

Web Application Engineering

Jeff Offutt

<http://www.cs.gmu.edu/~offutt/>

SWE 642

Software Engineering for the World Wide Web

“Where is the wisdom we lost in knowledge?
Where is the knowledge we lost in information?”
– T.S. Eliot

Motivation – Overview

- Modern web applications are:
 - Distributed (world-wide)
 - Heterogeneous (hardware and software)
 - Highly user interactive
 - Built on new technologies
- The software is:
 - Very loosely coupled
 - Written in multiple languages
 - Often generated dynamically

Diverse: In terms of software, communication, and people

Motivation – Overview (2)

- Web application software has to be better than most shrink-wrap or contract software
- The combination of higher quality requirements and unique technologies make for a very *interesting* situation
(Academics think “interesting” means fun, managers think “interesting” is scary ...)

This talk discusses why and in what ways web software must be better

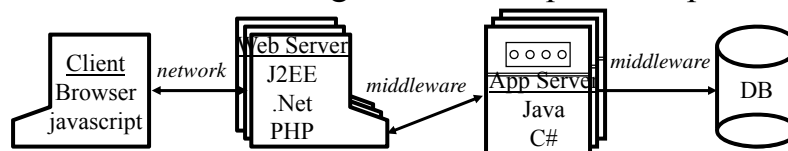
8/31/2010

© Offutt

3

General Web Terminology

- Web Page : Data that fits in one browser screen
 - Static : HTML exists as a file on a computer
 - Dynamic : Created as needed
- Web Site : A collection of connected web pages
- Web Application : A program that is deployed on the web
 - UI is in HTML
 - User interacts through HTTP’s request / response cycle



TAROT, June 2010

© Jeff Offutt

4

Software Deployment Methods

- Bundled : On your computer when you buy it
- Shrink-wrapped : Bought at a store on a CD
 - Downloaded from company's website or OSS site
- Contract : Single customer
- Embedded : Installed on an electronic device
- Web application : On the web through a URL
 - Component-based
 - Concurrent / distributed
 - One copy on the server
 - Can be updated at any time (fast update cycle)
 - User interactive

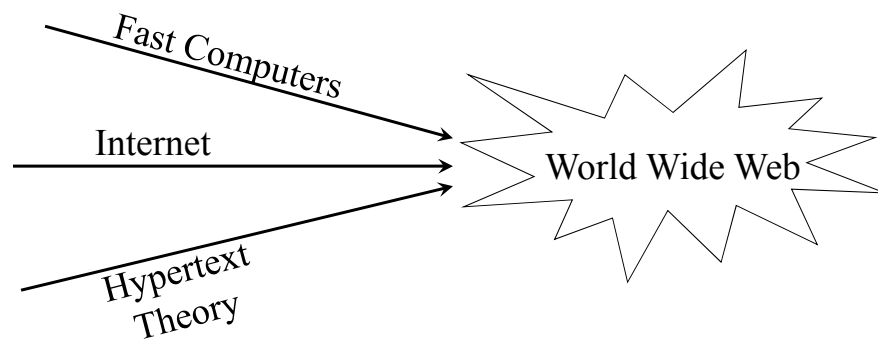
8/31/2010

© Offutt

5

Formation of the WWW

The World Wide Web is a result of the *confluence* of three developments



8/31/2010

© Offutt

6

Historical Highlights

Hypertext, Internet, and the Web

- 1945: Vannevar Bush proposes hypertext
- 1965: Ted Nelson coins the term “Hypertext”
 - Douglas Engelbart invents mouse
 - (The ACM Douglas Engelbart award is given yearly at the HyperText conference)
- 1968: Engelbart’s hypertext system “Augment/NLS”
 - Brown university’s HES
- 1969: Arpanet
- 1970: Unix
- 1972: Tomlinson invents email
- 1977: UUCP
- 1978: First bulletin board system – Christensen and Sues

8/31/2010

© Offutt

7

Historical Highlights (cont.)

- 1979: Truscott and Bellovin’s usenet news
- 1982: < 100 sites on “internet” (netnews)
- 1983: TCP / IP defines the internet
- 1986: NSFNET
 - SGML
- 1987: HyperCard (Apple)
 - First hypertext workshop
- 1989: WWW proposed by Berners-Lee
- 1990: HTML defined
- 1992: CERN (Switzerland) releases WWW
- 1993: NCSA Mosaic

8/31/2010

© Offutt

8

Historical Highlights (cont.)

