Web Software Model

CS 4640
Programming Languages for Web Applications

[Robert W. Sebesta, “Programming the World Wide Web”
Upsorn Praphamontripong, “Web Mutation Testing”]
Web Applications

• User interactive software programs, **deployed on a web server, accessed via a web browser**

• Use enabling technologies to
  • Make web site contents **dynamic**
  • Allow users of the system to implement business logic on the server

• Let users **affect state** on the server

• Constructed from diverse, distributed, and dynamically generated **web components**
  • Web components are software modules that implement different parts of the application’s functionality

**An enabling technology makes web pages interactive and responsive to user input**
How do Web Apps fit in with the World Around Them?

Object type: Hotel

Object type: Car

Object type: Car

Object type: Car
Objects and Properties

Object type: Hotel
Properties
Name: Awesome
Rating: 5
Rooms: 70
Bookings: 56
Pool: true
Gym: true

Object type: Car
Properties
Make: UVA1
currentSpeed: 30mph
Color: yellow
Fuel: gasoline

Object type: Car
Properties
Make: UVA2
currentSpeed: 20mph
Color: red
Fuel: gasoline

Object type: Car
Properties
Make: UVA2
currentSpeed: 35mph
Color: blue
Fuel: gasoline
Objects and Events

Object type: **Hotel**

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserve</td>
<td>reservation is made</td>
</tr>
<tr>
<td>Cancel</td>
<td>reservation is cancelled</td>
</tr>
</tbody>
</table>

Object type: **Car**

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break</td>
<td>driver slows down</td>
</tr>
<tr>
<td>Accelerate</td>
<td>driver speeds up</td>
</tr>
</tbody>
</table>

Object type: **Car**

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break</td>
<td>driver slows down</td>
</tr>
<tr>
<td>Accelerate</td>
<td>driver speeds up</td>
</tr>
</tbody>
</table>

Object type: **Car**

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break</td>
<td>driver slows down</td>
</tr>
<tr>
<td>Accelerate</td>
<td>driver speeds up</td>
</tr>
</tbody>
</table>
Objects and Methods

Object type: Hotel

<table>
<thead>
<tr>
<th>Method</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td>makeReservation()</td>
<td>increases value of bookings property</td>
</tr>
<tr>
<td>cancelReservation()</td>
<td>decreases value of bookings property</td>
</tr>
<tr>
<td>checkAvailability()</td>
<td>subtracts value of bookings property from value of rooms property and returns number of rooms available</td>
</tr>
</tbody>
</table>

Object type: Car

<table>
<thead>
<tr>
<th>Method</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td>changeSpeed()</td>
<td>increases or decreases value of currentSpeed property</td>
</tr>
</tbody>
</table>

Object type: Car

<table>
<thead>
<tr>
<th>Method</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td>changeSpeed()</td>
<td>increases or decreases value of currentSpeed property</td>
</tr>
</tbody>
</table>
## All Together

### Object type: Hotel

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
<th>Method called</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserve</td>
<td>reservation is made</td>
<td>makeReservation()</td>
<td>Name:</td>
</tr>
<tr>
<td>Cancel</td>
<td>reservation is cancelled</td>
<td>cancelReservation()</td>
<td>Awesome</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rating:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rooms:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bookings:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>56</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pool:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Gym:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>true</td>
</tr>
</tbody>
</table>

**Method**
- **makeReservation()**
- **cancelReservation()**
- **checkAvailability**

**What it does**
- increases value of *bookings* property
- decreases value of *bookings* property
- subtracts value of *bookings* property from value of *rooms* property and returns number of rooms available

### Object type: Car

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
<th>Method called</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break</td>
<td>driver slows down</td>
<td>changeSpeed()</td>
<td>Make:</td>
</tr>
<tr>
<td>Accelerate</td>
<td>driver speeds up</td>
<td>changeSpeed()</td>
<td>UVA1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>currentSpeed:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Color:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>yellow</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fuel:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>gasoline</td>
</tr>
</tbody>
</table>

