

Turbine: Building Model 2+1 Web Applications

9/24/00

[Click here to start](#)

Table of Contents

[Turbine: Building Model 2+1 Web Applications](#)

[Outline](#)

[What is Model 1?](#)

[Picture of Model 1](#)

[What is MVC?](#)

[Model - Core of the Application](#)

[View - Presentation Layer](#)

[Controller - Input Mechanism](#)

[A Picture of MVC](#)

[What is Model 2?](#)

[Picture of Model 2](#)

[Problems with Model 2](#)

[A JSP solution to Model 2 Problems](#)

[What is Turbine?](#)

[Problems Turbine Solves](#)

[Turbine Features 1](#)

[Turbine Features 2](#)

[Turbine Features 3](#)

[Turbine Features 4](#)

Author: Jon

Email: jon@apache.org

Home Page: <http://java.apache.org/>

[Turbine Features 5](#)

[Turbine's Community](#)

[Turbine's Size](#)

[What is Model 2+1?](#)

[How is Turbine Model 2+1?](#)

[Turbine and MVC](#)

[Picture of Turbine Modules](#)

[Picture Turbine Execution](#)

[Java Directory Structure](#)

[Template Directory Structure](#)

[Screen: HelloWorld.java](#)

[Template: HelloWorld.wm](#)

[Template: default.wm](#)

[Template: TopNav.wm](#)

[Template: BottomNav.wm](#)

[Example Output](#)

[Actions - SampleScreen.wm](#)

[Actions - Java Code](#)

[Turbine Developers Kit \(TDK\)](#)

[TDK Benefits](#)

[URL's](#)

Turbine: Building Model 2+1 Web Applications

By: Jon S. Stevens
jon@apache.org



Slide 1 of 40

Outline

- What is Model 1?
- What is MVC?
- What is Model 2?
- What is Turbine (Model 2+1)?
- How do I build applications with Turbine? (TDK)



Slide 2 of 40

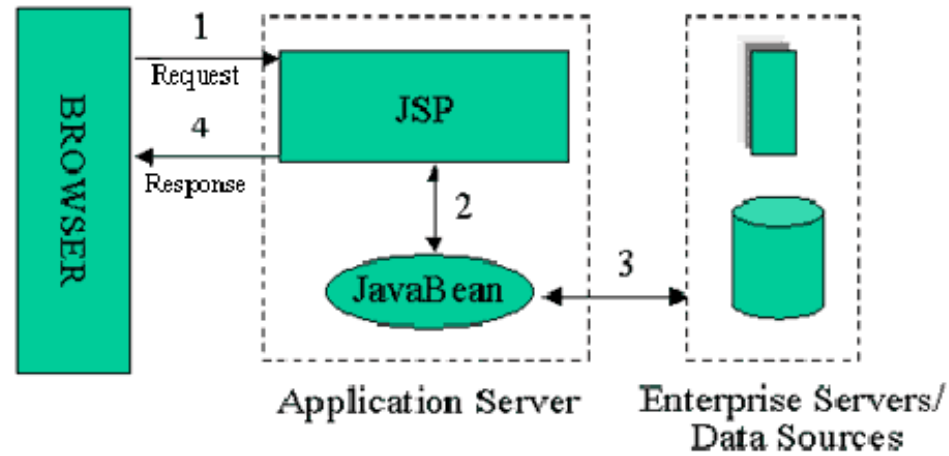
What is Model 1?

- Model 1 is the current method of creating sites using JSP/Servlets. A linear and non-expandable approach.
- Flow works like this:
 - Request comes in
 - JSP page executes Beans to do processing
 - Result is returned



Slide 3 of 40

Picture of Model 1





Slide 4 of 40

What is MVC?

