SIENAcollege
Computer Science

Computer Science 010
Introduction to Computer Applications
Siena College
Fall 2010

Topic Notes: World Wide Web

We next return to discuss more about the World Wide Web.

Web as a GUI

Think about some of today's popular *web-based services*:

- YouTube (video sharing)
- Flickr (image sharing)
- Twitter (short messaging)
- Facebook (social network)

None of these require you to install software on your computer beyond your web client/browser.

They use your browser as their GUI!

We are all familiar with this interface, but you may not have thought of it in those terms. What do we find in a browser when viewing a page?

- viewing area
- request bar
- pull-down menus at the top
- status bar at the bottom
- side bar for history

There are also some less visible parts of the browser user interface:

- *cookies* stored on your computer by web servers
- browser history
- request bar entry history
- cache of recently-visited pages

The standard browser interface is very limited in its capabilities, so a mechanism is provided for *plugins* and *addons* in modern browsers.

But it does have a significant advantage: once the browser is implemented for a particular platform, any of the applications that use the web for their GUI can work on that platform.

Plugins:

- Third-party programs to support a particular function such as playing audio or video.
- Generally work automatically once installed in the browser.
- If a plugin is not available for a particular type of content, the browser will help you find one.
- Work with the browser to process the specific content type.

Other ways to modify browsers are *addons*, *extensions*, *helpers*, *toolbars*.

- These change the browser interface itself and are not usually tied to a specific content type.
- They may change menu items, functionality of mouse buttons, provide new functions, or block content.

Cookies

Browser cookies are a mechanism to overcome a fundamental limitation in the client-server mechanism used to serve web pages: each web page request is a separate and self-contained event. We specify a URL to the server and it responds with content to be displayed by the browser.

However, it is often useful to connect a sequence of web page visits as a *session*.

Consider shopping at an e-commerce site. The site needs to know what items you have viewed and what items might be in your shopping cart, *etc.*

Cookies are small pieces of information stored on your computer by the web server that can help it keep track of who you are, where you've been, what's in your shopping cart..

Your browser will store this cookie on behalf of the web server and should only provide the contents of the cookie to that same web server in the future.

We will later discuss some of the problems and risks of the cookie mechanism.

Web Browser Scripts

Many web pages, even those that do not use any plugins like Flash, exhibit interactive behavior. This can be accomplished in a number of ways. We will consider briefly two common mechanisms.

First, HTML includes support for web *forms*. These are the text boxes, selectable buttons, drop-down menus, and pushable buttons that we see on so many pages.

The values we fill in on such pages are submitted to the web server when a "submit" or "OK" button is pressed on the page. Another page on the server, usually one containing a *script* is written to be able to interpret the values from the form elements and process them appropriately.

This is a very rudimentary mechanism but is enough to support many of the types of interactive web pages.

However, the web page is still static – it cannot react to any changes made to form values until someone submits the page.

We have all seen web sites that are much more interactive than that. A web page might be checking the validity of form data as it's being entered (for example, that a phone number has the correct number of digits) or some part of the form may be changed in response to a selection made in another (such as presenting only a list of U.S. states in a "state" menu when "U.S." is selected in the "country" menu).

The most common mechanism to support this kind of interaction is to embed a *program* within the web page that can execute and make changes to the web page itself in response to interactions with the viewer's browser. The most common language used is one supported by most modern browsers called *Javascript*.

By specifying some Javascript elements in a web page, we can control its behavior more precisely when values are entered into form elements or when buttons are clicked.

This capability provides a much richer foundation for web developers than straight HTML.

Safety of Scripted HTML

Can we trust these scripts? These are foreign prorams executing in your CPU moreover ones that you did not choose to install, you just happened to point your browser at a web page that included the scripts.

Fortunately, your browser supervises this execution, and doesn't (at least shouldn't) allow Web scripts to access, for example, your computer's hard disk, without permission. So why should we as web users worry?

