A Study of MVC – A Software Design Pattern for Web: Application Development on J2ee Architecture

Sharan Palleboina
Department of Computer Science
University of Bridgeport

Prof.Tarik Eltaeib
Department of Computer Science
University of Bridgeport
Email: teltaeib@my.bridgeport.edu

***Abstract*—The Model-View-Controller design pattern is justified as the structural base for many Java web development frameworks. This paper analyses the defects in the web frameworks, and proposes a separate Web-MVC pattern that more accurately describes how MVC is implemented in web frameworks. The MVC is very useful for constructing dynamic software systems. The MVC design will be created without changing the code of the application. This paper brings the plan of Web-Application Partitioning, and programming model that allows implementing the Model View Controller design pattern in a partition-independent pattern.**

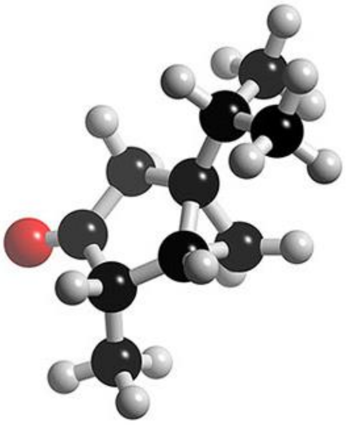
INTRODUCTION

Model View Controller is the design pattern for the architecture of many web applications in J2EE. Many languages have implemented the frameworks and adopted them universally. Basics Components of MVC model.

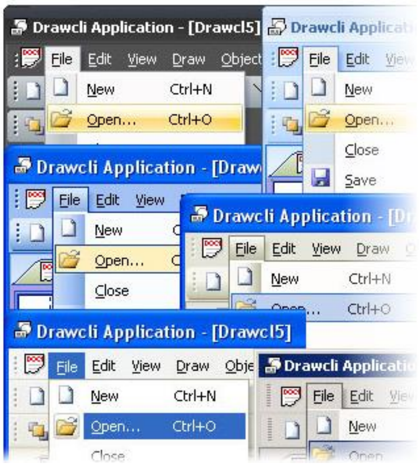
- A **model** is an object representing data or even activity, e.g. a database table or even some plant-floor production-machine process.
- A **view** is some form of visualization of the state of the model.
- A **controller** offers facilities to change the state of the model.

Smalltalk provides mechanisms to link models, views and controllers.

Model:



View:



Controller:

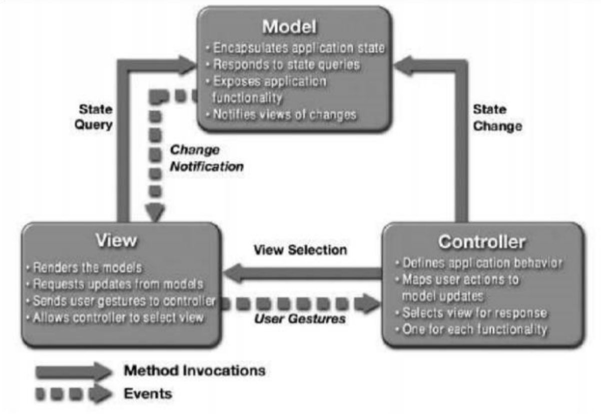

Model
• Encapsulates application state
• Responds to state queries
• Exposes application functionality
• Notifies views of changes
State Query
Change Notification
State Change
View
• Renders the models
• Requests updates from models
• Sends user gestures to controller
• Allows controller to select view
View Selection
User Gestures
Controller
• Defines application behavior
• Maps user actions to model updates
• Selects view for response
• One for each functionality
Method Invocations
Events

Fig 1. Generic MVC Structure

MVC Design pattern plays a major role for the design of web applications like J2EE, .Net, Rails and Struts etc.

J2EE and Java server page technologies are the fundamentals for Struts 2 framework. This Struts 2 framework consists of MVC pattern as follows.

a. User Interface component as views

b. Application logic as Model

c. Control functions as Controller

The view component of struts 2 framework is done by embedding JSP tags which provides mixed functionalities like flow control, accessing model component and effectual HTML forms structures. The controller component is corporeal with java classes for the actions to be implied. Each action has a responsibility to validate user Input and to engineer transaction processing by invoking appropriate model operations.

The XML configuration file or the java annotation mechanisms are used to outline and configure the actions. Such information is used to control the flow of web applications by finding the outcomes of each action. Value stack eliminates much of the tasks involved in handling HTTP requests and provides the information for JSP to display. It is the key factor which contains the information between view and controller and converts when needed. This section describes how the MVC is being represented in the web application frameworks. It also reflects the evolutionary changes in the web frameworks.

The primary responsibilities of MVC-Web model are:

1. It has to maintain a database for the data persistence

2. It has to execute the application logic that operates on the application state called transaction processing.

3. It has to manage interactions with external agents such as web services known as External Interface.

4. It should handle query to provide the information to view and controller elements in response for queries.

The primary responsibilities of the view component are:

1. It is used for information retrieval and display because it displays information to the user based on the query in the model.

2. It provides input forms and controls for the user to interact with the application.

3. It provides interactive dynamic behavior at the client side for the users.

The primary responsibilities for the MVC-Web Controller are:

1. It receives the incoming request and routes them to appropriate handler.

2. It receives the request parameters and handles the action such as invoking appropriate model elements.

3. It provides the response for the request depending upon the action invoked.