- Model-View-Controller
- Abstraction of each layer of development
- Model - Application Logic
- View - Presentation Logic
- Controller - Business Logic



Slide 5 of 40

Model - Core of the Application

- Application Logic
 - database accesses
 - data crunching/munging
 - Separation of UI from application
- Not dependent on input or output formats



Slide 6 of 40

View - Presentation Layer

- Responsible for displaying the combined work of the Controller and Model
- UI portion of the page
- Formats data to be presentable to the end user



Slide 7 of 40

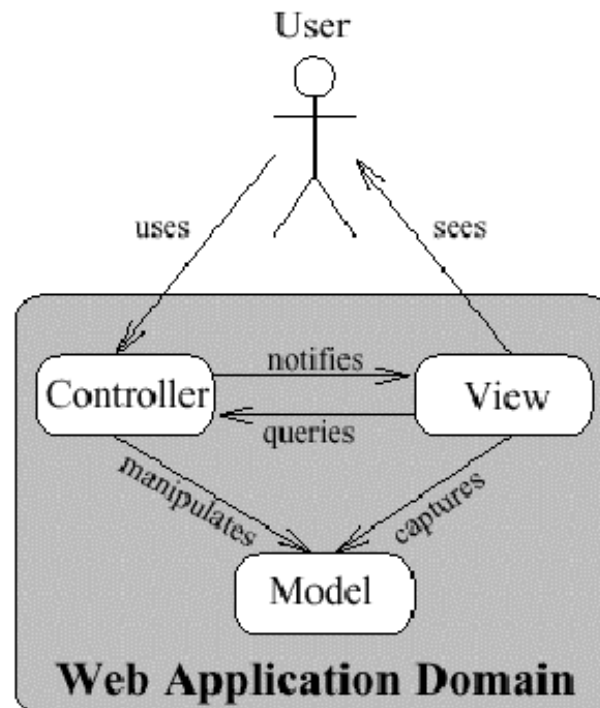
Controller - Input Mechanism

- Listens to what the User requests
- Supplies and communicates information to the Model and the View



Slide 8 of 40

A Picture of MVC



Copyright Lutris Technologies



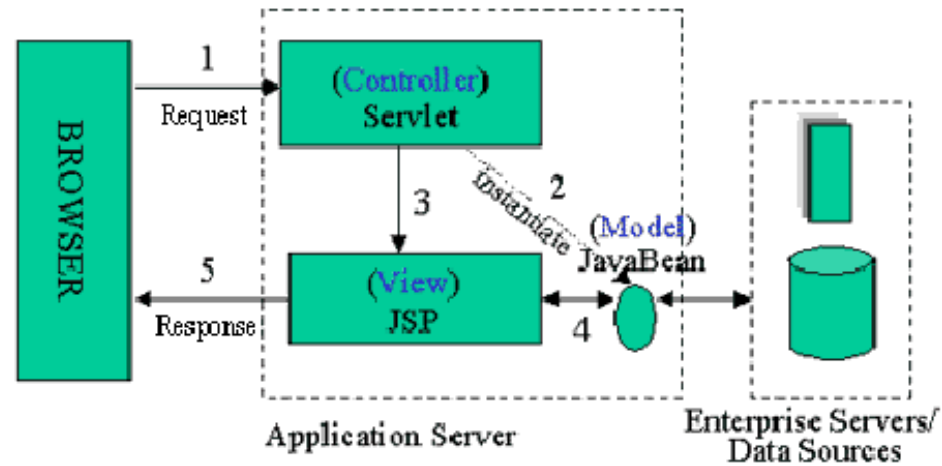
Slide 9 of 40

What is Model 2?

- It is a server side implementation of MVC using Java Server Pages (JSP)
- Flow works like this:
 - Request comes in
 - Servlet (Controller) is executed to build a Context (JavaBean)
 - Context is passed to JSP template to provide the View
 - Result is returned



Picture of Model 2





Slide 11 of 40

Problems with Model 2

- Does not address the issue of providing a consistent *View* across the application
- Does not mention a framework for reusable code.
- Does not address issue of tying the Controller to the View
- Does not address issue of receiving input from the View
- Needs to be more web application friendly



Slide 12 of 40

A JSP solution to Model 2 Problems

- For those of you who refuse to use anything other than JSP, the Struts project is the right way to go for now.
- [Http://jakarta.apache.org/struts/](http://jakarta.apache.org/struts/)



Slide 13 of 40

What is Turbine?

