

## International Journal of Computer Science and Mobile Computing



A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IMPACT FACTOR: 6.017

*IJCSMC, Vol. 8, Issue. 3, March 2019, pg.42 – 48*

# Multi-Platform Generic Dynamic Web Patterns

**Asst Prof. Dr. Sefer Kurnaz**

Electrical and Computer Engineering  
Altin Baş University  
Istanbul, Turkiye  
[sefer.kurnaz@altinbas.edu.tr](mailto:sefer.kurnaz@altinbas.edu.tr)

**Mustafa Abdulkhudhur Jasim**

Electrical and Computer Engineering  
Altin Baş University  
Istanbul, Turkiye  
[Toofy90@yahoo.com](mailto:Toofy90@yahoo.com)

*Abstract: It is complex and expensive to develop generic web-based software applications. Software solutions must understand each platform, technology and architecture's behavior and environment. The objective of the work is to produce a generic web pattern generator for three platforms of dynamic web developers: the Java EE, PHP Net. We have used the generic design architecture of the MVC (Model-View-Controller). Based on their specific technologies, the analysis of each layer within MVC can be analyzed. The lists of technologies being explored are as follows: Layers: Java EE: JSF and JSP .NET: Razor and Blade .NET; PHP: Twig and Blade; controllers layer: Java EE: Spring MVC .NET: .NET Framework. PHP: Laravel; Models Layers: Java EE: Hibernate and EclipseLink. We have developed a software application for student enrolment in courses to validate our proposal. The application has been implemented on the basis of all previously described platforms, technologies and architectures. A functional test to test the dynamic web functionality has been conducted. We have also conducted compliance tests to verify that the rules, specifications and structures have been followed by the specific platforms defined.*

*Keywords: Multi-Platform; Dynamic Web Patterns; .NET; Java; PHP; Model View Controller*

## Introduction

Dynamic web production is increasingly needed and different technologies have their own advantages at the same time. In addition to technology, in a dynamic web development software architecture becomes very important. The architecture is a basis for the definition and shape of dynamic web to achieve dynamic web quality. The Model View Controller (MVC) is one of the architectures widely used. Offering the opportunity to be reused, MVC strictly separates the application layer. The complexity and huge technical cost of stages, technologies and diverse web structural design. Developers work hard for every platform and technology's technology, architecture, ways, behavior, environment understanding. Programmers, on the other hand, need to produce dynamic web applications of high

quality over a shorter period. Various technical details are provided, which requires more time to understand, and would have an effect on the overall development of the software project.

If every platform and dynamic web technology is deleted, the details and complexity can be achieved in an invariant design that still guarantees its validity. Generic design describes the program architecture or general characteristics that defines structures of the implementing conduct of the general pattern.

The research goals to generate a broad sample for three web developers, namely Java EE, NET and PHP. A wide range of technologies can use each stage's resulting pattern to improve the productivity of programmer, while programmers need to transform technology into dynamic web development.

What other scientists did in this paper were examined. We then introduce the design, implementation and analysis of the software solution. A case study was conducted to verify the action and conformity of the software solution. The research has finally been completed.

## **Related Work**

Generic pattern can pull every platform and technology detail and complexity in an abstract way. Previous researchers have been conducting research on generic patterns in several attempts:

- Create a user interface from the database program code. A desktop-based user interface was created. A .NET (Visual Basic)- and Java (Swing, A WT) -based platform.
- The technology proposed to design a model that could at the time of implementation be raised to the user interface. This study was still necessary to show that the proposed model was verified on a user interface.
- Research for scenario generation of program code for web-based user interface. The logic of generating program code was presented in this study generally, but it showed no way to adapt engineering or design to a diverse platform or technology.
- Generation of Web - based logical user interface code of business. Based on the previously defined template, the user interface was created. The HTML tags generated the interface in this case.

In these studies, we found that the past researches from scientists specifically developed a basic pattern for one stage or technology and focused on suggested ideas.

