Web A2 Q1

by Matthew Abraham

Submission date: 17-Mar-2018 11:35PM (UTC-0400)
Submission ID: 931813273
File name: 45586_Matthew_Abraham_Web_A2_Q1_1553_1919972637.docx
Word count: 1247
Character count: 6779

Model View Controllers: Analysis and Implementation For Web Applications

Matthew Abraham
Department of Computing and Internet Technology
University of the West Indies
St. Augustine, Trinidad & Tobago
m.abrahamtt@gmail.com

***Abstract*—The Model View Controller is an architectural pattern used to separate the logical and semantic functions of client server application. There are clear benefits of separating the various goals of an application but facilitating a paradigm has always added additional effort and complexity into the development processes. There may also be ways to avoid the developmental cost of creating an Model View Controller framework for each project that requires it. In the context of web application development, its use of asynchronous DOM manipulation makes it an efficient client server framework.**

Keywords—model, view, controller, interface, coupling, architecture, encapsulation, abstraction, dependencies

I. Introduction

Model View Controller (MVC) architecture is comprised of three components which each work to achieve the various goals of an application. As a software developer, modularity and encapsulation allows great flexibility in the manipulation of the program. The Model contains core attributes and functions, view displays information, and controllers manage input [1]. These functionalities result in a more easily maintained application [2]. Due to the separation of the three core goals: processing, input and output, a model will contain the core of the application, and will perform the specific procedures, which are called on by the controller on behalf of the user [1].

II. Model Controller View

A. Advantages of Using MCV

The purpose of MVC is to build a flexible system that responds to varying user interface preferences and requirements. To achieve this, there may be multiple Controller-View pairs [1]. In the context of web programming, the controller; which implements the interface with the server, receives requests. This results in processing via the model and allows the view to dynamically update the page contents and data. Effectively encapsulating HTML code from program code [2]. This functionality essentially removes one downfall of static web pages; the need for a user to update or "pull" an update from the core code via a web page refresh. If an administrator makes a change to the model, the views (and controllers coupled) would receive the update message; which improves error tolerance as well as removing outdated information for the user in real time [3].

B. Disadvantages of Using MCV

Navigating the framework may become tedious with the addition of new layers of abstraction. The MVC architecture setup costs can be expensive; this is due to the inclusion of necessary dependencies and libraries required, as well as maintaining up to date server technologies [2]. There are various developed MVC frameworks that can be implemented, but in the necessity of a customized environment, the development will in turn increase training costs [2]. On the other hand, using an existing MVC framework (for Struts and JSF, see II), imposes a version requirement on the libraries used within the application. Due to the developmental time of implementing an MVC which are similar to standard architecture, it is best suited for larger projects, where the loose coupling between the model and view do not impose many restrictions, should the core code require any incremental changes [2].

III. Implementation Of MVC

A. PHP Implementation

The implementation of MVC using PHP is relatively similar to other languages, in that there must be PHP defined objects which handle the various functions of the three modules of this architecture. An object class which is used to define all domain application processes and functionality, that is, what data to process and how to process it. In PHP, it is just as simple to inject HTML as it is in Python or JavaScript while utilizing objects. In practice, the object that constructs the display for the user references some instance of the model object before outputting it. For example, a domain model object which caches data from a given table in a database (where the table to be displayed may be dynamically changed), a view object which injects HTML code to create a table after loading the data found in this instance of the model object. Lastly, the controller object which manages user input can be achieved similarly to the view object by using PHP code to create a class which injects HTML code. Following the example, the user may interact with the frontend controller to change the table of the database

being viewed, the request is received by the model, and the update propagates in the view object.

B. *Alternative Frameworks To Clean MVC*

"Clean MVC" refers to the implementation of MVC architecture without the use of an existing MVC or MV* framework to develop the application, but the use of entirely customised MVC development.

There are various frameworks for implementing MVC, the most popular are: Backbone JS (Backbone), Angular JS (Angular), Ember JS (Ember) and Knockout JS (Knockout). Backbone is a minimal solution to separating development concerns but as a consequence, without its own plugins acts more like a utility library. Angular has a built in HTML compiler for custom domain tags and provides alternative means to DOM manipulation. Ember is an intuitive framework that excels at streamlining the start-up point of your application compared to other frameworks. Knockout simplifies dynamic interfaces using MVVM; easier to build even complex interfaces with custom behaviours are easily implemented. Backbone is best suited for development on programs that require incremental developing to better understand obscure requirements. Angular allows for extensive testing and verification and tests the framework in an appropriate manner. Ember is the means for fast prototype development and seamless integration with little effort. Knockout provides an MVVM pattern; this allows support of legacy browsers and handles complex dynamic interfaces [4]. As we can see, not all of these frameworks adhere to the Model View Controller architecture, although they may mimic the paradigm. One such example is Backbone; this framework seeks to omit the controller by integrating it with one or more stages of the Model or View.

C. *Further Discussion on Javascript Frameworks*

Specifically, the advantages of some MV* architectures which seek to utilize and perform the functions of the MVC in their own way. Angular's extensive backing and consistent improvement by Google's development team makes it the perfect choice for an all in one MV* framework. For projects that are expected to be executed on a variety of devices, Angular's ability to reuse view modules, and bind couple views to the controller - by use of the HTML tags or JavaScript code, makes it an exceptional framework that will facilitate development time and costs [5]. Although Backbone is a relatively simple MVC, lacking features that the other frameworks provide, its support of various libraries makes up for lost features. Due to this unparalleled flexibility, it is a suitable option for developing a project where requirements are unknown, obscure, or changed during the development process [4]. Ember prefers the convention over configuration paradigm [5], this is suitable in cases where the server-side processing is relatively simple and does not need to manipulate or operate using a wide variety of objects or procedures.

REFERENCES

.

[1] B. Foote, "POSA MVC," pp. 125-143.

[2] Heidke, Nick, Joline Morisson, and Mike Morisson. "Assessing the Effectiveness of the Model View Controller Architecture for Creating Web Applications," research paper.

[3] C. Chuck. "Programming Jakarta Struts." O'Reilly.

[4] K. Dean. "An introduction to MVC frameworks." December 6, 2013.

[5] B. Phong. "Angular, React or Ember: Which JavaScript framework is best for you?" October 11, 2016.

Web A2 Q1

ORIGINALITY REPORT

2%	2%	1%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	mics.sdsmt.edu Internet Source	1%
2	kar.kent.ac.uk Internet Source	1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		