IT4409: Web Technologies and e-Services

Term 2020-1

Web Development Models

Instructor: Dr. Thanh-Chung Dao
Slides by Dr. Binh Minh Nguyen

Department of Information Systems
School of Information and Communication Technology
Hanoi University of Science and Technology

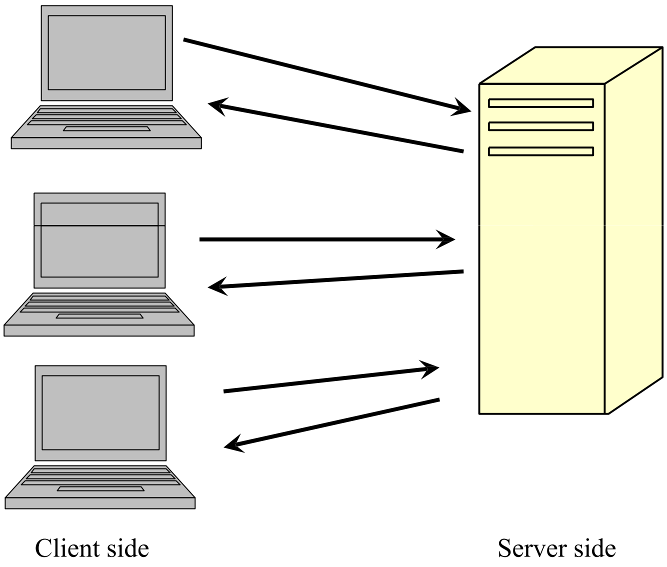
1

Content

- Web Application Architecture: client-server
- Programming Languages on client side
  - Javascript, Flash, Applet, ...
- Programming Languages on server side
  - PHP, Server page, Servlet, ...
- 3-tier architecture and MVC model

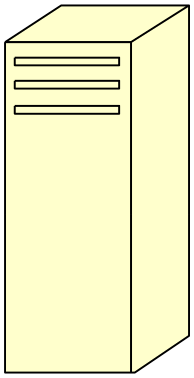
2

Client-Server Model



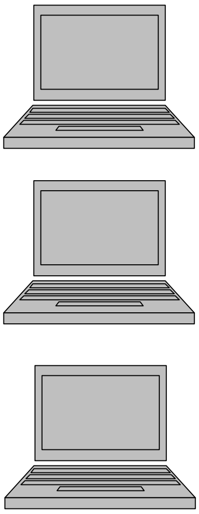
Server Roles

- Manage and store data, including:
  - User data
  - Application data
- Provide processing services for data
- Centralize data
- Manage user authentication, authorization mechanisms via login function



Client Roles

- Provide user interface
- Can store some small data (using cookie)
- Can process data (check validity of data that are entered by users)
    - Thin client: only provides user interface, centralize data processing on server side
    - Thick client: realizes data processing on client side
- Can be accessed from everywhere with minimal software installation

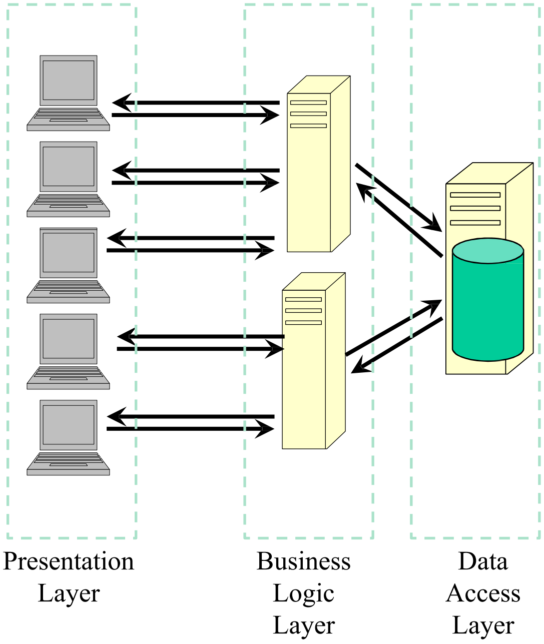


Client-Server Advantages

- Centralized storage and processing. Switch from CAPEX to OPEX
- No data redundancy
- Enhance the ability of sharing data
    - If data are distributed on multi-systems of users, it will cause difficulties in sharing the data because each system has its own database architecture