## **Design and Analysis**

### **A) Problem Solution Analysis**

The approach from the bottom up can achieve a generic pattern. The generic pattern is achieved through a number of platforms to analysis the source code. Before a generic pattern and pattern was developed for each platform and technology, the software code for each platform and associated technologies was analyzed.

Two stages are the main structure of a dynamic web application:

1) The dynamic development stage and 2) the production stage of codes. For the development of dynamic web pattern, also, two main sub - stages for the analysis and design of dynamic web development patterns.

We assessed the MVC architecture model concept model and its specification for each layer during the analysis stage in order to have the dynamic pattern web application well developed. We examined, for each platform and technology, the basic design patterns and specified patterns during the design phase.

The pattern in generic patterns defined, which can interact with specified rules requiring a specific technique. Furthermore, define standard patterns for all technology should be applicable. By its purpose, the standard pattern is defined as standard generic patterns of communication with different technologies. The generic pattern for development is used for generating program code during the implementation phase.

**B) Dynamic Web Results Basic Pattern Generic Analysis**

The results of the analysis leading to a generic pattern of dynamic websites are different technology platforms:

1. The developed pattern should contain a high abstraction level in each layer of the MVC and can also be applied in the program code and interact with another layer in each layer.
2. The details of each technological application interact with a different layer of different techniques. The technology specimens cause very exact interactions and could ends to the lack of addition of MVC layers. Interaction between layers through interface or mapping is the solution for integrating the MVC layers. The communication between MVC layers through txt file mapping is designed in this study.
3. Basic pattern definition that applies to the technology of all developers. The basic design is defined as standard generic patterns of interaction with different technologies.

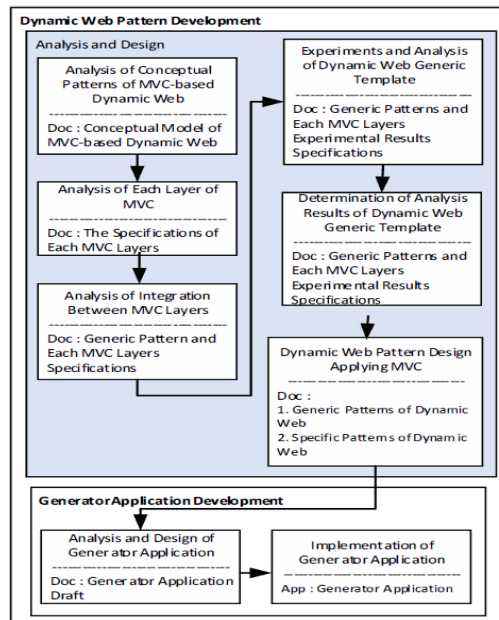


Figure 1. Analysis Phases

## Implementation

The applications analysis involve the structure of each MVC layer, libraries, platform files and the implementation of technology, and the characteristics and interaction of each project. Requirements Analyzer for the application. MVC architecture is an application with the ability to produce a web code dynamic on different platforms (Java EE, .NET, PHP). The web application generator is known as the web application generator. This program is called "Web Application Generator." Dynamic web functions like reading, creating, updating or removing can be provided by the WAG application. WAG can produce a database program code. A web program with the relevant technical podium and user selection is a WAG application output.

Figure 3 process flow description is as follows: 1) the database is ready, and then the data bases are loaded by WAG. 2) WAG can find all the data identities associated with the database from the database already loaded, like table names, tables' relationships, primary key for each table, attribute names, attribute types, and more. 3) Program code that is generated by WAG according to the platform technique and users required, once all the relevant identity databases have been obtained. WAG produces the code of the individual MVC layers and generate all settings (including database settings, application configuration, layers integration configuration, etc.), library loading, application structuring, etc. The code generation process is implemented by WAG. WAG uses package configuration files for application support, for example library and templates during the application code generating process, where the files are bundled when the applicable WAG is installed.

