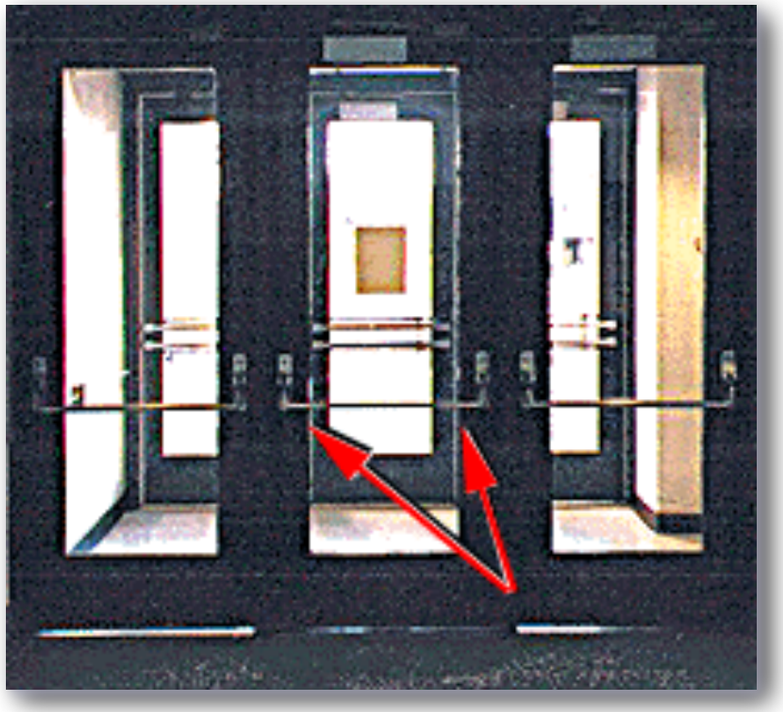

ArcCatalog - E:\documents\teaching\arcinfo\ca_change\arc
File Edit View Go Tools Help
Location: E:\documents\teaching\arcinfo\ca_change\arc
Stylesheet: ESRI
C:\
D:\
E:\
documents
ARCVIEW
csiss
dept
multimedia
personal
research
service
teaching
arcims
arcinfo
dbtest
ca_change
arc
region.polygon
tic
polygon
growth90_00
cybermapping
Contents
Preview
Metadata
Preview: Geography
Arc Feature Class selected

Usability problems

- Incompatibility of user's and system's conceptual models
- Poor error messages
- Extra functionality at the expense of usability
- Design not based on application-/user-specific requirements
- Usability is strongly correlated with users's productivity → **better design!**

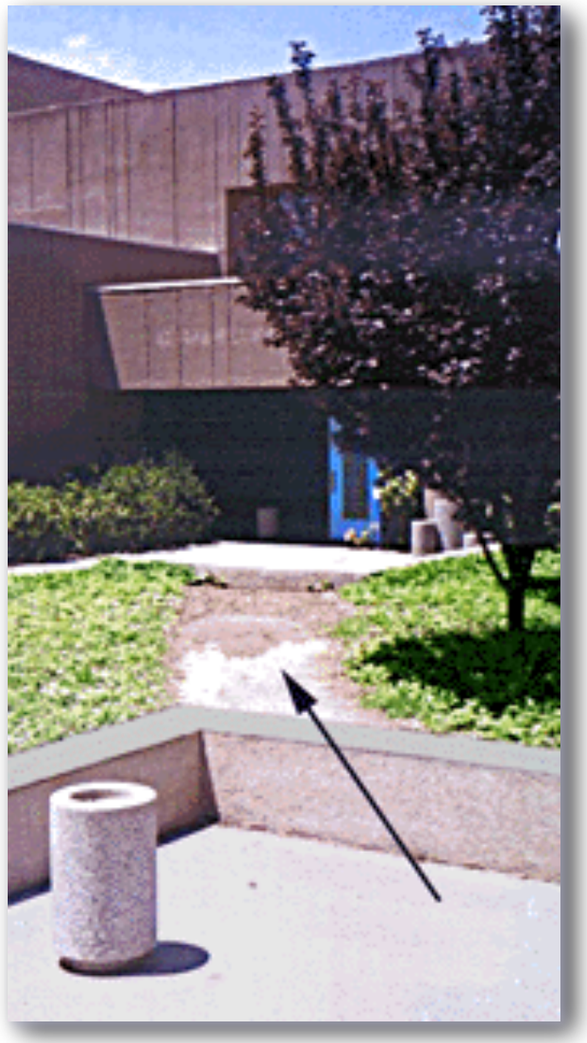
Why human-centred design?

