Hyper Text Design

James Baldo Jr.

SWE 432
Design and Implementation of Software for the Web
Thanks in part to: Jorn Barger (inventor of the weblog)
GUI Overview

• Graphical User Interfaces (GUI) are formed of
  – Windows
  – Icons
  – Menus
  – Pointers (WIMP)
  – Plus buttons, forms, commands, etc.

• We will study GUIs by starting with hyper text systems
Motivation – Overview

Hypertext provides **freedom from linear** representations of text.

It makes certain ways of accessing information more **convenient**.

We often **jump** around when we read:

- Technical papers
- Textbooks
- Newspapers
- Encyclopedia
- Tax forms

**Hypertext makes this more convenient**
The World Wide Web is a result of the *confluence* of three developments:

- Fast Computers
- Internet
- Hypertext Theory

World Wide Web
Hypertext

• ... is a non-linear way to display text
• ... is the basis for UIs on the web
• ... has been around in theory for 50 years
• ... provides information organization but not user interaction

We will discuss hypertext in general and then talk about GUIs and web interfaces
Five Categories of Problems in Hypertext Design

1. Page layout and navigation links
2. Making hypertext navigable
3. Connecting with the rest of the world
4. Platform issues: Speed, reliability, graphics, whitespace
5. Professionalism: Grammar, spelling
1) Layout and Text – Single Pages

• Reading from paper is easier than from screen
  – Don’t use as much text

• Organize text to let users get the summary first
  – Keep it short
  – Use the pyramid structure
    • First a summary paragraph
      – who, what, when, where, why, and how of the topic
    • Later paragraphs gives more detail

• People browse web pages, they don’t read
1) Layout and Text (2)

• In the **old days** … before the WWW
  – Small pages
  – Lots of clicking

• First realizations with the **WWW**
  – **Minor links** need to be in the same document
  – Sometimes we need to download entire long documents as a single PDF file, rather than lots of small HTML pages
1) Layout and Text – Images

- Images can be used for
  - Title of the page
  - Illustrations
  - Navigation
  - Buttons
- **Limit number** of images for users with low-bandwidth
- Always use **ALT** attribute to provide a text alternative
- Use **small size** images
1) Layout and Text – Animation

• **Animation** is usually annoying
  – It is as if the web page is shouting at you
• Things to animate: Shapes, logos, drawings, photographs
• Can be **effective** for
  – Attracting attention
  – Demonstrating transitions
  – Explaining complex systems
• Endlessly repeating animations will **annoy** the user
• Don’t use animation in **text**
1) Making Link Anchors

- **“Click here”** is intrusive – the control interferes with the content
- **Press the blue text is** even worse
- A link traversal should be **ancillary**
- **Good approaches:**
  1. Highlight **glossary terms**
     - Clicking on it will give a definition
     - On the Web, delays can be too great
     - Highlighted words in prose are distracting
  2. Embed the links in **footnotes**
  3. Allow the anchor to fit into the rest of the document **naturally**
1) Navigating Within Pages

• Try to avoid horizontal scrolling
• Navigation buttons should always be visible
• Action buttons should always be visible
• Put most important content on the top
• Logos and headers should not use too much space
• Use intra-page links to help users keep their hands on the mouse
  – (“home” button is on the keyboard)
1) HTML Design Hints

• Put search boxes on the landmark (speed)
• Do not use frames (SS, errors, speed)
  – Confuse navigation
  – Bookmarks do not work
  – Make printing hard
  – Browsers render them differently
  – Slower

Use `<TABLE>` to partition web pages

• On site maps, “low light” current page
1) Layout and Text – Writing

• Follow basic **principles of communication:**
  – Simple, direct style
  – Thorough proofreading for confusing grammar and misspellings
  – Use a spellchecker !
  – Avoid colloquialisms, slang, and culturalisms

• Text on a screen is **more difficult to read**
  – lower resolution

• Use **10 or 12 point** font (14 for older readers)
  – Stick to standard font types, don’t change too much

• **Remember:** If you make **mistakes:**
  – at best, people will think you are **careless**
  – at worst, they will think you are **ignorant**
2) Organization and Navigation

• Use a simple tree-structure as a basis
  – This is based on topology

• Every page needs a consistent menu with:
  • home, up, previous, next buttons
2) Organization and Navigation

- In hypertext, these three nodes will be documents
- Possibly a table of contents and two chapters

![Diagram of TOC and chapters]

TOC

CH1

CH2

prev

next

Up to TOC

ch1

ch2

Up to TOC
2) Organization and Navigation

• **Previous** button is the most problematic.
  – Imagine that you started at TOC, but chose to read Ch2 first
  – When you were done with Ch2, you might want to go back to TOC, which was the previous page you’d been reading ... but the Prev button will take you to Ch1 instead!

