

Building Web Mash-up using Agile Developer's Framework: Ruby on Rails

Anmol Kiran Uttarkar

Abstract.-A web mash-up is a method of creating a web application consisting of multiple applications in a single window. Web mash-up using agile developers framework we have created a web application that comprises of a blog, movie review and a to-do list. The agile frameworks support faster development of the applications and helps developers to develop software in a shorter period of time. Ruby on rails supports agile methodology which in turn helps to develop our project within a short period of time. Blog is for user comments on a particular topic, movie review is for giving a review on movies that has been released recently by user, and to-do list is for user daily works to be completed.

Keywords-Mash-up, Agile Design, Ruby on Rails.

##

1 Introduction

A web mash-up is a technique which can create a web application consists of multiple applications on a single window with multiple tabs in it. The proposed system with agile developer's framework has created a web application that comprises of a blog, movie review and a to-do list.

1.1 Ruby (The Programming Language)

Ruby is one of kind of object-oriented programming language which was developed by Yukihiro Matsumoto in mid 90's at Japan. The syntax and semantics of ruby is almost similar to Perl.

1.2 Rails (The Framework)

Rails framework was written in ruby which is a web framework so it's called Ruby on Rails. It makes the developer to create database-driven websites using scaffolding and code generation, taking convention over configuration.

1.3 Web Mash-up

Web mash-ups leverage the services and data already available on the Internet. With having rapid demand of decoupled applications, the API and Mash-up based architectures are solving real-life integration problems.

In Mash-up the authentication and access control functions are appropriately included and there won't be any compromise in the rapid development. We develop the applications to build services that can be reused with different context and can integrate them with larger applications seamlessly.

1.4 Agile Methodology

Agile design helps to simplify bigger tasks into smaller sub tasks called increments with the minimal planning and does not involve long-term planning. Each iterations takes one to four weeks at maximum by making use of team working in the full development cycle which includes: planning, requirements analysis, design, coding, unit testing, functional testing, iteration testing, and acceptance testing when completed product is shown to the client. This minimizes the overall risk which in turn helps the project to



All rights reserved editor@ijrct.org

adapt to the changes quickly. The objective of iteration is to have an accessible release with very much less bugs at the end of each of the iterations. To release a product or a feature requires multiple iterations.

2 Architecture

2.1 Agile Design Architecture [1]

The below figure explains the architecture of agile methodology.

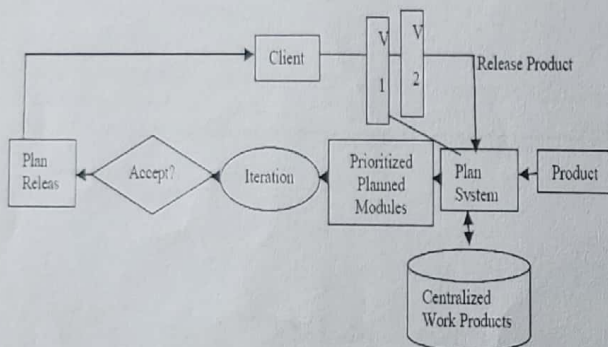


Fig 2.1: Agile Design Architecture

2.2 Rails Design Architecture

Rails focuses on a programming pattern called MVC, which means Model-View-Controller.

The MVC Architecture

Model-View-Controller architecture (MVC) is the important feature of Ruby on Rails as it depends on it. The main advantage of MVC is the detachment of Business logic from client-user interface. Other advantage of MVC is ease of keeping code - DRY (Don't Repeat Yourself) [3].

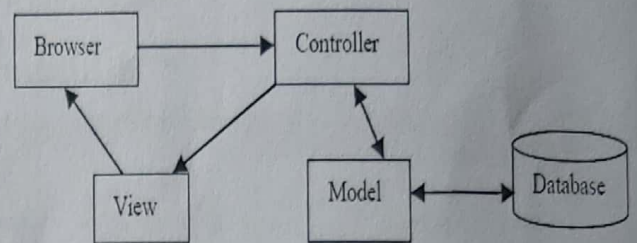


Fig 2.2: MVC Architecture

Model: Model which stores the data about all the tables of the database, if any ambiguity is there then it will do some primary validation of data before putting it into the database and one table relate to one model.

View: View is just an interface of the application. Views are HTML pages with some ruby code. Views submit the data to the web browser.

Controller: Controller is to identify the request which is coming from the web browser, and checks the models for information and passes that to the views for displaying that to the user.

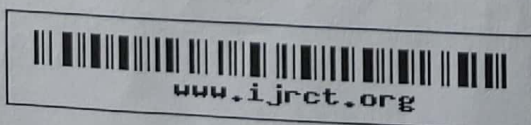
3 How Agile Development Helps Mash-Up Using Ruby On Rails [2]

This section helps to elaborate the Rails framework that makes it most appropriate for agile methodologies for creating mash-up.