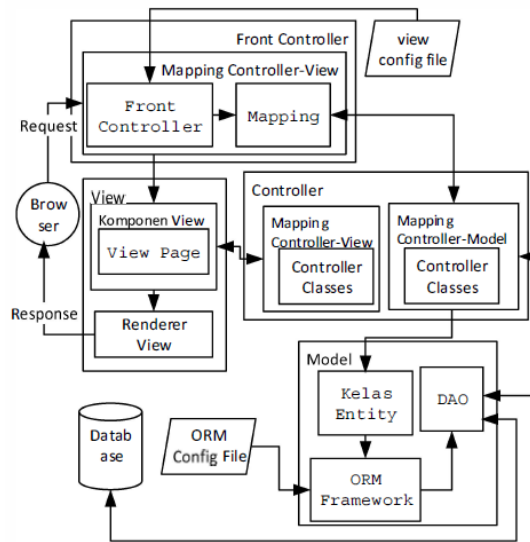


Figure 2. Dynamic web architecture generic pattern

### 1. Database Loading Algorithm

The flow of processes described in Figure 4 as follows: (1) connect to the database; (2) access the databank as connected to that database for the identity related to that particular database, such as attribute names, key of each table, attribute type of data, names, tables-to-person relationship; During program code generation, the process of mapping can be done by itself.

## 2. Code Generator Program Algorithms

The process of application code generating points to the data generated in the list algorithm of Figure 4. Dynamic web applications matching selected technology users and platforms are generated from List data. WAG generates the code in application generation on each MVC layer, creates all settings (e.g. database configuration, application settings and layer-to-layer settings), loads and structures application libraries. WAG depends on software files for supporting such as package and libraries which are templates when the program code production process takes place. The application generation output is a web application that is already functional and which creates, reads, updates and deletes the application.

3. The WAG UI is displayed in Figure 6. The WAG user UI.

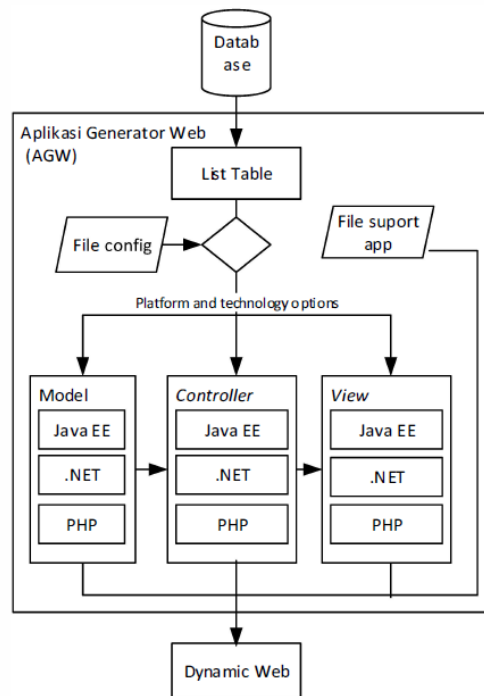


Figure 3. WAG architecture

## Application Testing

Using case studies, implementation of WAG was tested. The tests are aimed at validating the generic pattern that a dynamic web has developed. A functionality and conformance (conformance) test were used to validate generic patterns. Tests were conducted to confirm the dynamic web applications created by WAG's ability to properly create, read, update and delete with the features in the set. The application code that was generated for each platform was by functionality testing. The code was generated on each platform by experimental combinations of all techniques. The appropriateness tests in this study are to test the adequacy among generic patterns by producing dynamic web code.

