Usability paradigms and principles

Readings:
Dix Chapter 4
Direct Manipulation vs Interface Agents (available on the course site)
Announcements

- Assignment 2 will be released on June 12 (instead of June 5).
- Due date: June 26.
Paradigms

Strategies for building interactive systems
Different interaction styles are based upon different paradigms
Paradigms for interaction

- Direct manipulation
  - the metaphor
  - information visualization
- Agent-based interfaces
- Pervasive (ubiquitous computing)
- Computer supported cooperative work
Direct manipulation

- term coined by Ben Shneiderman
- earlier ideas, e.g., Sketchpad, NLS, already allowed direct manipulation.
Direct manipulation

"Like it? We've replaced the steering wheel with all those buttons. Here's the user manual."
Direct manipulation

- preferred paradigm for graphical user interfaces GUI
- Also in other types of UI:
  - E.g. a spreadsheet allows for operating directly with numbers
- The user interacts with the visual representation of the data objects
- Advantages:
  - Exploits perceptual and motor skills
  - Depends less on linguistic skills than command-based and menu-form interfaces
Direct Manipulation usually

Involves using the tools provided in the UI
- e.g. drawing a rectangle in a drawing program

Some operations are done directly by manipulating the object itself
- e.g. changing the size of a rectangle by dragging from a drag handle

Activating the object changes its visual appearance; thus feedback is immediate
Direct manipulation: an example

I/O brush: The world as a palette © 2003-2005 MIT Media Lab

drawing tool to explore colors, textures, and movements found in everyday materials by "picking up" and drawing with them.

I/O Brush looks like a regular physical paintbrush but has a small video camera with lights and touch sensors embedded inside.

Behaviour:
- Outside of the drawing canvas, the brush can pick up color, texture, and movement of a brushed surface.
- On the canvas, artists can draw with the special "ink" they just picked up from their immediate environment.

http://web.media.mit.edu/~kimiko/iobrush/
The I/O brush
Direct manipulation (cont’d)

- Three basic principles (Schneiderman)
  - Continuous visual representation of the visible data objects
  - Physical actions or labeled button presses
  - Effects of actions are rapid, incremental, and reversible
Continuous visual representation

- Visual representation of the data objects can be verbal (words) or iconic (pictures)
- Opposite to visualization on command (ex: dir command in DOS)
- Examples
  - Icons representing files and folders on your desktop
  - Graphical objects in a drawing editor
  - Text in a WYSIWYG editor
  - Email messages in your inbox
Physical actions and labeled button presses

- Physical actions are the most direct kind of actions in direct manipulation:
  - In standard desktop interfaces:
    - clicking on an object to select it
    - dragging it to move it
    - dragging a selection handle to resize it

Direct manipulation based on physical actions is a central paradigm in virtual reality.
Direct manipulation in VR
Direct manipulation: button presses

- In standard desktop interfaces, not every interface function can be mapped to a physical action
  - e.g. converting text to boldface, text alignment etc.

Allow for “command” actions triggered by pressing a button
Is this direct manipulation?
Effects of actions in direct manipulation

- Rapid (within 100 ms); why?
- Incremental (continuous): dragging objects, or manipulating scroll-bar
- Reversible: DM should allow to undo the operation
  - obvious for physical actions (dragging, resizing etc.)
  - Buttons have to be actionable in two states (do/undo)
Direct manipulation cues

- Affordances
- Constraints
- Natural mapping
- Visibility
- Feedback
exercise

- What are the useful affordances of the folder icon in the Windows interface?
- What process helps the user understand how the folder widget works?
Metaphors
How to build good conceptual models for GUI design?

- Formulating an appropriate conceptual model and assuring that everything else is consistent with it.

  - Use metaphors (carefully)
    - Find an analogue object in the real world so that you can use the power of the positive transfer
    - Not all users may understand the point
    - They may take the metaphor too literally and try to do actions that were not intended
Metaphor: example #1

- The Macintosh trashcan: good metaphor for deleting files and documents
- Stretching the trashcan metaphor to include the function of ejecting diskettes:
  - Drag an image of the diskette to the trashcan to eject it from the computer
  - Good idea or bad?

