Web interface origins

• Origins of the web interfaces lie in hypermedia and hypertext

• Early beginnings of the excitement...
  - Vannevar Bush (Roosevelt science advisor, 1945)
    • memex tool: microfilm with encyclopedias of information and associative trails
    • just stare at short text and it would be “amplified”
  - Ted Nelson (1960s)
    • coined term “hypertext”
    • along with “docuverse” and “stretch text”
    • “computopian hopes” !!!

Web interface origins

• Development & implementation...
  - Douglas Englebart (1960s) - remember him?
    • Human Augmentation system: point-and-click, expanding outlines, etc.
  - Andries van Dam
    • earliest electronic books
    • exploited new technologies, especially graphics and animation (2d & 3d)

• By mid-1980s, hypertext was mainstream
  - primarily as a publication tool — presenting information with “convenient jumps”
  - Apple HyperCard (Bill Atkinson, 1987)

Hypertext

• Writing & reading hypertext is different than writing/reading normal text

• Three Golden Rules (Shneiderman):
  - 1. There is a large body of information organized into numerous fragments.
  - 2. The fragments relate to one another.
  - 3. The user needs only a small fraction of the fragments at any one time.

• What’s not (easily) amenable to hypertext? (according to Shneiderman...)
  - novels, poems
  - reference books?
  - news articles??
Hypertext

• How long should each “page” be?
  - experiment @ U Maryland:
    • many short articles vs. few long articles
    • subjects asked to answer questions about articles
  - result: short articles wins out
    • subjects w/ short articles answered more questions correctly and answered in less time!
  - of course, article length depends also on screen size, user experience, nature of task, etc.

Hypertext to web pages

• Hypertext was a necessary condition for web pages, but not a sufficient one
• What else do we need?
  - layout
  - images/icons
  - styles
  - GUI elements
  - animation?
  - scripted objs?

Categorizing web sites

• Categorizing by site goals
  - Sell products
    • e.g., book sellers, eBay
  - Advertise products or services
    • e.g., real estate agents, auto dealers
  - Inform and announce
    • e.g., universities, governments
  - Provide access
    • e.g., libraries, newspapers
  - Create discussions
    • e.g., bboards, chat rooms
  - Nurture communities
    • e.g., professional orgs, political orgs

Categorizing web sites

• Categorizing by size/genre
  - 1 - 10 pages
    • personal site, project summary
  - 5 - 50 pages
    • conference program, organization overview
  - 50 - 500 pages
    • city guide, product catalog
  - 500 - 5,000 pages
    • technical reports, film database
  - 5,000 - 50,000 pages
    • university guide, newspaper site
  - 50,000 - 500,000 pages
    • directories/indices, airline schedules
  - 500,000 - 5,000,000 pages
    • congressional digest
  - > 5,000,000 pages
    • Library of Congress, NASA archives
Is web design different from GUI design?

- Hmm. Experts don’t agree on this one.
- The “No” side
  - web site is clearly a user interface, and often one with GUI(-like) elements
  - usability ideas for GUIs transfer to the web
  - evaluation like user testing & heuristic evaluation remain largely the same (though might need new heuristics/ideas in this domain)

Is web design different from GUI design?

- The “Yes” side
  - you only design part of the interface; you don’t control one big part: the browser
  - user can manipulate many aspects of interaction
    - e.g., resizing windows, changing fonts, navigating back and forth, bookmarking, etc.
  - some things are out of both your & users’ control
    - e.g., download times, security
  - your pages are a tiny part of the web space
    - thus, seen by a small fraction of people, or (likely) a self-selected group of people
  - scale of particular sites
    - can large sites really be designed?

Web site design

- Site prototyping
  - storyboards are very useful (like always!)
  - flowcharts / hierarchies provide nice overviews of entire sites (or parts thereof)
  - e.g., site for “Aquatic Entomology” course

Web site design

- Navigation types
  - menu-tree navigation
    - menu hierarchy is visible at all times, user can go down then backtrack up

Next bunch of slides derived from information at http://www.edtech.vt.edu/edtech/id/interface/
Web site design

• Navigation types
  - tab-stop navigation
    • like menu-tree, but uses “tab” physical metaphor

Web site design

• Navigation types
  - index navigation
    • (almost) all information visible/accessible from top level

Web site design

• Navigation types
  - pull-down menu navigation
    • uses Javascript to select next page
    • display current selection??

