The Model–View–Controller Architecture and Struts2

Sistemas Distribuídos 2014/2015

Tiered architectures

Consider the following functional decomposition:

- **presentation logic**, which is concerned with handling user interaction and updating the view of the application as presented to the user;
- **application logic**, concerned with the detailed application-specific processing associated with the application (also referred to as the business logic, although the concept is not limited only to business applications);
- **data logic**, concerned with the persistent storage of the application, typically in a database management system.

Tiered architectures

- In the **two-tier** solution, the three aspects must be partitioned into two processes, the client and the server.
  - Most commonly done by splitting the application logic, with some residing in the client and the remainder in the server.
- In the **three-tier** solution, there is a one-to-one mapping from logical elements to physical servers.
  - The first tier can be a simple user interface allowing support for thin clients. The third tier is often simply a database offering a relational service interface.
- This approach generalizes to **n-tiered** (or multi-tier) solutions where a given application domain is partitioned into n logical elements, each mapped to a given server element.

Thin clients

- The trend in towards moving complexity away from the end-user device towards services in the Internet.
- This trend has given rise to interest in the concept of a **thin client**, enabling access to sophisticated networked services.
- The term thin client refers to a software layer that supports a window-based user interface that is local to the user while executing application programs or, more generally, accessing services on a remote computer.

The Model–View–Controller Architecture

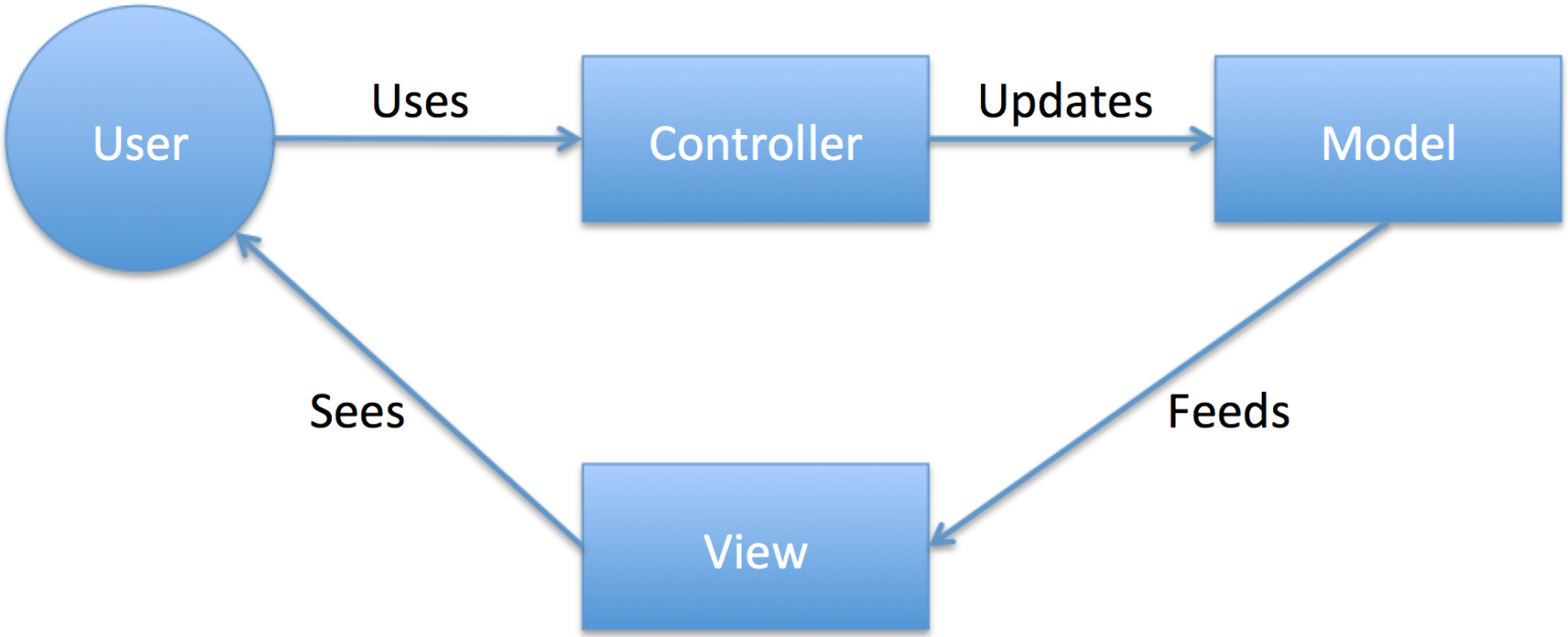
Model–View–Controller

- Model–View–Controller architectural pattern: MVC
- MVC is a pattern for implementing software applications with rich user interfaces.
- Major programming languages adopted MVC for Web applications
  - Although MVC existed before in the desktop environment...