- Turbine is a servlet based framework that allows experienced Java developers to quickly build secure web applications.
- A web based application is an application where users use their favorite web browser in order to access secure business logic.



Slide 14 of 40

Problems Turbine Solves

- Turbine attempts to not only provide a framework for development, but it also tries to provide all of the code that you would normally end up developing yourself anyway.
- Turbine is trying to be the wheel. Anything else is re-inventing that wheel.
- Turbine has a Module architecture for developing web applications.



Slide 15 of 40

Turbine Features 1

- Integration with template systems: WebMacro, Velocity, Cocoon, FreeMarker, JSP, ECS
- Utility code for working with WM/Velocity, such as a SelectorBox class for building <select> boxes
- Single Entry Point Servlet model for optimal security and control
- Singleton based Database Connection Pool (JDBC) with built in support for all of the major databases
- ParameterParsing for GET/POST/PATH_INFO
- Event based Action handling!



Slide 16 of 40

Turbine Features 2

- Strict MVC guidelines implemented through many interfaces and abstract classes as well as through the template systems.
- Integration with 5 Object-Relational tools: DatabaseMap/Peers, Village, Town, OPaL, [Castor](#)
- IDBroker that abstracts the autoinsert/sequence usage from the database
- DatabaseMap generation tool that will read a schema and generate the OR Java classes for you
- Failsafe Job based scheduler system (ie: a Java based cron)



Slide 17 of 40

Turbine Features 3

- GlobalCache (Singleton based system for caching data across servlets and requests)
- DateSelector (utility for building the HTML for popup date menus)
- File Upload API
- Generic Services API for creating Singletons
- Castor Service Integration
- [XML-RPC](#) Service Integration
- Localization Services API
- JNDI Services API



Slide 18 of 40

Turbine Features 4

- Visitor/Member API for managing users
- Temporary and Permanent storage/management for users session data
- ACL (Access Control List) based security system that uses Roles and Permissions (and the database schemas)
- BrowserDetector class for determining which browser someone is using
- GenerateUniqueld class for getting a unique number (ie: for sessions)
- Logging to a file facility.



Slide 19 of 40

Turbine Features 5

- Property file reader tools and a TurbineResources service to make it easy to retrieve properties
- Integration with JavaMail to make sending email painless
- Integration with JavaMail and Velocity to allow you to send processed Velocity templates as email!
- Built using Ant
- Initial application development WAR archive creation tool using Ant
- Works cleanly with Servlet API 2.0 and higher
- Works with JDK 1.1.x and higher - 100% Pure Java



Slide 20 of 40

Turbine's Community

- Turbine has a mailing list with over 360 people on it
- There are over 30 people contributing code and documentation
- Used by people all over the world (many foreign contributors)
- Used as the basis for other large OSS projects like Jetspeed



Slide 21 of 40

Turbine's Size

- Turbine now has almost 300 classes (and 5 slides of features!)
- Not all parts of Turbine have to be used
- Pick the parts you want and ignore the rest...until you find you need it. :-)

Template: HelloWorld.wm

```
$page.setTitle("Welcome to ApacheCon EU 2000")
```

This is an example WebMacro template!

```
<p>
```

Here is what was put into the context:

```
<p>
```

```
<b>$hello</b>
```

What is Model 2+1?

- Solves the problems of Model 2.
- Is not Model 3, but is instead Model 2 plus feature additions.
- Is a funny buzzword that I invented for this talk and has now caught on.



Slide 23 of 40

How is Turbine Model 2+1?

- Adds lots of reusable code
- Interfaces (and implementations) for Security, User Management and Services
- Turbine Modules for tying MVC together
 - Actions and ActionEvents
- Designed to be used as a webapp framework