3-Tier Architecture

- Database Tier (Data Access Layer)
  - Stores and accesses data in low-level
- Server Tier (Business Logic Layer)
  - Manages application connections and process data
- Client Tier (Presentation Layer)
  - Provides interface and processing



7

3-Tier Architecture Advantages

- Centralized Database can be accessed by many servers at the same time
- Allow load balance of user connections on many application servers
- **Data Access Layer** is consistently designed with hardware in order to serve specific its tasks:
  - Data manipulations: update, insert, remove, etc.
  - Need more reliable hard drives
- **Business Logic Layer** are designed to provide connection points for user connections and run multi-applications
  - Need more computing power of CPU

8

Programming Languages

Client
Html
JavaScript
Flash

Server
Java, Ruby
Visual Basic
PHP, Perl
Python

Database
SQL
NoSQL

Client Programming Language

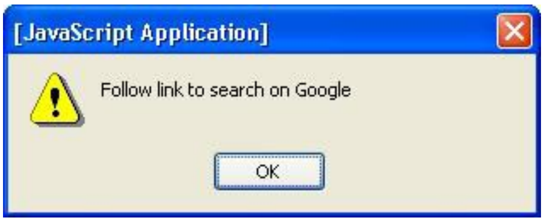
JavaScript

- Event Handling
- Statements (like C / Java)
- Operators
- Variables global (default)
  - Or local (e.g. var x = 1)
- Types can change
  - Eg. x = 1; x = ‘Hello’
- Function definition (reuse)
- Message Alerts
- Page element access with Document Object Model
  - Views HTML page as a tree of elements

Hello World Example

- This provides an annoying popup – try it!

```
<html>
<body>
<a href="http://www.google.co.uk"
  onMouseOver="(alert(
'Follow link to search on Google') )">
Search on Google
</a>
</body>
</html>
```



Follow link to search on Google

What are Applets?

- An ***applet*** is a special Java program that can be embedded in HTML documents.
- It is automatically executed by (applet-enabled) web browsers.
- In Java, non-applet programs are called ***applications***.

Application vs. Applet

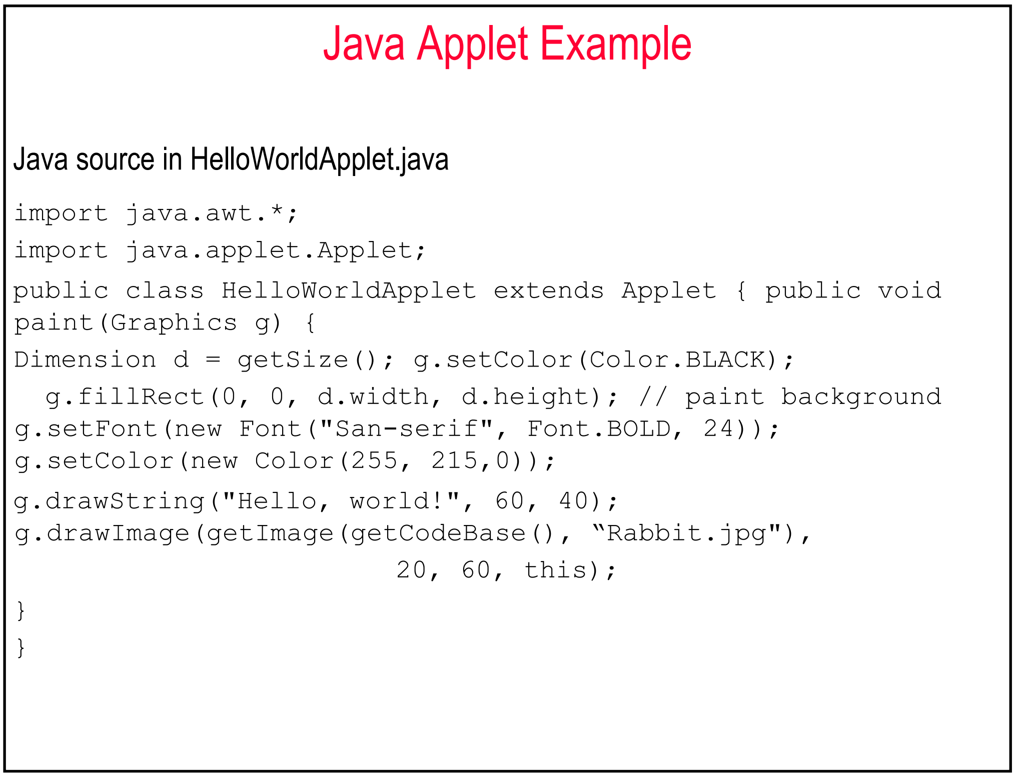
- Application
  - Trusted (i.e., has full access to system resources)
  - Invoked by Java Virtual Machine (JVM, java), e.g. `java HelloWorld`
  - Should contain a main method, i.e., `public static void main(String[])`
- Applet
  - Not trusted (i.e., has limited access to system resource to prevent security breaches)
  - Invoked automatically by the web browser
  - Should be a subclass of class `java.applet.Applet`