Web site design

• Navigation types
  - iconic navigation
    • find information by picture
    • (perhaps) less ordered than textual menus (?)
Web site design

• Navigation types
  - page-turning navigation
    • natural for sequential information, not as useful for large and/or complex sites

Web site design

• Navigation types
  - navigation by site map or table of contents
    • site map has grouping+linking, table of contents doesn’t

Web site design

• Navigation types
  - navigation by search
    • essentially, builds an index on the fly based on given search terms

Web site design

• Navigation types
  - mixing navigation types...
Web site design

- Screen layout
  - balance is an essential component, as it is for any window

- Screen layout
  - focal point guides viewer’s eye to desired places

Web site design

- Screen layout
  - high-density layouts are difficult to navigate, especially without strong visual grouping

- Screen layout
  - consistency (as always) is key from page to page
Web site design

- Screen layout
  - metaphors can evoke mental models of well-known objects and improve recognition

Web GUIs

- Two ways to think about web GUIs...
  - (1) The web page itself is a GUI.
    - links = buttons
    - radio buttons & check boxes
    - input fields = text fields
    - pull-down menus
    - layout
    - keyboard shortcuts (browser)
  - (2) The web page can contain “sub-GUIs.”
    - i.e., applets!
    - Swing & Java give us the tools to make applets

Exercise #1

- First, let’s consider Web GUIs in their raw (non-applet) form as simple pages.
- Can we design iTunes for the web?

Exercise #2

- Can we design a Java IDE for the web?
  - provide editing, compiling, running, debugging
  - what would it look like?
  - what functions would you expect from it?
  - how does this compare to “on-your-machine”?
    - interaction feel?
    - other issues?

  - Note: Not too far-fetched! Could be good for a new programming language... REportal uses this, sort of...
Applets

• Ok, let’s return to Web GUIs as applets.

- Applet methods
  - init() : fast one-time initialization
    • should contain code normally in the constructor
  - start() : performs some work or starts threads
  - stop() : pauses work when applet not visible
  - destroy() : final cleanup

```java
public class Simple extends Applet {
    public void init() {
        // code
    }
    public void start() {
        // code
    }
    public void stop() {
        // code
    }
    public void destroy() {
        // code
    }
}
```

Applets

- Event handling, Part I
  - drawing by painting + handling events manually

```java
public class Simple extends Applet {
    public void paint(Graphics g) {
        // code
    }
}
```

- Event handling, Part II
  - treat applet like a normal window
    - JApplet is a top-level container, like JFrame & JDialog

```java
public class HelloSwingApplet extends JApplet {
    public void init() {
        JLabel label = new JLabel(
            "You are successfully running a Swing applet!");
        label.setHorizontalAlignment(JLabel.CENTER);
        label.setBorder(BorderFactory.createMatteBorder(1, 1, 2, 2, Color.black));
        getContentPane().add(label, BorderLayout.CENTER);
    }
}
```

Applets

- Example: HelloSwingApplet

```java
public class HelloSwingApplet extends JApplet {
    public helloSwingApplet() {
        // code
    }
    public void init() {
        JLabel label = new JLabel("You are successfully running a Swing applet!");
        label.setHorizontalAlignment(JLabel.CENTER);
        label.setBorder(BorderFactory.createMatteBorder(1, 1, 2, 2, Color.black));
        getContentPane().add(label, BorderLayout.CENTER);
    }
}
```

Applets

- Example: AppletDemo

```java
public class AppletDemo extends JApplet implements ActionListener {
    protected JButton b1, b2, b3;
    protected static final string DISABLE = "disable";
    protected static final string ENABLE = "enable";
    protected String leftButtonFilename = "images/left.gif";
    protected String middleButtonFilename = "images/middle.gif";
    protected String rightButtonFilename = "images/right.gif";
    private boolean isApplet = true;

    public AppletDemo() {
        // code
    }
}
```
Applets

Example: AppletDemo

```java
public AppletDemo(boolean isAnApplet) {
    ...}
public void init() {
    getContentPane(makeContentPane());
    public Container makeContentPane() {
        bi = new JButton("Enable Button", leftButtonIcon);
        bi.setHorizontalTextPosition(AbstractButton.CENTER);
        bi = new JButton("Disable middle button", leftButtonIcon);
        bi.addActionListener(DISABLE);
        JFrame pane = new JFrame();
        pane.add(b1);
        pane.setBorder(BorderFactory.createMatteBorder(1, 1, 2, 2, Color.black));
        return pane;
    }
    ...}
public void actionPerformed(ActionEvent e) {
    ...}
```