**Method**
- **changeSpeed()**

**What it does**
- increases or decreases value of *currentSpeed* property
Web Browsers and Objects

Object type: **Window**

**Properties**
- **Location**: http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html

Object type: **Document**

**Properties**
- **URL**: http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html
- **lastModified**: 01/14/2019 10:19:23
- **Title**: CS4640: WebPL

---

Spring 2019 — Syllabus

**Contact information**

**Instructor**: Upsorn Praphamontripong
**Office**: Rice Hall 206
**Office hours**: TBD (until they are settled, whenever my door is open)

**TAs**: TBD

**Class hours**

Tuesday, Thursday 2:00PM - 3:15PM, Clark Hall

**Overview**

The way web software is built has been rapidly changing. There are now companies gathering to direct customer sales (e-commerce) and running software on a large scale (cloud computing). Many new technologies and frameworks have emerged over the past few years. The question is, how do we determine what programming languages should one learn and what programming styles should you use? What programming languages are in great demand? How will you keep up with Web 2.0? These are just some of the questions that you will be considered by companies in the future. Web developers who know more than one language generally have a better chance of getting a job.

This course will help you see how fundamental concepts of web development can be applied to develop reliable and usable web software regardless of the technologies or languages or frameworks. Although we put our emphasis on the concepts, you will develop dynamic web software with several commonly used programming languages and technologies.

You will work on user interface design, the front end development, back end development, and web-based information retrieval and processing. The class will be blended with practical experiences. You will implement a web-based application that involves team work, interpersonal skills and conflict management.
BOM: Browser Object Model

BOM – collection of objects that the browser makes available to us for use with JavaScript

- window Object
- location Object
- history Object
- document Object
- navigator Object
- screen Object
- form Object
- link Object
- image Object
DOM: Document Object Model

**Object type:** Document

**Properties**
- URL: [http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html](http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html)
- lastModified: 01/14/2019 10:19:23
- Title: CS4640: WebPL

**Event**
- **Load**
  - Page and content have finished loading
- **Click**
  - User clicks the mouse over the page
- **Keypress**
  - User presses down on a key

**Method**
- **write()**
  - Adds content to the document
- **getElementById()**
  - Accesses an element of a given id attribute

---

**Contact information**
- **Instructor:** Upsorn Praphamontripong
- **Office:** Rice Hall 206
- **Office hours:** TBD (until they are settled)
- **TAs:** TBD

**Class hours**
- Tuesday, Thursday 2:00PM - 3:15PM

**Overview**
The way web software is built has been gathering to direct customer sales (e-commerce). Many new technologies and frameworks styles you should use? What programs in great demand? How will you keep up? Companies prioritize the foundation of a software with several commonly used

You will work on user interface design, the front end development, back end development, and web-based information retrieval and processing. Over the semester, you will work on each programming assignment with a partner. Each assignment will be a step in creating a web application. You will be able to choose what to build, with the assignment constraining the features that must be used. By the end of the semester, you will have built a dynamic web application. With a large portion of the process that involves teamwork, interpersonal skills, and conflict management.
How A Browser See A Web Page

The browser receives an HTML page

It creates a model of the page and stores it in memory

It shows the page on screen using a rendering engine

The way web software is built has been rapidly changing. As use of the world wide web has shifted from information presentation to information gathering to direct customer sales (e-commerce) to enterprise applications, the amount and complexity of software has steadily been increasing. Many new technologies and frameworks have emerged everyday. Have you ever wondered what technologies, frameworks, or architectural styles you should use? What programming languages you should be familiar with? After you graduate, what languages will be popular or will be in great demand? How will you keep up with web development technologies? According to the UVA CS Industrial Advisory Board meeting, companies prioritize the foundation of web development over any specific technologies or frameworks.

This course will help you see how fundamental concepts of web development can be applied to develop reliable and usable web software regardless of the technologies or languages or frameworks. Although we put our emphasis on the concepts, you will develop dynamic web software with several commonly used programming languages and technologies.