- Your browser can store passwords and browsing history.
- Your browser can have more than 1 page open at once in tabs.
- Broswers have bugs!
  - Browsers are complex programs with lots of instructions and are bound to have programming mistakes.
  - There are many versions of browsers some are quite old and would contain well-known bugs.

  - Sometimes there are bugs that can be used as "exploits" to let script code get to your persistent storage, grab data your browser stores, or grab data you are entering on a different tab!

Some help: The NoScript addon for Firefox, which will block scripts from being executed unless you specifically permit them (which you would do only for sites you know and trust).

---

eCommerce

We next consider the use of the World Wide Web as a tool for commerce: *eCommerce*.

This can take many forms. One way to categorize:

- Business to Customer (B2C) Sales
- Customer to Customer (C2C) Sales
- Business to Business (B2B) Sales
- Providing on-line Services directly:
  - web hosting
  - electronic payment
  - storage space
  - word processing/spreadsheets/presentations

Web hosting is the provision of some or all of the following for a web site:

- a web server program
- storage space for the site
- domain name registration
- site email
- support services such as page design tools, a shopping cart mechanism, site advertising, connection to electronic payment systems, and database support

However, not all eCommerce sites charge their users money.

Consider the big example: Google. How do they make money? Selling advertising!

- AdWords – companies purchase the "sponsored links" on search results – they can bid for words that can be associated with their site

- AdSense – place ads on web pages deemed relevant to their site based on the page's content

Google also offers many other free services like Gmail and Google Docs. How do they make money from these?

- more advertising
- sell the services to large customers while giving them for free to individuals

Think about the many free social networking sites: Facebook, Twitter, Flickr. How can they make money?

Business Services on the Web

Companies sell business services to each other on the web. If one company has a resource or technology needed by another, they can sell or even rent it to the other company.

Many good examples can be found at Amazon services.

We think of them as a B2C operation, but they are also very much a B2B.

Businesses can also share information for mutual benefit such as managing a supply chain: when a retailer sells an item, the sale can immediately be reported to a supplier or even a manufacturer for tighter inventory controls.

Much of this communication is not practical to have humans involved – these are computer to computer communications.

We have considered HTML, which is a way for information to be specified with the intent of a visual representation (in a browser). For computer-to-computer transactions, the visual representation is not a factor, but a similar language is often used to represent the data: *XML (eXtensible Markup Language)*. HTML is in fact a subset of XML, but XML allows one computer to organize and represent data so that it is easy for another computer to interpret and understand it.

Web Search

You can find pretty much everything on the web. This includes

- Informative sites
  - vary in level of detail and level of authority
- Inaccurate or biased sites
- Misinformation
- Dangerous

- fraudulent

Search Engines

With so much out there, how do we find the accurate, reliable, trustworthy information about a topic of interest? We as individuals do not have the ability to scan through all sites looking for what we want.

Search engines have the ability to find web sites based on one or more *search terms* or *words*. Given the sheer size of the web, every aspect must be automated.

We will focus primarily on search engines, but there are also *subject directories* that organize links to web pages, often with the aid of human experts.

Let's consider how modern search engines work.

1. Web *crawlers* visit web pages to collect the words found on that page, then follow the links on the page (to find other pages).
2. This information is used to create and update a huge *index* of words and the web sites where those words are found.
3. The words specified for a search are located in the index, and a list of web pages that contain the words are gathered.
4. These pages are *ranked* to determine the order of their presentation in the search results.

This *ranking* of pages containing the search results is crucial. A search engine needs to present those pages that are most likely to contain what the searcher is seeking. A searcher needs to know that the pages presented first are the most relevant, authoritative, and accurate.

Early search engines had simple ranking schemes, such as ranking pages based on the number of times the search terms appear on the page.

Google developed a straightforward idea called *PageRank*: the more other sites that link to a given site, the more likely it is to be authoritative. So sites are ranked in the search results based (in large part) upon this measure.

Google also ranks sites higher that pay to be ranked higher.