Example: AppletDemo

```java
public void actionPerformed(ActionEvent e) {
    ...}
    // Following code only for running code as an application!
    public static void main(String[] args) {
        JFrame frame = new JFrame("Application version: AppletDemo");
        frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        AppletDemo applet = new AppletDemo(false);
        frame.setContentPane(applet.makeContentPane());
        frame.pack();
        frame.setVisible(true);
    }
```

Applets

Including applets on a web page
- a simple way...

```html
<APPLET CODE="AppletSubclass.class" WIDTH=anInt HEIGHT=anInt>
</APPLET>
```

- with parameters and alternate text...

```html
<APPLET CODE="Animator.class" WIDTH=460 HEIGHT=160
ALT="If you could run this applet, you'd see some animation">
<PARAM NAME="imageSource" VALUE="images/Beans">
<PARAM NAME="backgroundColor" VALUE="0xc0c0c0">
<PARAM NAME="endImage" VALUE=10>
<PARAM NAME="soundSource" VALUE="audio">
<PARAM NAME="soundtrack" VALUE="spacemusic.au">
<PARAM NAME="pause" VALUE=200>
Your browser is completely ignoring the &lt;APPLET&gt; tag!
</APPLET>
```

Applets

Applet security
- What applets typically cannot do (subject to particular browser)
  - read or write files on the host that's executing it
  - make network connections except to the host that it came from
  - start any program on the host that's executing it
  - read certain system properties
  - control certain aspects of window appearance
### Applets

- **Applet security**
  - What applets typically *can do* (subject to particular browser)
    - make network connections to the host they came from.
    - cause HTML documents to be displayed.
    - invoke public methods of other applets on the same page
    - keep running after you leave their page (though this is not recommended/desired for most applets)
  - Note: Applets loaded from the local file system have fewer restrictions than network applets

### Web 2.0

- **What is “Web 2.0”?**

- **Who uses Web 2.0 sites?**
  - If you believe that bloggers are roughly like Web 2.0 users...

- **Surprisingly, users find advertising acceptable**
  - Web 2.0 could open up marketing options?
Web 2.0

- What changes might we expect from marketers?
  - More users are connecting to each other and content through networked, peer-driven activities & content
    - LinkedIn now has service referrals as part of their package
  - API’s and Content syndication will lead to more machine generated connections
    - “Non-compliant” content won’t fit into the flow as readily
  - Web 2.0 is truly two-way
    - Marketers need to be very willing to “listen” and receive more than broadcast
  - User-generated content may be more valuable to users than yours
  - Adoption will drive investments in online advertising
    - Investment in blog marketing will increase

Engineering Models of User Behavior

- We’ve talked in recent lectures about...
  - engineering models to predict user behavior in certain situations
  - user model frameworks (e.g., KLM-GOMS) that predict time-on-task
  - computational user models (e.g., production systems) as used for intelligent interfaces and model tracing
- The basic ideas also apply to the web domain... but we can do better by focusing on specifics of this domain
  - #1: SNIF-ACT model [Fu & Pirolli]
  - #2: Bloodhound system [Chi et al.]

SNIF-ACT

- Goal: Encapsulate web-browsing behavior in a computational model
  - ... to better understand behavior
  - ... to predict behavior (as we will see)
- Approach: Model based on...
  - Information Foraging
  - the ACT-R cognitive architecture -- a production system framework

SNIF-ACT

- Key component: Information Scent
  - distal info = desired info a few clicks away
  - proximal cues = info right now (e.g., link names)
SNIF-ACT

- Key component: Information Scent

Try finding information about majoring in Sociology at Drexel...
www.drexel.edu

SNIF-ACT

- Key component: Information Scent

\[ A_i = B_i + \sum_{j \in L} W_j S_j \]

Base-level activation

\[ A_i = B_i + \sum_{j \in L} W_j S_j \]

SNIF-ACT

- Key component: Information Scent

- for \( G \) = information goal (what user is seeking) and \( L \) = link to that information...

SNIF-ACT

- Key component: Information Scent

- Where do we get \( S_{ji} \)?