You will work on user interface design, the front end development, back end development, and web-based information retrieval and processing. Over the semester, you will work on each programming assignment with a partner. Each assignment will be a step in creating a web application. You will be able to choose what to build, with the assignment constraining the features that must be used. By the end of the semester, you will have built a dynamic web application. With a large portion of the process that involves team work, interpersonal skills and conflict management.
Server Side Processing

UI implemented in a browser

Client

HTTP Request

data

Web server

Container engine

Program components

Server

HTTP Response

HTML

XAMPP
Architectural Styles

• How to partition a system
• How components identify and communicate with each other
• How information is communicated
• How elements of a system can evolve

Web architectural styles constantly change

The goal is to separate logic from presentation and to separate as many concerns in the logic as possible
Page-centric Design

Requests are made to a main component and the main component response to clients.
Requests are sent to a dispatcher that then forward the requests to another component (using *forward* or *redirect* control connection)
Model View Controller

• An abstraction frequently used in web app design
• Provide a way to divide the responsibilities of objects
• Decreases coupling between objects and layers (supports easier maintenance)
• Help divide the work (supports development expertise areas)
Model View Controller (2)

Model
- Encapsulates application state
- Responds to state queries
- Exposes application functionality
- Notifies views of changes

View
- Renders the model
- Requests updates from the model
- Sends user inputs to controller
- Allows controller to select view

Controller
- Defines application behavior
- Maps user actions to model updates
- Selects a view for response
- One view for each function

[Graphic from Designing Enterprise Applications with the Java 2 Platform, Enterprise Edition, Nicholas Kassem et al., October 2000]
Server-Side Scripting

- Generate HTML on the server through scripts
- Early approaches emphasized embedding server code inside HTML pages
- Examples: PHP, JSP

```php
<!doctype html>
<html>
<head>
<title>Login example</title>
</head>
<body>
    You logged in as <font color="green">b><?php echo $_POST["name"]; ?>\b</font> with password <font color="green">b><?php echo $_POST["pwd"]; ?>\b</font>
</body>
</html>
```

```
<html>
<head>
<title>Counting with a JSP</title>
</head>
<body>
    <!-- Set global information for the page -->
    <%@ page language="java" %>
    <!-- Declare the variable -->
    <%!
    int count = 0;
    %>
    <!-- Scriptlet - Java code -->
    
    for (int i = 0; i < 10; i++)
    {
        count = count+1;
    }
    
    The counter value is: <% out.println(count); %>
</body>
</html>
```
Server-Side Scripting Site

Advantages
- Server-side processing, browser independent, search optimization improvement, increased security

Disadvantages
- Poor modularity, hard to understand, difficult to maintain
Server-Side Framework

- Structure server into tiers, organizes logic into classes
- Execution on the server
- Can be single-page or multiple pages with page-centric design, dispatcher design, model-view-control design, or combination of any architectures
- Example: JSP (with separation of concerns)

![Diagram showing the layers of a server-side framework](image)

- **Presentation**: HTML and JSP
- **Logic**: Java classes (non-servlet)
- **Data**
  - Data content
  - Data representation
  - Data storage
  - Servlets and beans
  - Data structure, beans, Java classes
  - Database and files, Oracle, SQL
Server-Side Framework Site

Advantages
Separation of concerns, maintain and reuse

Disadvantages
Need to load an entire page to get new data
Front End Frameworks

• Client is organized into separate components, capturing model of web application data

• Components separate logic from presentation

• Components dynamically generate corresponding code based on component state

• Example: Angular
Front End Framework Site

Advantages
- Code organization, reuse, quick and easy to develop

Disadvantages
- Duplicate logic in client and server
**Single Page Application (SPA)**

- Client-side logic sends messages to server, receives response
  
- Logic is associated with a single HTML page, written in JavaScript
  
- HTML elements dynamically added and removed through DOM manipulation

**Enabling technologies**
- AJAX
- DOM Manipulation
- JSON
- Jquery
- Angular (include and routing are useful)

- Example: Gmail
Single Page Application Site

Advantages
- Fast (load most resources once; only data are transmitted)

Disadvantages
- Hard to maintain and reuse, cross-site scripting, memory leak
Summary: Web Apps

- Deploy across the Web
  - Other (software) deployment methods include bundling, shrink-wrapping, embedding, and contracting

- New technologies

- New conceptual language constructs for programming
  - Integration
  - Data management
  - Control connections

- Browser features

These differences affect every aspect of how to engineer high quality software