1994: WWW surpasses gopher

1994: Netscape

1995: Lycos search engine

Java

DEC's Alta Vista search engine

15 million pages

1996: US Telecommunications Bill

1997: 31,000,000 pages, 476,000 servers, 14,000 news groups

2000: More than 100,000,000 hosts

Amount of back-end programming exceeds the front-end
hypertext

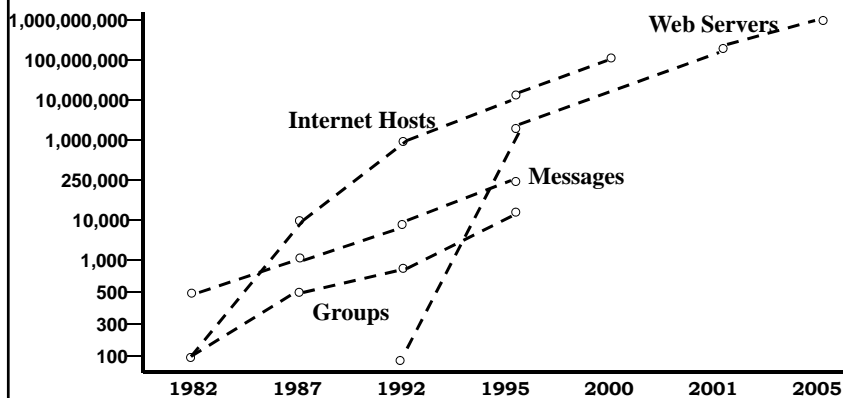
2004: 3,307,998,701 pages (google), 500,000,000 hosts,
1,200,000,000 users

8/31/2010

© Offutt

9

Internet Size



2000 : Over 2.1 Billion web pages

8/31/2010

© Offutt

10

The World Wide Web

The world wide web was invented by Tim Berners-Lee, an Englishman at CERN, the physics research lab in Switzerland

Tim Berners-Lee's vision of the world wide web was to make all information available to all people at all times

General Hypertext Terminology

- Node : Atomic amount of text – smallest unit of information known to system (pages)
- Link : Connects one node to another. Depicted by buttons or highlighted region.
 - Link Region (anchor) : Location in a node where link starts.
- Traverse (visit or view) : Moving from one node to another
 - Backtrack : Following links in reverse
- Landmark : Semantically important nodes that are linked to many nodes
- Hypermedia : Incorporates sound, video, pictures, or other non-textual information

General Web Terminology

- Web Page : Data that fits in one browser screen
 - Static Web Page : A web page whose HTML exists as a file on a computer
 - Dynamic Web Page : A web page that is created as needed
- Web Site : A number of web pages that are *semantically* related and *physically* linked
- Web Application Software : Software that makes web sites dynamic
 1. Dynamic Web Pages
 2. User Interaction

Web Software Terms

- HTTP, HyperText Transfer Protocol : The network protocol used to connect to web applications
 - It is fast, lightweight, and connectionless
- CGI, Common Gateway Interface : The protocol used to transfer data from web browsers (client-side) to executable programs (server-side)
- RDBMS, Relational Database Management System : A type of database that uses tables of rows and columns based on SQL
- DBA, Data Base Administrator ; Someone who manages the data, schema, and connections to a DB
- SQL, Structured Query Language : The primary language used to communicate with a databases

Modern Applications

Web applications have continuously evolved in the last decade

We have moved from:

- ftp/email ...
- to gopher ...
- to simple html pages ...
- to web sites ...
- to dynamic html ...
- to web commerce ...

with amazing speed!

Modern Web Applications

- Web applications are too complicated for individuals to manage
- They need to be engineered by teams of people with diverse talents:
 - Programming skills
 - Graphics design
 - Usability
 - Information layout and engineering
 - Data communications
 - Data base

We need web application engineering

Important Quality Attributes for Traditional Software

Traditional

1. Efficiency of process (time-to-market)
2. Efficiency of execution (performance)

○
○
○

50. Reliability
51. Safety
52. Maintainability
53. Security

○
○
○

Important Quality Attributes for Web Software

1. Reliability
2. Usability
3. Security

Customers have little “site loyalty” and will switch quickly, thus time to market is much less important than in other application areas.

-
4. Availability
 5. Scalability
 6. Maintainability
 7. Performance & Time to Market

(but still important!)