Java Application Example

```
HelloWorld.java

public class HelloWorld {
public static void main(String[] args) {
System.out.println(“Hello World!”);
}
}

HelloWorldApplet.java
```

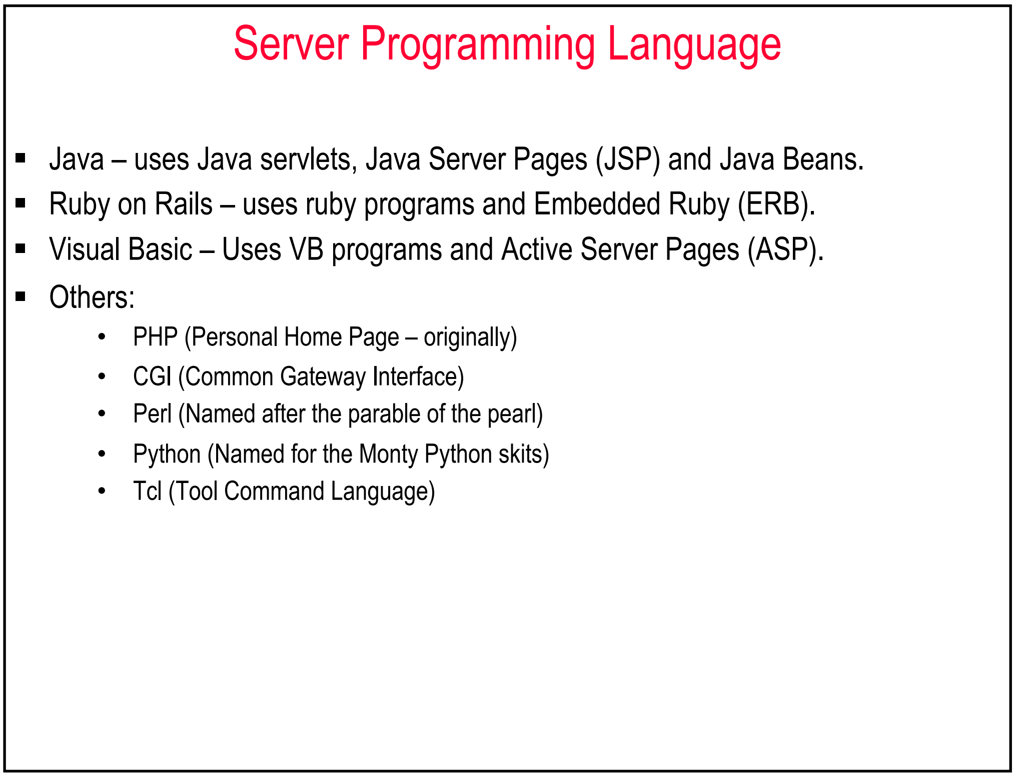


Java Applet Example

Java source in HelloWorldApplet.java

```
import java.awt.*;
import java.applet.Applet;
public class HelloWorldApplet extends Applet { public void
paint(Graphics g) {
Dimension d = getSize(); g.setColor(Color.BLACK);
  g.fillRect(0, 0, d.width, d.height); // paint background
g.setFont(new Font("San-serif", Font.BOLD, 24));
g.setColor(new Color(255, 215,0));
g.drawString("Hello, world!", 60, 40);
g.drawImage(getImage(getCodeBase(), “Rabbit.jpg"),
                        20, 60, this);
}
}
```

