University of Regina

CS 215
Web Oriented Programming

The Internet and the Web

Dr. Orland Hoeber
orland.hoeber@uregina.ca
http://www.cs.uregina.ca/~hoeber/cs215/

Readings

- Chapter 1 of the textbook

The Internet and the Web

- The World Wide Web, and the Internet upon which it is built, is a fundamental tool in the modern connected instant-access world we live in

- Designing and building software that operates within the framework of the Web requires knowledge of:
  - how the Internet and the Web work
  - mark-up and formatting languages
  - client-side and server-side programming languages
  - databases and data representation technologies

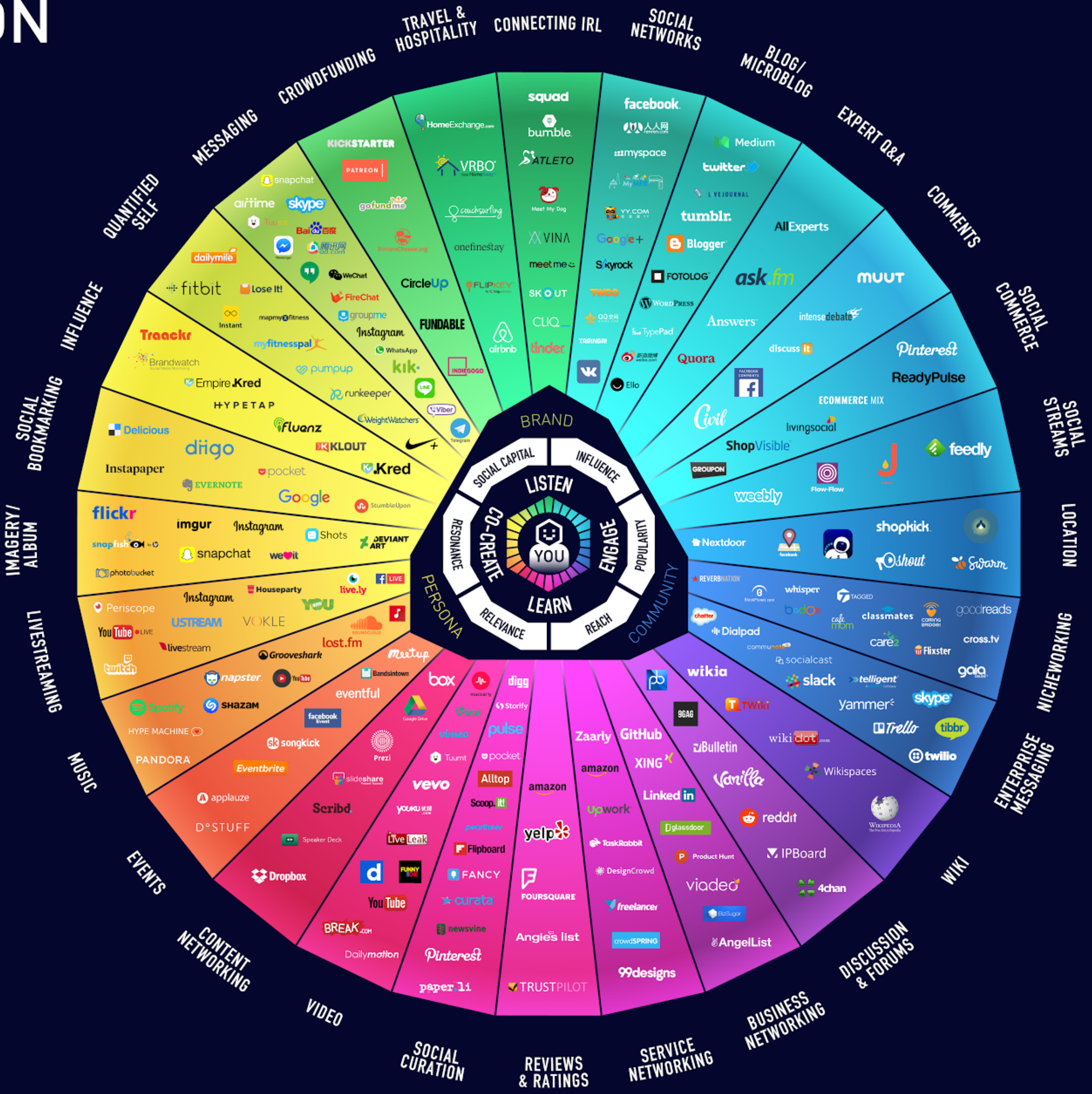
CONVERSATION PRISM 5.0

Brought to you by
Brian Solis & JESS3

Social Media Gave Everyone a Voice

The Conversation Prism debuted in 2008 as social media was exploding online. Social media would change everything about how we communicate, learn and share. It forever democratized information and reset the balance for influence.