- Minimize **hazards**
  - Accidents: transportation, industry, computing…
- Avoid and reduce **human error**
  - Avoid confusing or difficult to use designs
  - Avoid designs not matching our needs
- Need for **good designs** and **usability evaluation**
  - Design focusing on users, needs, tasks and goals
  - Usability evaluation based on empirical data with actual users



Examples of poor design…

http://seibt-bautzen.de/ews/bilder/herd.jpg



Examples of poor design…

http://upload.wikimedia.org/wikipedia/commons/thumb/5/58/Magic_Roundabout_Schild_db.jpg

http://www.roundaboutsofbritain.com/catalog/images/MR%20chopping%20board.jpg

Examples of poor design…

http://www.visitwiltshire.co.uk/xsdbimgs/X-20081018094755593.jpg

http://www.roundaboutsofbritain.com/catalog/images/mr%20t-shirt%20design%205.jpg

User-centred systems design

- Lessons to learn…
  - Most failures (errors) of human-tool/system use is due to poor design
  - Good design leverages human capabilities
- User-centred systems design (UCSD): focusing on **usability** during **development** process and system **life cycle**

UCSD: key principles

Vision
and plan

- ✓ initial concept
- ✓ business objectives and goals
- ✓ plan for UCSD

Analyze
requirements and user needs

- ✓ users, user context and scenarios
- ✓ user needs, usability requirements and design goals