MVC

- A **model** captures data-related behaviour, focusing on data, logic, and application rules.
- A **view** is a representation of information, such as a Web page with charts, pictures, and text (the same information may have multiple views).
- A **controller** receives input from the user, sends commands to the **model** and decides which **view** should be presented as output to the user.

MVC

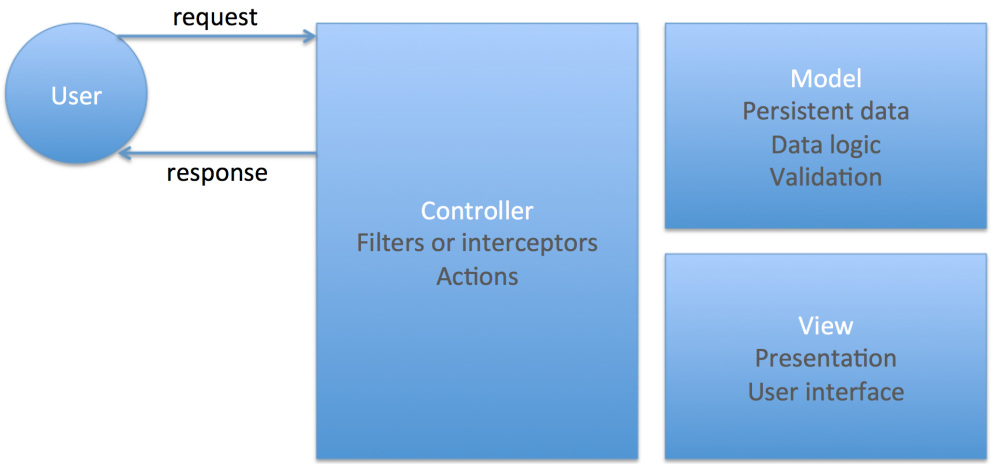


Advantages of MVC

- Separates implementation concerns
  - Persistent data and data logic
  - Presentation
  - Application logic
- Less code duplication
- Simplifies maintenance (updates, modifications)
- Simpler to test software units

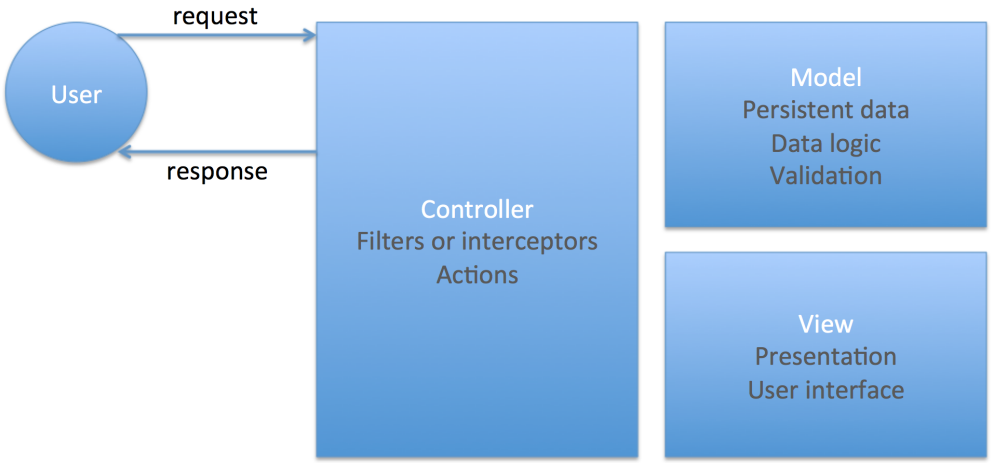
M – Model

- The model deals exclusively with data logic and rules.
- Responsibilities:
  - DB querying
  - Inserting records
  - Updating information
- Data behavior independent of presentation



