

COMP519 Web Programming

Lecture 30: Summary and Outlook)

Handouts

Ullrich Hustadt

Department of Computer Science
School of Electrical Engineering, Electronics, and Computer Science
University of Liverpool

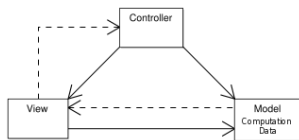
Contents

1 Summary

2 Outlook

Distributed Systems: Model-View-Controller

We use the **Model-View-Controller** software design pattern to discuss some of the fundamental question we need to answer when developing **distributed systems**

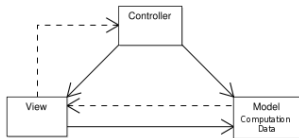


- The **model** manages the **behaviour** and **data**
- The **view** renders the **model** into a form suitable for interaction
- The **controller** receives user input and translates it into instructions for the **model**

- ❶ Where should the **view** be **rendered**?
- ❷ Where should the **behaviour** of the **model** be computed?
- ❸ Where should the **data** for the **model** be held?

Distributed Systems: Model-View-Controller

We use the **Model-View-Controller** software design pattern to discuss some of the fundamental question we need to answer when developing **distributed systems**

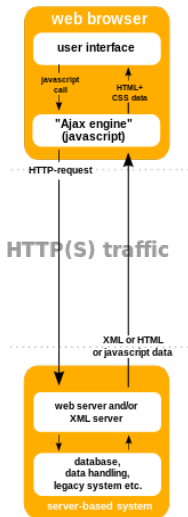


- The **model** manages the **behaviour** and **data**
 - The **view** renders the **model** into a form suitable for interaction
 - The **controller** receives user input and translates it into instructions for the **model**
-
- Assignment 2: Everything is on the client-side
 - Assignment 3: **View** is client-side, **Controller** and **Model** are on the server-side
 - Assignment 4: **Model** only, with a specific API (View is for testing purposes only)

Distributed Systems: Web Applications

- Web applications almost always combine **client-side scripting** and **server-side scripting**
- **Ajax** (Asynchronous JavaScript and XML) is a set of techniques for **sending and retrieving data** from a server (asynchronously)
- On the server-side often **PHP scripts** act as mediators that retrieve **data** from a **database** in response to **Ajax requests**, possibly in the form of a **web service**
- **Data** is typically exchanged in **JSON** (JavaScript Object Notation) format

Ajax model of a web application



By DanielSHaicht,
via Wikimedia Commons
<https://commons.wikimedia.org/wiki/File:%3AAjax-vergleich.svg>,
CC BY-SA 3.0.
<https://commons.wikimedia.org/w/index.php?curid=29724785>

What next?

- Development of applications typically does not start from scratch
 ~> modules and libraries / frameworks are used

- PHP frameworks

Laravel

Zend Framework

CodeIgniter

Phalcon

Symfony

FuelPHP

CakePHP

PHPPixie

- JavaScript frameworks

jQuery

Ember.js

Angular (Google)

Node.js

React (Facebook)

Mithril

Vue.js

Polymer

- Being able to use a framework is a skill in itself
 Popularity / use of frameworks changes quite frequently
 ~> not clear which ones to teach / learn