Based on an informal survey of around a dozen software development managers, 2000

First Generation Web Sites (Mosaic, '93)

- Text with few images
- Weak layout facilities
- Very limited CGI programming – forms
- No background color

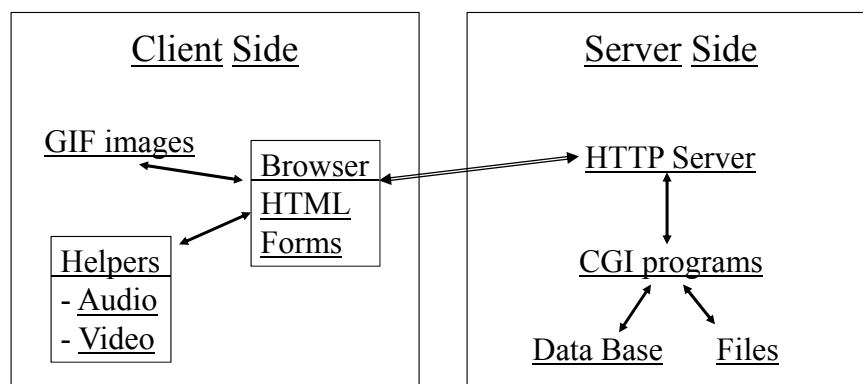
One “webmaster” could do it all!

8/31/2010

© Offutt

19

First Generation Web Sites



Web sites were usually created by one “early adopter”
Most were poorly designed and incomplete

8/31/2010

© Offutt

20

Second Generation Web Sites (Netscape, '94)

- Centering and other layout options
- Background colors
- Tables

- Graphics designers started to develop web sites
- Technical & creative jobs diverged
- A lot of emphasis on appearance (over-designed brochures)

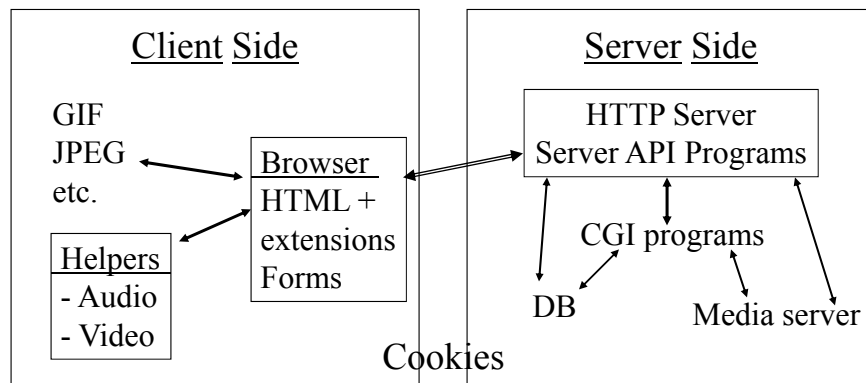
- Designers realized: People did not read websites, they visited them for a purpose

8/31/2010

© Offutt

21

Second Generation Web Sites



More programming, but mostly print-oriented
Cookies allowed state information to be saved

8/31/2010

© Offutt

22

Third Generation Web Applications (Netscape, IE, 96—97)

- People expected interaction
- Bandwidth became a serious problem (still is!)
- More layout, proprietary tags, frames, stylesheets, ...

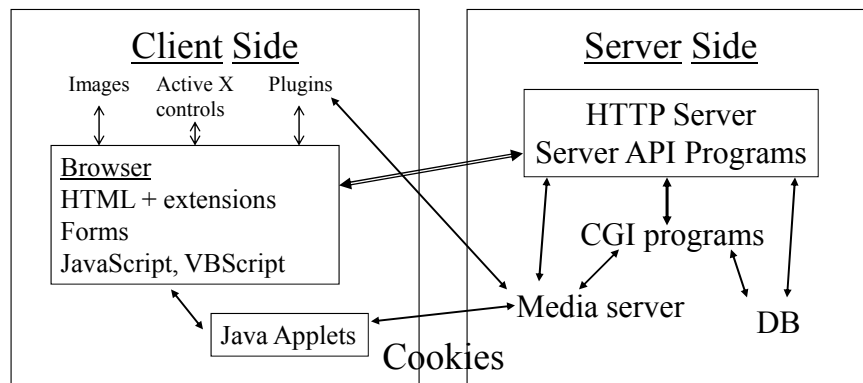
- Java
- JavaScript
- ActiveX
- Applets
- ...

8/31/2010

© Offutt

23

Third Generation Web Applications



Full interactivity and rich software abilities

8/31/2010

© Offutt

24

Third Generation Web Applications

New interactive applications

- FedEx document tracking
- Airline flight services
- Ticket agencies
- Maps

The birth of E-commerce

Impossible for one person to do everything.

