Module II: Interfaces

Human-Computer Interaction
Graphical User Interfaces

Tools and Interfaces

- tools *augment* our power to think (and act)
  - example: tools for visualization of phylogenetic trees

- tools *constrain* our power to think (and act)
  - example: Roman number system:
    
    | I  | II | III | IV  | V  | VI | VII | VIII | IX | X  | XI | ... |
    |----|----|-----|-----|----|----|-----|-------|----|----|----|------|
    | 1  | 2  | 3   | 4   | 5  | 6  | 7   | 8     | 9  | 10 | 11 | ...  |

  - M: 1000   D: 500   C: 100   L: 50   X: 10   V: 5   I: 1
  - (try multiplying MMMCMDCXXIX * MMCMMLXXV)

Learning Goals [today]

*you should be able to*

- explain how tools augment and constrain our power to think and act, define the “myth of human error” and give examples that dispel this myth;

- explain strengths and weaknesses of human-computer interfaces, referring to concepts such as familiarity and consistency, mappings and metaphors, feedback, negative transfer, or additional concepts that you identify

The Myth of Human Error

- **myth**: faulty use of technology is usually the result of human error

- **fact**: many so-called human errors are actually errors in design
The Myth of Human Error

example: Harvard airplane user interface

when pilots practice stalling, they push the undercarriage horn cut-out button first, to turn off the alarm:
  - button, stall;
  - button, stall;
  - button, stall...
  ➔ conditioned response

The Myth of Human Error

problem: negative transfer

What happens now when a pilot stalls?

The Myth of Human Error

• deeper understanding of human physiology and psychology is changing the ways that digital interfaces are designed
• example: Ron Rensink's work on change blindness
  www.psych.ubc.ca/~rensink

Four Principles for Interface Design

1. aim for familiarity and consistency

  • on which side does the door open?

  • which status light indicates ok to proceed?
Four Principles for Interface Design

2. *use well chosen mappings and metaphors*
   - which burner does the blue knob control?

3. *provide useful feedback*
   - examples:
     - clicking sound of virtual keyboard
     - signal that battery is low

4. *manage complexity*
   - examples:
     - hierarchical organization of menus
     - ability to hide features that are not of interest
HCl research @ UBC

involves researchers from CS, Psychology, Commerce, Forest Resource Management and Engineering, with projects such as:
– D’Groove (digital haptic turntable):
  • www.cs.ubc.ca/labs/spin/projects/dgroove/
– the Aphasia Project:
  • www.cs.princeton.edu/aphasia/

Food for Thought

“… enjoy yourself. Walk around the world examining the details of design. Take pride in the little details that help…Give mental prizes to those who practice good design: send flowers. Boos to those who don’t: send weeds.”
– Donald A. Norman

Exercise:

• explain strengths and weaknesses of human-computer interfaces, referring to concepts such as familiarity and consistency, mappings and metaphors, feedback, negative transfer, or additional concepts that you identify
  • try this for Mapquest (www.mapquest.com) and CycleVancouver (www.cyclevancouver.ubc.ca)
• use whole sentences! provide concrete examples to illustrate your points

Module II:
Interfaces

Text-based Interfaces
Human Computer Interaction (HCI)

history in one slide

• The early days
• Punched cards
• Terminals and keyboards
  – text based interfaces
• The mouse
  – graphical user interfaces
  … but text is still important!

Human Computer Interaction (HCI)

history in one slide

• The early days
• Punched cards
• Terminals and keyboards
  – text based interfaces
• The mouse
  – graphical user interfaces
  … but text is still important!
Learning Goal

you should be able to

• use basic features of text-based interfaces such as Unix or search engines, with knowledge of the ways that special symbols are interpreted (or misinterpreted) by such interfaces.