The Conversation Prism was designed as a visual map of the conversational networks that continue to reshape everything. Its purpose is to help you understand and appreciate the statusphere so that you can play a productive and defining role in the conversations shaping our future.



For more information check out
conversationprism.com

http://www.theconversationprism.com

Internet Technologies

- The Internet is a worldwide network of computer networks
- There are three fundamental technologies that form the basis of the Internet
  - IP: Internet Protocol
    - addressing and routing of data
  - TCP: Transmission Control Protocol
    - reliable communication connection between sender and receiver over IP
    - send/acknowledge receipt
  - UDP: User Datagram Protocol
    - unreliable, but low latency, communication between sender and receiver over IP
    - just send

IP Addresses

- Every node on the Internet can be accessed via an unique numeric address
  - 32-bit number, normally represented as a group of four 8-bit numbers:
    - 142.3.150.53

- The pool of IP numbers is limited
  - IPv6 (128 bit IP numbers) was approved as a standard in 1998, but hasn't been widely deployed
  - NAT (network address translation) is a common solution to the problem of having limited IP addresses

Domain Names

- Since we are not good at remembering numerical IP addresses, textual domain names can be mapped to IP addresses
- Domain names use a nested naming structure
  - left-most domain: smallest
  - right-most domain: largest
  - example: www.cs.uregina.ca
- Decoded using a Domain Name System (DNS)
  - communicates over UDP to a DNS server
    - low latency
    - assumption that the DNS server is reliable
    - just re-send the request if you don't get a response in time

Applications on the Internet

- There are many different applications and associated communication protocols built upon the Internet
  - remote computer access: telnet
  - remote file systems and file transfers: FTP
  - electronic mail: mailto
  - hyptertext file transfers: HTTP

- Two common misunderstandings
  - the Web and the Internet are the same thing
  - email is sent over the Web

The World Wide Web (the Web)

- Origins:
  - specification of a new protocol for the exchange of hyperlinked documents in 1989: HTTP
  - the first graphical web browser appeared in 1993
- Key features:
  - client-server architecture
  - only client (web browser) can initiate requests
  - communication via HTTP
  - web browsers understand the format of web content, and know how to render/display/play these things

Web Browsers (continued)

- Since the first graphical Web browser, many more have been developed:
  - Netscape
  - Internet Explorer
  - Firefox
  - Safari
  - Opera
  - and many niche browsers
- Web browsers now run not only on desktop/laptop computers, but also mobile devices
- They are even embedded within other programs

Web Servers

- Web servers are programs that provide documents to requesting Web browsers
  - exist on well-known hosts on the Internet
  - tells the operating system that it can accept connections on a specific port (commonly 80 or 443)
  - runs as a background process, waiting for connections and requests via HTTP
  - acts only when requests are made to them by browsers
  - can send a specified file or run a specified program sending the results back to the browser

Web Servers (continued)

- Modern Web servers are complex and powerful
  - efficient at accessing files locally and sending these out over the Internet
  - can dynamically run software programs
    - PHP, ASP, JSP, Perl, Ruby, etc.
  - can access databases via these programs
    - MySQL, PostgreSQL, SQLServer, Oracle, etc.
  - can be configured to run may different websites simultaneously on the same machine
    - virtual hosts
  - can encrypt and decrypt data sent over the Internet
    - HTTPS

Client-Server Architecture

- The Web operates on a client-server architecture
  - Client
    - Web browser
    - initiates actions
  - Server
    - Web server, databases, etc.
    - responds to requests

- Web programming can be done at either or both ends
  - e.g., client-side programming in JavaScript; server-side programming in PHP
  - the choice of where the programming should occur should be driven by the strengths and resources of the client/server

Uniform Resource Locators (URLs)

- A critical aspect of the Web is the method for specifying the hypertext document to retrieve
- Rather than creating a Web-specific format, a more general structure is used: the URL
- General form:

  scheme:object-address

  - scheme
    - communications protocol such as http, telnet, or ftp
  - object-address (depends on the scheme)
    - for http, //fully-qualified-domain-name/document-path

URLs

- Example:
  - http://www.cs.uregina.ca/~hoeber/cs215/2017F/

- The object address when using http as the scheme includes a couple of important elements:
  - “//<host name>”
  - optional: “:<port number>”
  - optional: “/<path to file>”

- URLs cannot contain spaces or a set of reserved characters
  - special characters can be encoded using % and the hexadecimal ASCII code for the character
  - e.g., %20 = space