Design for usability
by prototyping

- ✓ conceptual design
- ✓ interaction design
- ✓ detailed design

Evaluate
use in context

- ✓ evaluate early and continuously
- ✓ measure usability, business and effects

Construct
and deploy

- ✓ continuous focus on users and usability
- ✓ usability testing and monitoring

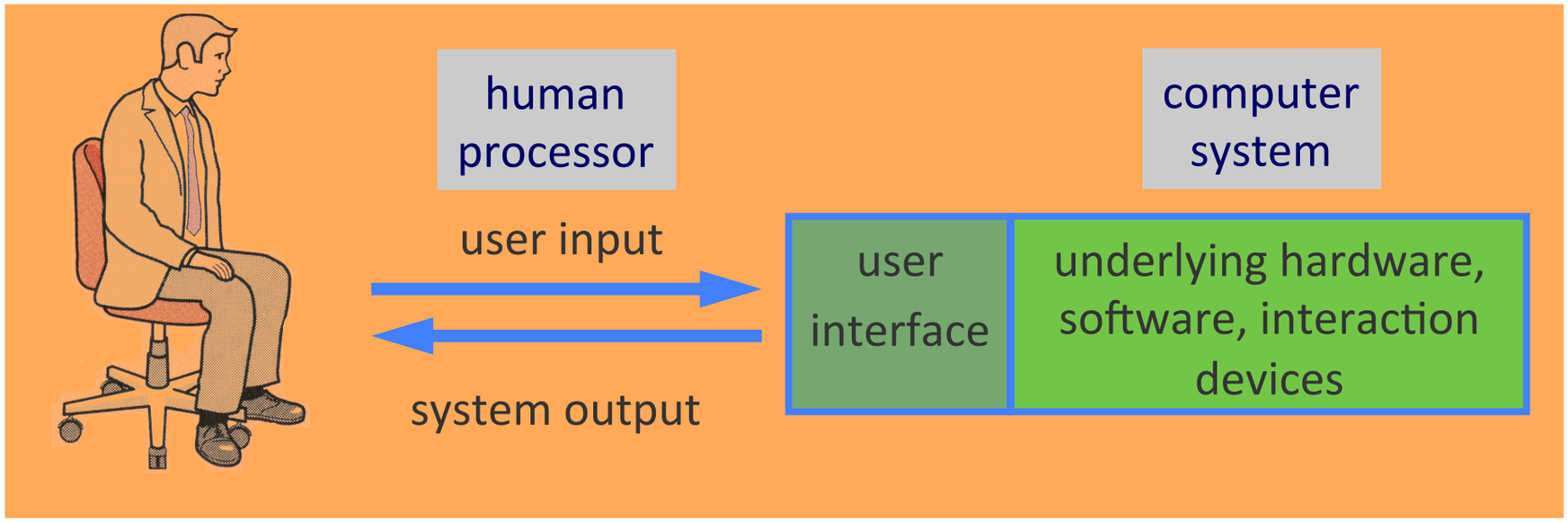
Feedback
plan the next iteration

- ✓ suggestion for changes
- ✓ project planning based on the outcome

http://www.it.uu.se/research/hci/acsd/KeyPrinciplesPoster-v.1.2en.pdf

Cognitive-oriented HCI

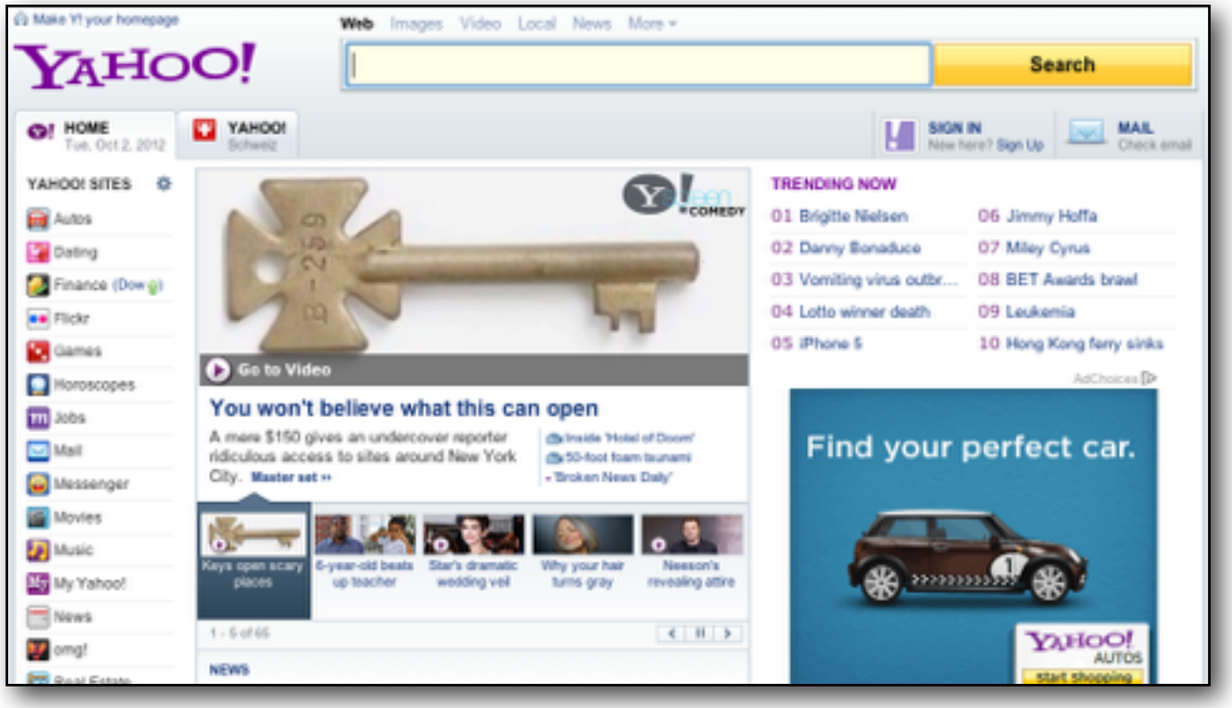
- Interactive system is a **distributed system** running on two information processors
- Based on cognitive theories: **perception**, **attention**, and **memory** to understand problem-solving and productivity in the sense of response times