Slide 24 of 40

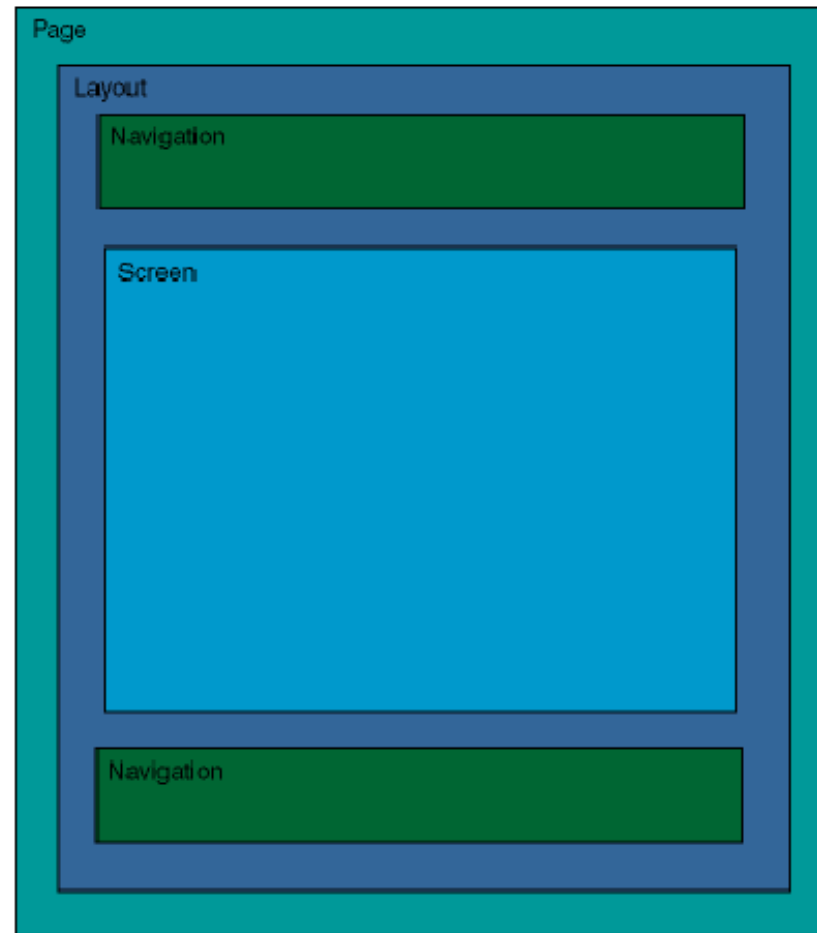
Turbine and MVC

- Turbine blurs the MVC model a bit
- Turbine provides the larger container for building MVC applications
- View is managed by one of the template solutions, but controlled by Turbine



Slide 25 of 40

Picture of Turbine Modules





Slide 26 of 40

Java Directory Structure

- **modules/**
 - **screens/**
 - HelloWorld.java
 - **actions/**
 - MyAction.java
 - **navigations/**
 - No file needed right now.
 - **layouts/**
 - No file needed right now.



Slide 28 of 40

Template Directory Structure

- **templates/**
 - **screens/**
 - MyScreen.wm
 - **actions/**
 - MyAction.wm
 - **navigations/**
 - TopNav.wm and BottomNav.wm
 - **layouts/**
 - Default.wm



Slide 29 of 40

Screen: HelloWorld.java

```
public class HelloWorld extends WebMacroSiteScreen
{
    public void doBuildTemplate( RunData data, Context context )
        throws Exception
    {
        // the context object has already been setup for you!
        context.put ("hello", "this is a test...");
    }
}
```



Slide 30 of 40

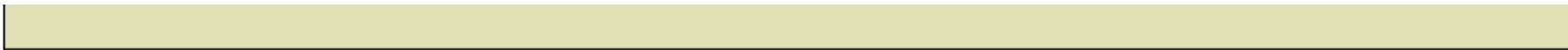


Slide 31 of 40

Template: default.wm

```
$page.setStyleSheet("/myapp/style.css")  
$page.setBgColor("#ffffff")  
$navigation.setTemplate("/TopNav.wm")
```

```
<table width="100%">  
  <tr>  
    <td>$screen_placeholder</td>  
  </tr>  
  <tr>  
    <td>$navigation.setTemplate("/BottomNav.wm")</td>  
  </tr>  
</table>
```