15



Server Programming Language

- Java – uses Java servlets, Java Server Pages (JSP) and Java Beans.
- Ruby on Rails – uses ruby programs and Embedded Ruby (ERB).
- Visual Basic – Uses VB programs and Active Server Pages (ASP).
- Others:
  - PHP (Personal Home Page – originally)
  - CGI (Common Gateway Interface)
  - Perl (Named after the parable of the pearl)
  - Python (Named for the Monty Python skits)
  - Tcl (Tool Command Language)

16

PHP

Hacky, but (also?) very c-like

Classes, etc., work very much like c/c++

Designed to work in the world of HTML

Is run-time interpreted by the web server

Reminder: it's hacky

17

Simple PHP Example

- PHP is meant to be invoked inline with content Page “escapes” into and out of a regular html document
- File extension is .php (was .php3 for version 3)

```
<html>
<head>Test page</head>
<body>
     The time is now
          <?php
echo date();
?>
          <hr>
</body>
</html>
```

18

JSP Example

```
<html>
<head> <title> Hello JSP </title> </head>
<body>
<p> Hello World:
   <%= new java.util.Date() %>
</p>
</body>
</html>
```

See also the Servlet this page is translated to

19

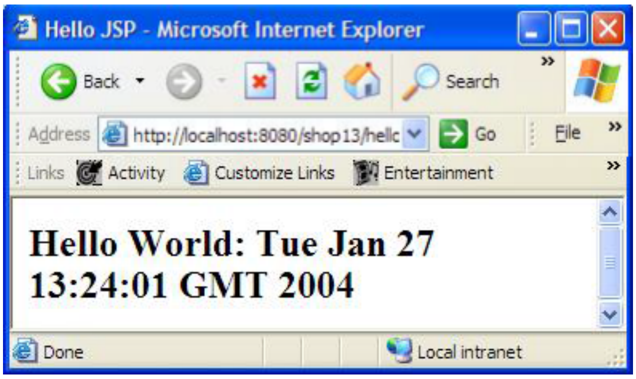
Date_jsp.java (extract)

This extract shows the part that produces the output – compare it with the JSP:

```
out = pageContext.getOut();
_jspx_out = out;
out.write("<html>\r\n");
out.write("<head> ");
out.write("<title> Hello JSP ");
out.write("</title> ");
out.write("</head>\r\n");
out.write("<body> \r\n");
out.write("<p> Hello World:\r\n     ");
out.print( new java.util.Date() );
out.write("\r\n");
out.write("</p>\r\n");
out.write("</body>\r\n");
out.write("</html>\r\n");
```

20

Produced



Hello World: Tue Jan 27 13:24:01 GMT 2004

21

MVC Development Model

- Architectural Pattern from Smalltalk (1979)
- Decouples data and presentation
- Eases the development

22

MVC – The Model

- The “Model” contains the data
- Has methods to access and possibly update it’s contents.
- Often, it implements an interface which defines the allowed model interactions.
- Implementing an interface enables models to be pulled out and replaced without programming changes.

24

MVC – The View

- The View provides a visual representation of the model.
- There can be multiple views displaying the model at any one time.
    - For example, a companies finances over time could be represented as a table and a graph.
    - These are just two different views of the same data.
- When the model is updated, all Views are informed and given a chance to update themselves.

25

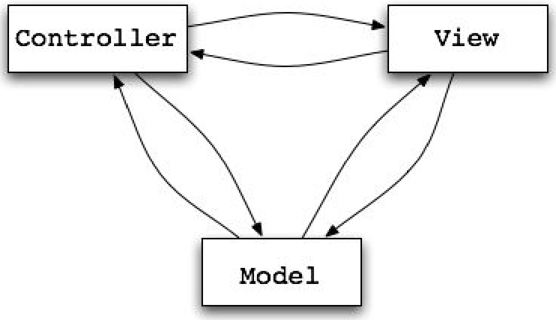
MVC – The Controller

- It interprets mouse movement, clicks, keystrokes, etc
- Communicates those activities to the model – eg: delete row, insert row, etc