MVC architecture - allows web professional to partition business logic, user presentation, and control features which in turn encourages the process of web development.

CRUD (Create, Read, Update, Delete) - The Rails routing engine having CRUD, or Create, Read, Update, and Delete, which all the web pages includes these four actions. For example, create a product, view it, or read a client-user, edit, or update a category, and delete an order.

DRY - "Don't Repeat Yourself", - The DRY principle requires every piece of system knowledge to have a unique, definite, and relevant representation which means the



developers can repeat themselves seldom or never. A piece of knowledge can be a data or a metadata, logic, functionality, or an algorithm. For example in an Active Record, which determine attributes and class values separately in the database and not in the program code.

Convention Over Configuration – This implies that Rails makes a presumptions about what it needs to do and how it's going to do it.

Scaffolding- In RoR the database scheme and the scaffold console command creates a basic skeleton of controllers and view templates that have create-retrieve-update-delete (CRUD) functionality. RoR also offers generator that saves a lot of time for recurring elements, such as a login form. The automatic option for creating of project scaffolding saves time and helps the developer to start the work immediately on the application's core functionality.

Active Record: The Active Record sub framework which establishes the connection between the domain objects and the database and also it transforms the CRUD functions that come from the Action Controller into SQL commands and sends the requests to the database, and returns the received results to the Action Controller. [4].

Action Controller: Action Controller which receives requests from active record and sends a view to the client.

Action View: Action View represents data. RoR defines templates to separate the presentation layer from the application. RoR has various template forms in that most important are RHTML and RXML. In RXML files, RoR uses its own builder library. The builder transfers Ruby commands into XML tags.

Action Web Services: Active Controller supports the lighter-weight REST (Representational State Transfer) approach. Through Action Web Services, RoR supports XML-RPC-based and SOAP-based Web services, thus the inclusion of external services and facilitating the publishing of functionality.

Web Mash-up: In web development, a Mash-up is a web-page or a web application that makes use of content from more than one source to create a single new service displayed in a single graphical interface.

4 Experimental Results

Step 1- Open the terminal and move to the folder where project is installed

```
$ cd MTech/demo/blog
```

Run the server- \$ Rails s (Fig 9.1)

Step 2- Open the browser and type an url- localhost: 3000

Blog/home page is opened with tabs blog, movie review, todo and about. (Fig 9.2)

Step 3- Click on sign-up page for registration and provide the details like username, password and confirm password. (Fig 9.3)

Step 4- After login click on the new post button to add new post, in new post provide title and body and click on create post. (Fig 9.5)

Step 5- Now add comment to the post just now added, with name and body or can be deleted if previously added comment. (Fig 9.7)

Step 6- Click on movie review tab to redirect to the movie review page. (Fig 9.9)

Step 7- Click on the new movie button to add new movie to the list by filling the details. (Fig 9.10 and 9.11)

Step 8- Click on write review button to give a rating to the movie created and provide the rating in stars and comment. (Fig 9.12 and 9.13)

Step 9- Click on todo tab to redirect to the todo page now clicks on new todo list hyperlink. (Fig 9.15)

Step 10- Now fill the form with title and description and click on create todo list button. Now we can see the new todo list page with new todo item creation option. (Fig 9.16 and 9.17)

Step 11- Fill the new todo and click on create todo item. Now item is added which can be seen in the same page. If user has completed the work then click on the tick mark or can delete the work by clicking on the trash button. (Fig 9.18)



Step 12- Click on the logout button to sign-off from the session. (Fig 9.20)

5 Conclusion And Future Enhancements

5.1 Conclusion

With the help of proposed system, it is useful:

- For designers to understand the design easily.

5.2 Future Enhancements

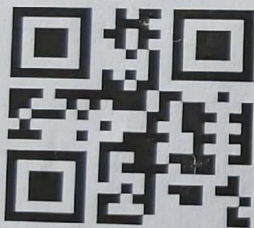
In this application we can add further more different applications so that user may find easy to access the application and which saves more time. More security

features can be added by using different gems provided in Ruby Gems website.

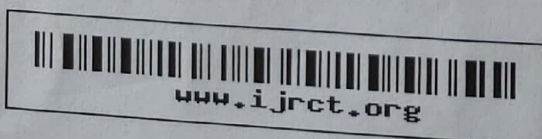
References

1. S Meenakshi, "Ruby on Rails - An Agile Developer's Framework", Vol 112 - No. 1, February
2. http://en.wikipedia.org/wiki/Agile_software_development.
3. Ruby S., Thomas D., Heinemeier Hansson D. - Agile Web Development with Rails, 4th Edition, Rails 3.2 (The Pragmatic Programmers) - 2011
4. https://en.wikipedia.org/wiki/Ruby_on_Rails

Author Name-Anmol U K,(Anmol Kiran Uttarkar) Email-
Id-anmolkiranuttarkar@gmail.com, Mobile-No-+91-
8904845758



www.ijrct.org



All rights reserved editor@ijrct.org