• Browsers have **back** buttons, but all buttons in HTML are **static**

• **Dynamic** buttons can be implemented with JavaScript
2) Organization and Navigation

• If there is **no next**:
  – omit the Next button, or
  – No-Next button, or
  – a grayed out Next button

• A **helpful technique**
  – Prev/Next/Up/etc. buttons should include some additional description of where they will go
  – HTML “alt” tag can be used
2) Organization and Navigation

• When a third level is added to this tree:
  – TOC
    • Ch1
      – Ch1p1
      – Ch1p2
    • Ch2
      – Ch2p1
      – Ch2p2
  – Next at Ch1p2 should go to Ch2p1
  • Stairmaster fallacy: Making the user go from Ch1p2 to Ch1 to TOC to Ch2 to Ch2p1 …
  • Example: http://www.gmu.edu/facstaff/handbook/
2) Organization and Navigation

From ch1p1:
• Offer an Up that leads to Ch1
• Offer a TOC (or Top) that leads to TOC. Top should not look like “Top of this file/doc”
2) Organization and Navigation

- Things get messy in hypertext when we go beyond a simple tree/hierarchy structure:
  - TOC1
    - Ch1
    - Ch2
  - TOC2
    - Ch1
    - Ch2 \(\text{– same as in TOC1, via a link}\)
- An **UP** link from **TOC2**, **CH2 should** go to **TOC2**, but it goes to **TOC1**!
- One solution: [Up to TOC1] and [Up to TOC2] – not general

We need **dynamic buttons**
2) Navigation Anchors

• Make clickable items **obvious**
  – Use standard color (blue) and underline

• Do not use blue and underline for **emphasis**
  – Users will get confused

• **Highlight** clickable items with mouseover

• Navigation should be on the **top or the left**

• Navigation buttons should not use much space
2) Case Study – 432 Website
2) Case Study – 432 Website

- Home
- Syllabus
- Schedule
- Resources
- Assigns
- Quizzes
- HTML files
- Lots
- PDF files

Questions:
- q1
- q2
- q3
- q4
- q5
- q6
- q7
- q8
- q9
- q10
- q11
- q12
- q13
3) Connecting With the Rest of the World

• Remember that users do not always come in through the “front door”
  – Bookmarks, URLs emailed from friends, search engines

• Each page should have:
  – Clear identifiers to indicate its context
  – Titles that are meaningful without the context
  – Navigation to other pages in the website

• Every page must have a meaningful <TITLE> tag
3) Connecting With the Rest of the World

- Do not break up information into chunks that are too small
- Do not use excessive graphics, applets, or other large files
- Clearly indicate non-HTML links
- We need a system to automatically update links when the target is relocated
- Add last modified notes – remember that creation is fun, but maintenance is hard
4) Browser Compatibility

• Each browser displays HTML differently
  – KISS – Keep It Simple Stupid

• There are many incompatibilities in the JavaScript DOM
  – Try not to get too fancy

• Browsers behave differently with broken HTML
  – IE tends to be more “forgiving” of mistakes
  – The Netscape / Firefox family tends to conform better to standards

Must test with multiple browsers !!
5) Professionalism: Grammar and Spelling

Follow basic principles of communication:

• Simple, direct style
• Thorough proofreading for confusing grammar and misspellings
• Avoid colloquialisms, slang, culturalisms

Remember: If you make mistakes, at best, people will think you are careless, at worst, they will think you are stupid
HTML Design Hints

• Put **searches** on the front page (speed)
• **Do not use frames** (SS, errors, speed)
  – Confuse navigation
  – Bookmarks do not work
  – Make printing hard
  – Browsers render them differently
  – Slower

Use `<TABLE>` to partition web pages

• On site maps, “**low light**” current page
• Remember the page may be **too big** for one screen
  – Put **navigation** buttons on top **and** bottom
  – Put **action** buttons on top **and** bottom
Seven Ways To Make Me Leave Your Homepage
Jeff Ruby, SWE 632, Spring 1997

1. Use **black** as a background color
   I have trouble reading the foreground text, regardless of its color, on almost every single page that has a black background. As soon as I see a black background, I pretty much ready to move onto something else.

2. Use a **busy background** image
   Busy backgrounds make the text difficult to read. When I see a busy background, that is my clue to get busy doing something else.