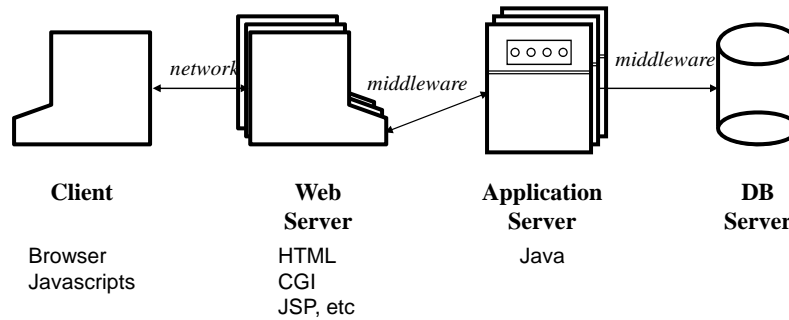
A Web Manager must manage programmers, graphics designers, DB administrators, security experts, usability engineers, business experts,

Third Generation Web Applications

Uses of web applications:

- Marketing
- Promotion
- Commerce
- Entertainment
- Technical support
- Investor relations
- Employee recruitment
- Personal satisfaction and expression
- Distributing research results
- ...

Fourth Generation Web Applications (software-centered)



Client-server ... 3-tier ... N-tier ...

8/31/2010

© Offutt

27

Problems Can Occur Anywhere

- 1995 : Web sites were 100% interface
- 1998 : Web sites were about 90% interface
- 2001 : Web sites are less than 50% interface
- 2005 : Web applications about 25% interface
- 2009 : Web application development dominates the software industry

There is still a shortage of knowledgeable, skilled web programmers and software engineers

8/31/2010

© Offutt

28

A Time of Transition

- The “.com-.gone” crash of the early 2000s was terrible for many of my friends and students who lost jobs and took pay cuts
- But I see a positive sign – these claims are validated:
The companies that “figured it out” are still alive, doing well, growing, making money, and hiring
- Henry Ford said “everybody is selling something”
- I’m not selling tools, languages, technologies or even research – I’m selling engineering

8/31/2010

© Offutt

29

Current Issues in Building Web Apps

- Software development process for the web?
- Design modeling for web applications?
- Testing web applications?
- How to maintain and evolve web applications?
- Which design patterns and frameworks work, and when?
- Programming technologies:
 - 1998 : HTML 3.2, Java 1.1, CGI, Perl, JavaScript, AWT, Active Server Pages, ...
 - 2001 : HTML 4.0, Java 1.2, Java Servlets, EJB, XML, JSP, ...
 - ☺ More centered (around Java)
 - 2002 : .NET
 - 2004-2010 : Lots of frameworks and APIs

8/31/2010

© Offutt

30

Summary Changes in Web Technologies

<u>1997</u>	<u>1998-1999</u>	<u>2000-2004</u>	<u>2004-2010</u>
<ul style="list-style-type: none"> ■ Static web pages ■ “Soft brochures” ■ Webmasters ■ HTML, CGI, JavaScript 	<ul style="list-style-type: none"> ■ Dynamic HTML ■ Programs (poorly written) ■ Confused webmasters ■ ASP, CSS, ... 	<ul style="list-style-type: none"> ■ Web applications ■ ECommerce + ■ Web services ■ WebManager + programmers, DB, network, UI, graphics designers, ... ■ Java (JSP, Servlets, beans), {HT,U,X}ML, Component-based ■ .NET 	<ul style="list-style-type: none"> ■ Web application frameworks ■ Web 2.0 ■ AJAX ■ Widespread enterprise applications

8/31/2010

© Offutt

31

Summary Concerns of Software

<u>Traditional</u>	<u>Web Software</u>
<ol style="list-style-type: none"> 1. Efficiency of process (time to market) 2. Efficiency of execution <p style="text-align: center;">○ ○</p> <ol style="list-style-type: none"> 50. Reliability 51. Safety 52. Maintainability 53. Security <p style="text-align: center;">○ ○ ○</p>	<ol style="list-style-type: none"> 1. Reliability 2. Usability 3. Scalability 4. Security 5. Availability 6. Maintainability 7. Performance & Time to Market <p style="text-align: center;">○ ○ ○</p>

8/31/2010

© Offutt

32

Technology Predictions for 2020

- 1990 : Command-based email, FTP
- 2000 : Widespread use of web applications
- 2010 : A web-based world



I am not foolish enough to make 10 year predictions about the web!

8/31/2010

© Offutt

33

Summary

The concept of
Web Application Engineering
is just beginning ...

8/31/2010

© Offutt

34