After D. Stone, C. Jarrett, M. Woodroffe, S. Minocha. User interface design and evaluation. Elsevier. 2005

Design for **attention**

- Make information that needs attention **salient**
  - Colour, animation, ordering of items, etc.
  - Avoid cluttering the interface with too much information
  - Plain interfaces are easier to use



Design for **perception & memory**

- **Perception**
  - Icons should enable users to know easily (not guess) their meaning
  - Sound should be audible and distinguishable
  - Text should be legible: typeface, size, contrast, etc.
- **Memory**
  - Do **not overload** the user's memory with complicated procedures
  - Promote interfaces that support recognition

Design for perception: example

- Recent study from MIT's Age Lab shows:

  «The Font on Your Car's Dash Might Increase Your Risk of Crashing»

  http://agelab.mit.edu/files/AgeLab_typeface_white_paper_2012.pdf

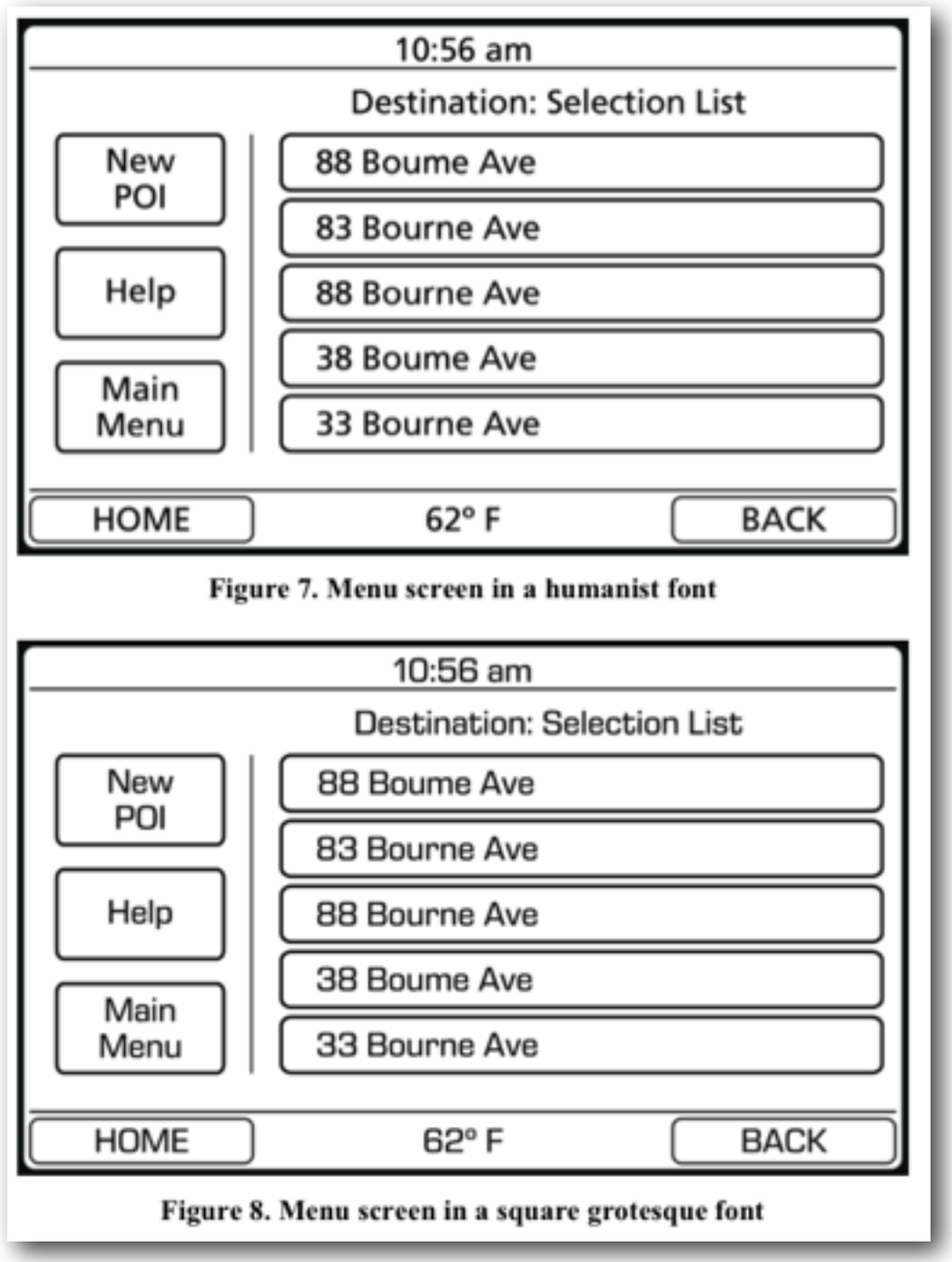


Figure 7. Menu screen in a humanist font

Figure 8. Menu screen in a square grotesque font

Design for learning & problem solving

- Learning
  - Encourage exploration (and allow come back easily)
  - Constrain and guide users to select appropriate actions
- Problem solving
  - Provide additional «hidden» information easy to access for advanced users, i.e. information how to accelerate and automate tasks (e.g. keyboard short cuts)

Some design guidelines…

- Make relevant parts visible (salient)
  - Systems are more usable when they clearly indicate their status, the possible actions that can be performed and the consequence of performed actions
- Give immediate feedback
  - Feedback is about sending back what action has been done and what has been accomplished allowing the user to continue with the activity
  - When anything changes it should be made visible (be salient!)
- Be consistent
  - Use of same interface elements everywhere
  - Use known elements if possible, and use metaphors
- Constrain the user
  - Restrict the possible interaction, but be consistent with the interface

8 golden rules for designing interfaces
(Ben Shneiderman)

1. Strive for consistency
   - identical terminology
   - similar menus, colour, layout, fonts, etc.
2. Enable frequent users to use shortcuts
3. Offer informative feedback