Unix

• operating system deployed in the early 1970s to enable real-time sharing of computing resources among multiple users and tasks
• supports purely text based commands to act on data

Unix

• a user's files and directories are organized hierarchically under the home directory

Unix Commands

navigating directories

• when you log in or open a unix/linux window, your current directory is your home directory
• to do things with files in another directory, you need to change your current directory
Unix Commands

*navigating directories*

- `ls`: lists the contents of the current directory
- `cd` (change directory):
  - `cd dir` changes the current directory to the “child” directory `dir`
  - `cd ..` changes the current directory “up” to the parent directory

---

Exercise

- starting from the home directory, what sequence of commands could you use to learn the contents of each of the directories `dir2` and `dir3`?

---

Unix Commands

*navigating directories*

- special directory names:
  - `~`: home directory
  - `..`: parent directory
  - `.`: current directory

---

Unix Commands

*navigating directories*

- we can refer to a file or directory via its *path* from the current directory
- example: if the current directory is `dir2` then the path to `dir3` is `../dir3`
- exercise: if the current directory is `dir2`, what is the path to `dir4`?
More Unix Commands

- `more file1 ... filek`: displays file contents one screenful at a time
- `sort file1 ... filek`: sorts lines of all files together and lists the result
- `sort –u file1 ... filek`: remove duplicate lines from the sorted list

More Unix Commands

- `cp (copy) command`: copies files or directories
  - `cp file1 file2`
  - `cp file1 ... filek dir`

Exercise

- design experiments to answer the following:
  - `cp file1 file2`
    - does the command execute differently depending on whether `file2` already exists?
  - `cp file1 ... filek dir`
    - does the command execute differently depending on whether `dir` already exists?

Unix Commands

- expressiveness is provided through:
  - `variation of form`, e.g. `cp`
  - `options` specified after a `–` sign, e.g. `sort`
  - the `wildcard symbol` `*`
  - combining commands using the `unix pipe` `|`
  - `redirecting` output displayed on the screen into a file using `>`
Unix Commands

• examples:
  – **sort file**: sort all files starting with `file`
  – **ls | more**: list contents of current directory and display one screenful at a time
  – **ls > file**: store the list of contents of the current directory in `file`

Unix Command Summary

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ls</code></td>
<td>list contents of the current directory</td>
</tr>
<tr>
<td><code>cd dir</code></td>
<td>move to the “child” directory <code>dir</code></td>
</tr>
<tr>
<td><code>cd ..</code></td>
<td>move “up” to the parent directory</td>
</tr>
<tr>
<td><code>more file</code></td>
<td>display file contents</td>
</tr>
<tr>
<td><code>sort file</code></td>
<td>sort file lines and display</td>
</tr>
<tr>
<td><code>sort -u file</code></td>
<td>sort, suppressing duplicate lines</td>
</tr>
<tr>
<td><code>cp file file2</code></td>
<td>copy <code>file</code> to <code>file2</code></td>
</tr>
<tr>
<td><code>cp file ... file dir</code></td>
<td>copy files into directory <code>dir</code></td>
</tr>
</tbody>
</table>

Exercise

• what to you think would be the result of the following commands:
  - `cp file1 file2`
  - `cp file1 file2.*`
  - `more file.*`
  - `sort file.*`
  - `sort file.* > more`
Exercises

- write a sequence of commands that will merge the contents of the files `dir1/criteria1.txt`, `dir2/criteria2.txt` and store the result in a file called `all-criteria.txt` in directory `dir4`
- write a sequence of commands that will display on the screen all of the files in `dir1`, `dir2`, `dir3`, and `dir4`

Text vs Graphical File Navigation

*pros and cons*

Learning Goals [review]

*you should be able to*

- explain how tools augment and constrain our power to think and act, define the “myth of human error” and give examples that dispel this myth;

- explain strengths and weaknesses of human-computer interfaces, referring to concepts such as familiarity and consistency, mappings and metaphors, feedback, negative transfer, or additional concepts that you identify

Learning Goal [review]

*you should be able to*

- use basic features of text-based interfaces such as Unix or search engines, with knowledge of the ways that special symbols are interpreted (or misinterpreted) by such interfaces.