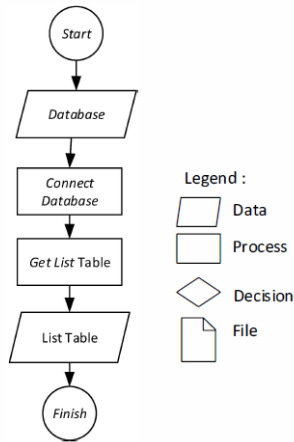


Figure 4. Flowchart of Algorithm of Database Load

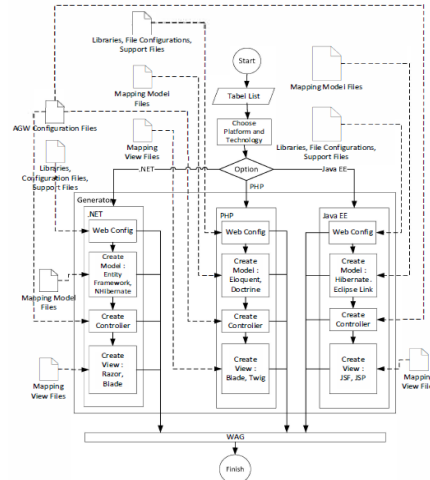


Figure 5 Program generator of algorithms or codes

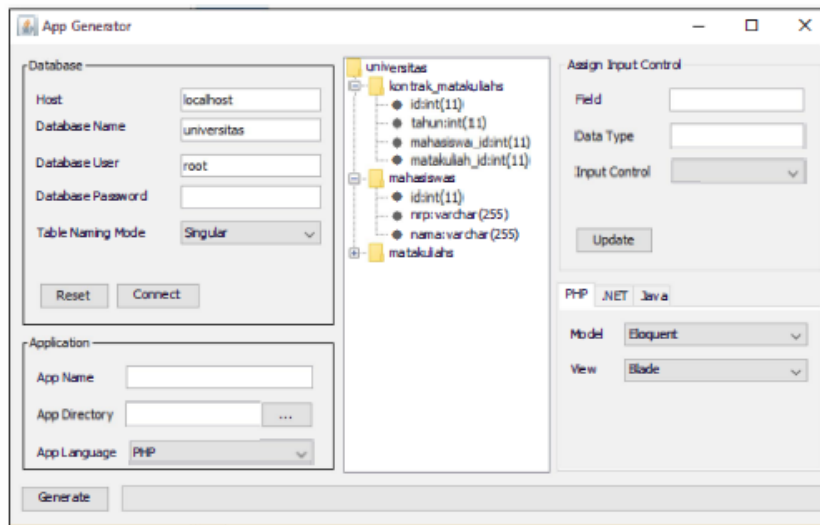


Figure 6 User interface o/WAG

## Conclusion

A dynamic web pattern with multiplatform has been proposed in this study, based on this proposal and a program generator. We hope that our research will improve productivity of the developers. Different platforms and certain technology were examined for each platform. We proposed a basic pattern for recognizing variances in these techniques integrated at different levels of MVC, in keeping with our initial study. Mapping is required to integrate every MVC layer. The map shows the interaction mechanism between the layers of MVC. We think the addition of new technologies can contribute to the programming of our dynamic generic web pattern.

# References

- [1] I. Sarker and K. Apu, "MVC Architecture Driven Design and Implementation of Java Framework for Developing Desktop Application," International Journal of Hybrid Information Technology, pp. Vol.7, No.5, 317-322, 2014.
- [2] I. Chhikara, "A Web Architectural Study of HTML5 with MVC Framework," International Journal of Advanced Research in Computer Science and Software engineering, pp. 451 - 454, 2013.
- [3] F. K. Frantz, "A Taxonomy of Model Abstraction Techniques," in Simulation Conference Proceedings, 1995. Winter, Arlington, 1995.
- [4] V. Tran, I. Vanderdonckt, M. Kolp and S. Faulkner, "Generating User Interface from Task, User and Domain Models," in Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, 2009. CENTRIC '09. Second International Conference on , Porto, 2009.
- [5] U. Zdun, "Dynamically Generating Web Application Fragments from Page Templates," in SAC '02 - Proceedings of the 2002 ACM Symposium on Applied Computing, Madrid, 2002.
- [6] A. Loh and M. Robey, "Generating Web Applications from Use Case Scenarios," in ASWEC '04 Proceedings of the 2004 Australian Software Engineering Conference, Australia, 2002.
- [7] H. M. Hao and A. Jaafar, "Tracing User Interface Design Pre-requirement to Generate Interface Design Specification," in Electrical Engineering and Informatics, 2009. ICEEI '09. International Conference on (Volume: 01), Selangor, 2009.
- [8] A. Ginige and S. Murugesan, "Web engineering: An Introduction," IEEE Multimedia (Volume: 8, Issue: 1), pp. 14 - 18, 2001.