4. Design dialogs to yield closure
   - Group actions, make a sequence with clear progress and end
5. Offer simple error handling
   - No serious errors
   - Detect errors and offer simple handling
6. Permit easy reversal of actions
   - Undo, cancel action
7. Support internal locus of control
   - Users should be initiators of actions and not simple responders
8. Reduce short-term memory load
   - Keep displays simple
   - Don’t ask user to memorise some information

B. Shneiderman and C. Plaisant. Designing the user interface: strategies for effective human-computer interaction. Pearson/Addison Wesley, Boston. 2004.

Apple’s interface design guidelines

- Metaphors
  - familiarity for abstract domain
  - trash can => delete
  - no real world constraints
- Direct manipulation/user control
  - empowerment of user
  - see, point & click, drag & drop
- Feedback and communication
  - keep user informed
- Consistency
  - knowledge transfer
  - «Apple» & «File» always first two menu items
- What You See Is What You Get (WYSIWYG)
  - no hidden codes in file
  - bold is **bold**
- Forgiveness
  - UNDO, escape, cancel
- Perceived stability
  - clear set of options
  - grayed text, options
- Modelessness
  - «what & whenever they want it»
  - acceptable if familiar: drawing tool

HCI for Geovis?

- Cognitive principles are the same
  - Apply the same design principles!
- HCI principles also valid for cartography…!
  - Make relevant information visible!
  - Easy to recognise symbols on the map
  - Consistent symbols and colours
- … especially for interactive maps!
  - State of interactive map should always be clear
  - Forgiveness: go back to initial view
  - Offer informative feedback
  - Support exploring

HCI for Geovis: evaluate…

- Usability tests
  - Provide typical task to users
  - Measure efficiency (time)
  - Measure accuracy (correct answers)
  - Compare!
- Interface design is an iterative process

Users don’t read, they scan!

http://www.useit.com/alertbox/reading_pattern.html

Exercice

- Interactive mapping with Leaflet
- Applied HCI …
- Don’t forget these principles!

One more thing

- If one day you do a project:
  - in urban planning, development, tourism, etc.
  - Don’t forget the users
  - Don’t forget human-xyz-interface