26

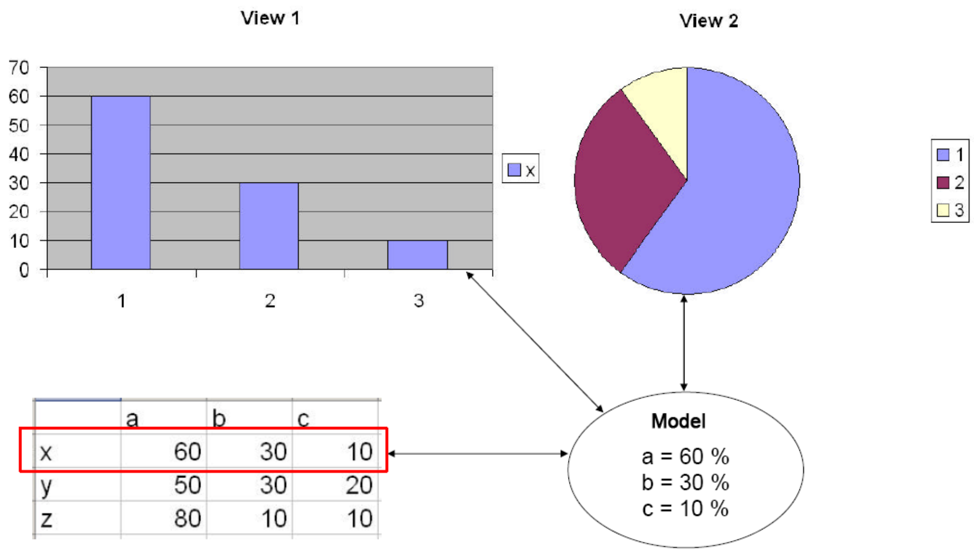
Example Control Flow in MVC

- User interacts with the VIEW UI
- CONTROLLER handles the user input (often a callback function attached to UI elements)
- CONTROLLER updates the MODEL
- VIEW uses MODEL to generate new UI
- UI waits for user interaction



27

MVC – General Example



	a	b	c
x	60	30	10
y	50	30	20
z	80	10	10

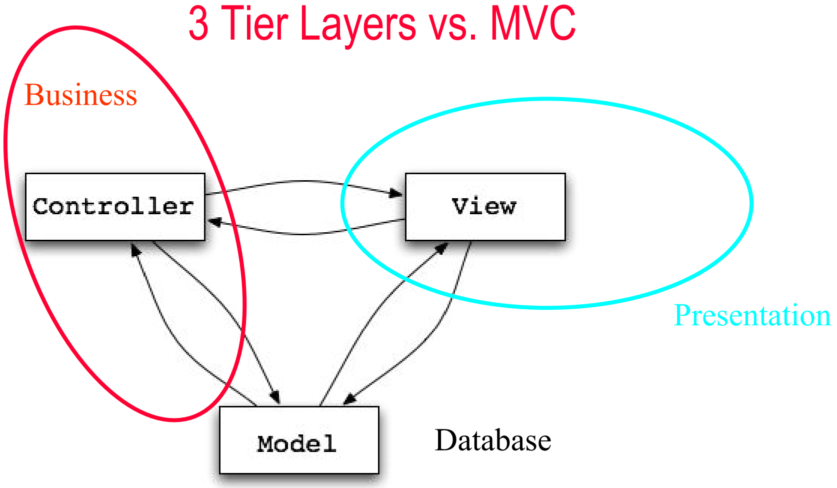
Model

a = 60 %
b = 30 %
c = 10 %

MVC Advantages

- MVC decouples the model, view, and controller from each other to increase flexibility and reuse.
  - You can attach multiple views to the model without rewriting it.
  - You can change the way a view responds to user input without changing the visual presentation. For example, you might use a pop-up menu instead of keyboard command keys.

3 Tier Layers vs. MVC



Presentation:

- View is the user interface (e.g. button)
- Controller is the code (e.g. callback for button)

Data:

- Model is the database

30

Summary

- Client-Server Model
- 3-Tier Architecture
- Dynamic Web Programming Languages
- MVC Model

31

email: chungdt@soict.hust.edu.vn

Q&A