3. Use **frames**
   I don’t know why, but I just can't stand frames. They just annoy me. Maybe it is the whole scroll bar thing.
Seven Ways To Make Me Leave Your Homepage (2)

4. **Use large graphical images without including text links also**
   I use a 14.4 modem and big images often take a long time to load. I don't like to wait to see where I can go. It is also quicker to read text than to try to figure out how to get somewhere by looking at some fancy-dancy picture with 1/3 of all items in the world included in it.

5. **Use lots of spinning, whirling, twirling, twisting, rotating, running, scrolling or flipping images or Java applets**
   These busy items usually just distract and annoy me and make me want to go lie down and rest.

6. **Use blinking text**
   Blinking text is just the pits. My eyes always just go to it even when I'm reading something else.

7. **Use lots of large images**
   When I’m surfing around on the web, my second favorite thing to do is to sit and wait for images to load. Providing lots of them only doubles my pleasure. Coincidentally, my first favorite thing to do while surfing the web is to stare at blinking text.
Things That Make Me Like Your Homepage
Jeff Ruby, SWE 632, Spring 1997

• Include a quick index at the top of the page
  – It should concisely describe the contents of the page. This is usually in the form of a index bar, box or list which will link to spots farther down in the page or to other pages.
• Speed, speed, speed is the key
• Include a date or date last-updated at the top of the page

Make pages that are simple, readable and quick to use

1. Using **frames**
   - Navigation, bookmarking, printing, rendering, speed

2. Too much “**bleeding edge**” technology
   - Emphasize usability, not being cool

3. **Action**: scrolling, marquees, animations
   - We cannot look away from something moving

4. **Complex URLs**
   - Make them easy to type and remember

5. **Orphan pages**
   - Indicate which web site in which the page belongs
6. Long **scrolling** pages
   • Consider breaking it up into multiple pages
7. Lack of **navigation** support
   • Provide a site map
8. Non-standard **link colors**
   • How will users recognize the links?
9. **Outdated** information
   • Hire a “web gardener”
10. Overly **long download** times
    • Keep files and images small
10 Web-site Design Tips

1. Keep **graphics** and other bandwidth-intensive design elements to a minimum
2. Make **search** easy to find
3. Keep **content current**
   - Make archives of previously published content available
4. Don’t use **frames**
   - Navigation, bookmarking, printing …
5. Minimize the need to **scroll**
10 Web-site Design Tips

6. Structure content into **hierarchies**, but don’t make things too complex

7. Put your company’s **name and** logo on every page

8. **Key functions** such as shopping carts and help buttons should be easy to find

9. **Avoid pop-ups** that open new browser windows

10. **Links** to pages not yet seen should be **blue**; links to pages already viewed should be red or purple
A Wishlist for HTML

• There should be two ways to select a link, one that causes you to go there, and the other to “inspect” the nature of the link
  – where it leads, how big it is, what kind of document

• Navigation commands need to be standardized and built into HTML, so that we don’t have to put our own “up,” “next,” and “home” buttons on every page
  – This would require supplying a map of the topology, which the browser could translate into buttons
A Wishlist for HTML (2)

• A “virtual anchor” like “http://... bigtext.html#%75” that would allow you to link to a particular position (here 75% down from the top) in someone else’s document, without them having to insert a real anchor
  – http://ise.gmu.edu/~offutt/classes/432/sched.html#%50

• A “temporary bookmark” that expires at the end of a session. Used when you have to choose between following an interesting link and continuing to read an interesting page. It would make sure you get back to the branch-not-chosen, eventually ...

• Allow URL printing to be turned off

• Allow me to print one page of an HTML document
A Wishlist for HTML (3)

- HTML Variable tag:
  
  ```html
  <VARIABLE>
    <NAME>BaseUrl</NAME>
    <VALUE>http://www.ise.gmu.edu/~offutt/classes/432/</VALUE>
  </VARIABLE>
  
  To be used anywhere in the document:
  
  ```html
  <A Href=&BaseUrl;/sched.html>schedule</A>
  ```
• Improved radio tag syntax
  
  ```html
  <INPUT Type="radio" Name="major" Value="SWE">
  <INPUT Type="radio" Name="major" Value="CS">
  <INPUT Type="radio" Name="major" Value="INFS">
  ```

  This leads to errors with the Name attribute. We should use the “select” model:

  ```html
  <RADIO Name="major">
    <Option Value="SWE">
    <Option Value="CS">
    <Option Value="INFS">
  </RADIO>
  ```

  ASP has a specially designed control that looks similar to this