Hypertext Transfer Protocol

- The underlying protocol for performing all Web communications: HTTP
  - operates over TCP, which operates over IP
  - layered protocol structure of the Internet allows more complex features to be built upon more fundamental ones

- Client-server protocol that consists of
  - creating a connection between the Web browser and Web server
  - Web browser sending a request
  - Web server sending a response

HTTP Request

- Format

  <HTTP Method> <domain part of URL> <HTTP version>

  <Request Header Fields>

  <blank line>

  <Message Body>

- Example:

  GET /ClassesLabs/class.html HTTP/1.1

- The typical HTTP request methods are:
  - GET – fetch a document
  - POST – execute the document using the data in the body

HTTP Request

- The header fields are used to provide additional information as part of the request
  - the common use for these are to either indicate the host name of the server, the types of files that are acceptable, or to specify only to retrieve the file if it has changed since a certain date

    Host: www.cs.uregina.ca

    Accept: text/*

    If-Modified-Since: Sun 2 Sep 2012 16:05:00 GMT
  - If the request has a body (e.g., POST), then the Content-Length header field is mandatory

- The end of the request is signified in two ways:
  - no body (e.g., GET): the blank line
  - body (e.g., POST): the specified Content-Length is reached

HTTP without a Web Browser

- Since HTTP is a protocol that exists on the Internet, it is possible to communicate with this protocol using other applications:

```
eve:~ hoeber$ telnet www.cs.uregina.ca 80
Trying 142.3.150.53...
Connected to hermes.cs.uregina.ca.
Escape character is '^]'.
GET / HTTP/1.1
host: www.cs.uregina.ca

HTTP/1.1 200 OK
Date: Mon, 09 Sep 2013 22:03:37 GMT
Server: Apache
Accept-Ranges: bytes
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

…
```

HTTP Response

- Format

  <Status Line>
  <Response Header Fields>
  <blank line>
  <Response Body>

- Status Line format:

  <HTTP Version> <Status Code> <Explanation>

  HTTP/1.1 200 OK
  HTTP/1.1 400 Bad Request
  HTTP/1.1 404 Not Found

Response Headers

- The response headers are provided by the Web server to specify to the Web browser some information about the content
- The Content-Type header is mandatory
- Other useful ones are Server, Date, Last-Modified, Content-Length, etc.

- The response header is followed by a blank line, and then the content

HTTP/1.0 vs. HTTP/1.1

- The fundamental difference between HTTP version 1.0 and 1.1 deals with the length of time that the TCP connection to the server is kept open
  - HTTP/1.0
    - close the connection after receiving the response from the Web Server
  - HTTP/1.1
    - keep the connection open for a short period of time
    - speeds up subsequent requests for additional content
      - if a Web page is downloaded, it may include specifications for images that are to be embedded in the page
      - additional HTTP Requests must be made to retrieve these images

The Web Programmers Toolbox

- Although there are a range of tools to use in the development of a Web-based application, we will focus on just a few in this class
  - Client-side
    - HTML and CSS
    - JavaScript
  - Server-side
    - PHP and MySQL
  - Hybrid
    - AJAX

HTML and CSS

- HTML and CSS are markup languages, not programming languages
- Purpose:
  - HTML: to describe the general form and layout of documents
  - CSS: to describe the presentation rules for documents
- In an HTML document, tags are used to describe particular kinds of content
- In a CSS document, style specifications are provided to instruct the browser on how to render the content

JavaScript

- JavaScript is a client-side scripting language
  - interpreted language
  - Java-like syntax
  - dynamically typed
  - embedded within HTML documents, and executed on the Web browser

- JavaScript can be used to perform many useful functions on the client-side:
  - pre-process form data
  - manipulate the Web document itself
  - etc.

PHP and MySQL

- PHP is a server-side scripting language
  - interpreted, and dynamically typed
  - similar syntax to JavaScript
  - many built-in functions for array and text processing
  - PHP code is embedded in HTML documents
  - when the server processes the documents, the results of the PHP code are inserted into the HTML document in place of the PHP code, and then sent to the browser
  - easy to connect to and access databases such as MySQL for inserting and retrieving data

AJAX

- Asynchronous JavaScript and XML
- Traditional approach:
  - When something needs to be changed on a page, a new request is sent to the server, the new page is dynamically created, and the entire page is re-loaded with the change
- AJAX approach:
  - When something needs to be changed on a page, an AJAX request is sent, the new element is encoded in XML (or another structured format, such as JSON) and sent to the client, and JavaScript is used to change only what is needed in the page
  - During the process, the page remains loaded and can continue to be browsed and interacted with

Homework

- Next topic: Interface Design & Sketching

- First assignment is now posted to UR Courses