Slide 32 of 40

Template: TopNav.wm

This is the Top Navigation

<hr>



Slide 33 of 40



Slide 22 of 40

Template: BottomNav.wm

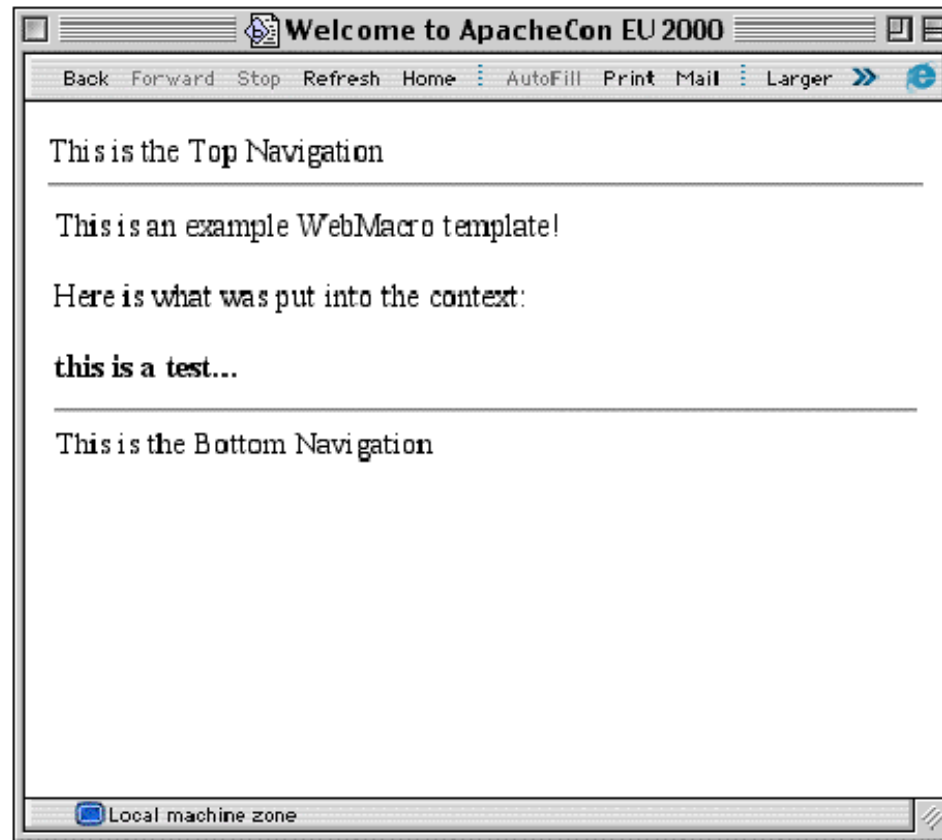
<hr>

This is the Bottom Navigation



Slide 34 of 40

Example Output





Slide 35 of 40

Actions - SampleScreen.wm

```
<form action="$link.setPage("Next.wm")" method=post>  
<input type="hidden" name="action" value="MyAction">  
<input type="text" name="f1" value="">  
<input type="submit" name="eventSubmit_doSubmit"  
value="Hit me!">  
</form>
```



Slide 36 of 40

Actions - Java Code

```
public class MyAction extends WebMacroSiteAction
{
    public void doSubmit (RunData data, Context context)
        throws Exception
    {
        String text = data.getParameters().getString("f1", "blank");
        context.put ("username", text );
        data.setMessage("Done!");
    }
    public void doPerform(RunData data, Context context)
        throws Exception
    {
        data.setMessage("Button not found!");
    }
}
```



Slide 37 of 40

Turbine Developers Kit (TDK)

- Everything you need to start developing webapps with Turbine out of the box
- Everything is pre-configured. Just start Tomcat and go.
- Includes:
 - Tomcat
 - Turbine
 - Lots of .jars
 - Documentation



Slide 38 of 40

TDK Benefits

- Easily create a webapp with the directory structure and sample pages already in place
- All the external .jars are already there
- WebServer/Servlet engine
- Easily upgraded
- 100% Free



Slide 39 of 40

URL's

- <http://java.apache.org/turbine/>
- <http://java.apache.org/turbine/tdk/>
- <http://jakarta.apache.org/velocity/>
- <http://www.webmacro.org/>



Slide 40 of 40