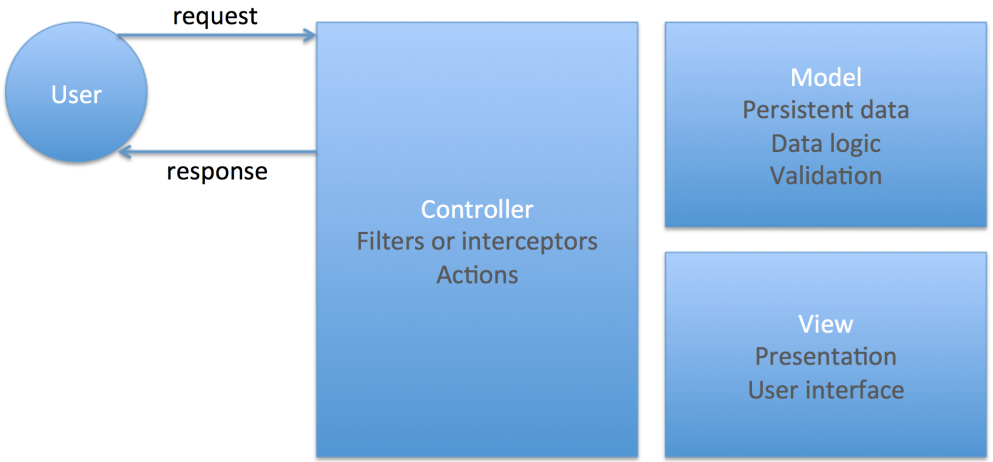
V – View

- The view implements the presentation layer
- Displays information to clients
- Concerned with how results and interface are presented to the user



C – Controller

- Receives requests from user
- Decides how the model is updated
- Decides which view is presented to the user
- Serves as the connection between the user, the data logic, and the presentation logic



MVC frameworks

- Model-view-controller (MVC) is a pattern intended to separate data logic (model), presentation logic (view) and business logic (controller).
- As an architectural pattern, anyone may design according to MVC, *e.g.*, JavaBeans (M), and JSPs (V), Servlet (C).
- Several frameworks build upon it:
  - Struts2
  - Spring
  - JavaServer Faces
  - Stripes
  - Wicket
  - Play!
  - Tapestry
  - Ruby on Rails
  - ...

Thymeleaf

Sistemas Distribuídos 2022/23

Thymeleaf

- Thymeleaf is a modern server-side Java template engine for both web and standalone environments.
- Thymeleaf's main goal
  - bring elegant natural templates to the development workflow — HTML that can be correctly displayed in browsers and also work as static prototypes, allowing for stronger collaboration in development teams.

Natural templates

- HTML templates written in Thymeleaf still look and work like HTML, letting the actual templates that are run in your

```
<table>
  <thead>
    <tr>
      <th th:text="#{msgs.headers.name}">Name</th>
      <th th:text="#{msgs.headers.price}">Price</th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="prod: ${allProducts}">
      <td th:text="${prod.name}">Oranges</td>
      <td th:text="${#numbers.formatDecimal(prod.price, 1, 2)}">0.99</td>
    </tr>
  </tbody>
</table>
```

Who is using

Auchan RETAIL FRANCE

broadleaf commerce

CAS

connect

ERKO INFORMATIK

enonic

Lagerwey

ppi

sahibinden.com

Travelcompositor COMPOSITOR

VEDA HR Software that connects

History

- Thymeleaf is a Java XML/XHTML/HTML5 template engine that can work both in web (servlet-based) and non-web environments.

FAQs

- Is Thymeleaf a web framework?
  - No, it is a template engine.
- What types of templates can Thymeleaf process?
  - HTML (HTML5, XHTML 1.0/1.1, HTML 4)
  - XML
  - TEXT (plain text)
  - JAVASCRIPT (.js files)
  - CSS (.css files)

FAQs

- Can Thymeleaf be used as a complete substitute for JSP and JSTL?
  - Yes.
- Can it be used outside web applications in non-web environments?
  - Yes it can. Although Thymeleaf (especially its Standard dialects) offers many features that are especially useful in web environments, it can be used for processing non-web HTML or XML documents (data XML, for example) or other types of templates that are not meant for being sent via HTTP (for example, text/HTML email content).
- I don't use Spring at all. Can I still use Thymeleaf?
  - Absolutely. Thymeleaf offers nice integration with Spring MVC through its SpringStandard dialect (included in the thymeleaf-spring3, thymeleaf-spring4 and thymeleaf-spring5 packages), but Spring integration is completely optional and the Standard dialect is in fact meant to be used without Spring.

Dialects

- Thymeleaf is an extremely extensible template engine (in fact it could be called a template engine framework) that allows you to define and customize the way your templates will be processed to a fine level of detail.
- An object that applies some logic to a markup artifact (a tag, some text, a comment, or a mere placeholder if templates are not markup) is called a processor, and a set of these processors – plus perhaps some extra artifacts – is what a dialect is normally comprised of. Out of the box, Thymeleaf's core library provides a dialect called the Standard Dialect, which should be enough for most users.

Attribute Processors

- Most of the processors of the Standard Dialect are attribute processors. This allows browsers to correctly display HTML template files even before being processed:
- <input type="text" name="userName" value="James Carrot" th:value="${user.name}" />
- This helps your designer and developer to work on the very same template file and reduce the effort required to transform a static prototype into a working template file.