  - can construct activation networks from online text corpora and calculate \( S_{ji} \) for different words and information goals
  - and base-rate frequencies \( B_i \) of all words and pairwise co-occurrence frequencies of words can also be computed
SNIF-ACT

- **Declarative Memory**
  - perception puts info in memory: link names, etc

- **Procedural Memory**
  - knowledge of using the browser
  - stored as condition-action production rules
  - e.g., attend-to-link, click-link

SNIF-ACT

- **Web Behavior Graphs for two domains...**
  - (different colors = different web sites)

SNIF-ACT

- **Model predicts...**
  - (1) which links user will click on
  - (2) when people decide to leave a site

- **Testing**
  - these two actions extracted from log files
  - actions compared to model predictions

SNIF-ACT

- **Results**
  - histogram showing predicted model rank for clicked links...

(bias to the left better)

(Position model simply ranks by position, top & left better)
**SNIF-ACT**

- Results
  - scent values before and right when leaving site
    - values decrease below overall mean (dotted line)

**Bloodhound**

- So SNIF-ACT, Information Foraging, etc. provide reasonable predictions of behavior
- How can we instantiate this into a real tool?
- Bloodhound is a system that analyzes the information cues on a web site
  - strives for a simple application of theory: enter a web site + search words, get a usability report

**Bloodhound**

- Overview of algorithm (Web User Flow by Information Scent) and spreading activation

**Bloodhound**

- Overview of algorithm (Web User Flow by Information Scent) and spreading activation
Bloodhound

Overview of algorithm (Web User Flow by Information Scent) and spreading activation

Bloodhound

User testing
- 244 users, 1386 user sessions

Domains
- help.yahoo.com (Yahoo! help system section)
- www.rei.com (a camping/outdoor online store)
- hivin-site.ucsf.edu (AIDS and HIV medical site)
- parcweb.parc.com (company intranet)

Bloodhound

Tasks

Help.yahoo.com (7484 documents):
- You want Yahoo! to add your site to the Yahoo! Directory. Find some guidelines for writing a description of your site.
- When is the playing season for Fantasy Football?
- You want to get driving directions to the airport, but you don’t know the street address. How else can you get accurate directions there?

REI (36422 documents):
- You are planning a week-long hiking trip for this summer, and you’re on a budget. Find a single person tent for less than $120.
- Find the location of the REI store nearest you.
- Find yourself some warm, fairly heavy long underwear for the upcoming ski season.
- etc.

Bloodhound

Results

<table>
<thead>
<tr>
<th>Corr. Coef.</th>
<th>Yahoo</th>
<th>REI</th>
<th>Hivin-Site</th>
<th>Parc-web</th>
</tr>
</thead>
<tbody>
<tr>
<td>task 1a</td>
<td>0.7528</td>
<td>0.6048</td>
<td>0.6811</td>
<td>0.7394</td>
</tr>
<tr>
<td>task 1b</td>
<td>0.7218</td>
<td>0.6048</td>
<td>0.7885</td>
<td>0.8756</td>
</tr>
<tr>
<td>task 2a</td>
<td>0.7489</td>
<td>0.9892</td>
<td>0.6671</td>
<td>0.8210</td>
</tr>
<tr>
<td>task 2b</td>
<td>0.8540</td>
<td>0.7073</td>
<td>0.6880</td>
<td>0.8573</td>
</tr>
<tr>
<td>task 3a</td>
<td>0.7768</td>
<td>0.7321</td>
<td>0.8835</td>
<td>0.7197</td>
</tr>
<tr>
<td>task 3b</td>
<td>0.6973</td>
<td>0.6979</td>
<td>0.5600</td>
<td>0.7123</td>
</tr>
<tr>
<td>task 4a</td>
<td>0.9022</td>
<td>0.9415</td>
<td>0.8407</td>
<td>0.8340</td>
</tr>
<tr>
<td>task 4b</td>
<td>0.9052</td>
<td>0.7600</td>
<td>0.9631</td>
<td>0.9344</td>
</tr>
</tbody>
</table>

Table 2: correlation coefficients for frequency distribution comparisons between Bloodhound generated frequency vector versus user study data.
Summary

• Information Foraging + SNIF-ACT provides the theory.
• Bloodhound provides the usable system.
  - part theory, part database (for associations),
    part usable interface

• What about Web 2.0 applications?
  They’re working on it…