"I don't want to delete the files on the diskette, I just want the computer to spit it out."
Metaphor: example #2

Source: Interface Hall of Shame
Metaphor: an example (cont’d)
Metaphor: an example (cont’d)

Source: Interface Hall of Shame

mouse over
Lessons learned

- interface metaphors can be misused;
- the presence of a metaphor does not at all guarantee an “intuitive”, or easy-to-learn, user interface.
- that beautiful graphic design doesn’t equal usability, and that graphic designers can be just as blind to usability problems as programmers can.
- metaphor is not the only way to build good conceptual models and achieve learnability.
Lessons learned (cont’d)

- Metaphor selection: one of the first and most important tasks in interface design.
- But...
- Good user interface design is not only based on metaphors
- Metaphors constrain the conceptual model of the interaction. ‘Magical’ extensions of metaphors (ex. Trashcan) are mostly perceived as counterintuitive.
- Metaphors don’t usually scale very well
Information visualization

- Also based on the direct manipulation paradigm, since the user can manipulate data objects
Information visualization

- Interactive information visualization = dynamic queries

Filmfinder
Agent-based interfaces

- Software agent = a paradigm for personalized HCI

- A software agent knows the individual user’s habits, preferences, and interests
  - Proactive (takes initiatives)
  - Adaptive (tracking users interests as they change over time)
  - Run autonomously (can act on your behalf while you are doing other things)
Why do we need agents?

- The current computer environment is getting more and more complex
- Number of tasks to take care of (# of issues to track) are continuously increasing
- Whenever workload gets to high, there is a point where a person has to delegate.
Agents learn about the user

- Example of an agent:
  Letizia – an agent that assists web browsing
  No visible body or character in the interface
  The user operates a conventional web browser
  The agent tracks user behaviour and attempts to anticipate items of interest

"Letizia Avarez de Toledo has observed that this vast library is useless: rigorously speaking, a single volume would be sufficient, a volume of ordinary format, printed in nine or ten point type, containing an infinite number of infinitely thin leaves."

- Jorge Luis Borges, *The Library of Babel*

Letizia video

What triggers the agent?

- Eager – an eager assistant; video
- Allan Cypher, Eager: Programming repetitive tasks by example. Proc CHI91
- Helps with repetitive tasks
- Problems:
  - What if the user makes an error
  - How can you make Eager stop
  - Users reluctant to let Eager finish the work

http://www.acypher.com/Publications/CHI91/EagerCHI.html
Interactive assistance in MS Office 97

The notion of interactive assistance is OK.

But...

The implementation in MS Office 97 as an animated paperclip is intrusive:

- Always on screen
- Incessant animation is distracting the user from the task; when it “speaks”, you cannot help but listen
- Closing it is only temporary; the paperclip reappears when the users accesses the Help menu
- Children love the paperclip

What went wrong in the design of the Office Assistant?
Agent-based interfaces versus direct manipulation

- The two paradigms do not exclude each other

“Opponents of agents typically argue that well-designed visualization interfaces are better. [...] you still need a well-designed interface when incorporating agents in an application. However, some tasks I may just not want to do myself even if the interface was perfect. If my car had a perfect interface for fixing the engine, I still wouldn’t fix it. I just don’t want to bother with fixing cars. I want someone else to do it.”

Pattie Maes
Human interaction with pervasive computers

- Major aspects of pervasive computing:
  - Computing is spread throughout the environment
  - Users are mobile
  - Information appliances are becoming increasingly available
  - Communication is made easier
    - Between individuals
    - Between individuals and things
    - Between things
A simple scenario for pervasive computing

- You walk up to an exercise machine and you wear some sort of identifier, biosensors etc.
- The exercise machine knows how to train and reward you
- “Hi Wendy, it’s been two days since you worked out. I think you’re ready to raise the weight by ten pounds. What do you say?”

User-machine dialogue
Continuous feedback as the user works out: the machine encourages the user to pull faster, longer, or slower to get a better workout.

After finishing the workout:
“ You’ve had a great workout. Go over to the vending machine and get yourself a reward”
- The exercise machine activates the vending machine.
Ongoing projects on pervasive computing: intelligent environments

- MIT House_n project:
  http://architecture.mit.edu/house_n/

- Georgia Tech: the Aware home:
  “Is it possible to create a home environment that is aware of its occupants whereabouts and activities? If we build such a home, how can it provide services to its residents that enhance their quality of life or help them to maintain independence as they age?”
  http://www.awarehome.gatech.edu/
Computer-supported cooperative work

- CSCW removes bias of single user / single computer system

- Electronic mail is most prominent success: facilitates collaborations between people in remote locations

- Other examples: tele-medicine, tele-presence etc.

- But also...

- Small group collaboration when people are in the same place: the Merl Diamond Touch table

- A display interface that allows users to maintain eye contact while interacting with the display simultaneously (i.e., without having to take turns).

The MERL Diamond touch table