Ruby on rails: The new GEM of web development

Anagha Kelkar[1], Komal Mukadam [1]

[1] ***Department of Information Technology, B.E.* Information *Technology, V.E.S Institute of Technology, India***

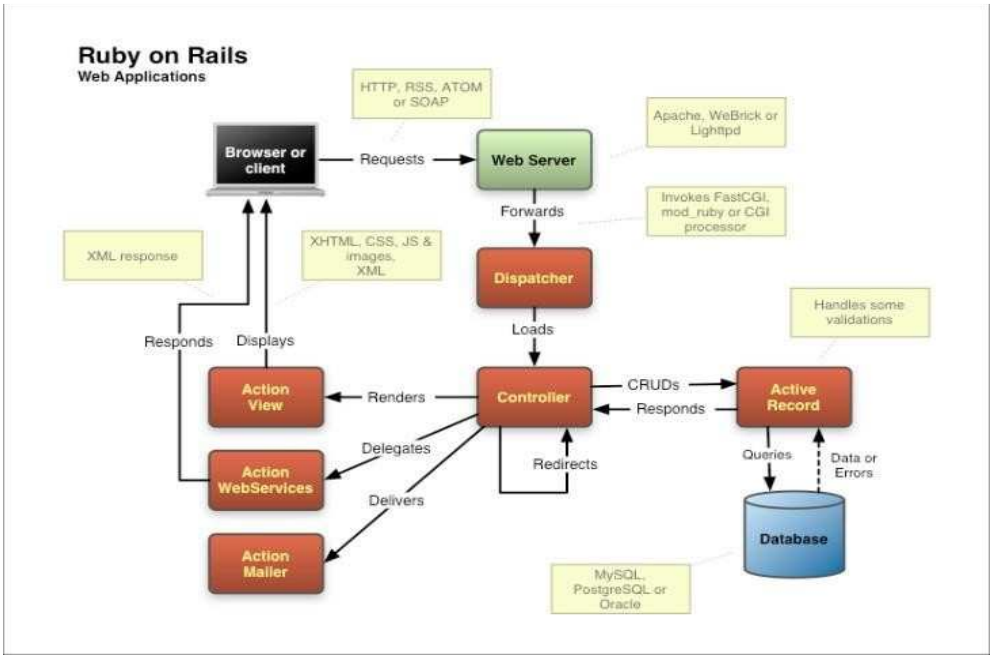
ABSTRACT:

Ruby on Rails is a web application framework written in a Ruby, a dynamically typed programming language. Rails is model-view-controller framework, providing default structures for a database, a web service and web pages. It encourages and facilitates the use of web standards such as JSON or XML for data transfer, and HTML, CSS and JavaScript for display and user interfacing. Ruby on Rails includes tools that make common development tasks easier "out-of-the-box" such as scaffolding. Ruby on Rails is also noteworthy for its extensive use of the JavaScript libraries, prototype, for scripting Ajax actions. Rails emphasizes the use of software engineering patterns including convention over configuration, don't repeat yourself and the active record pattern. Hence amazing productivity provided by this framework claims of Rails is the current buzz in the web development community.

Keywords: MVC Architecture, Multi-feature, Faster development, Convention over configuration, Active record, Ajax, Ruby gems, Ecommerce, Security

[1] INTRODUCTION:

[2]



Website development is not a new conception now-a-days and many organizations like educational institutes, business establishments, industrial houses, writers, entrepreneurs, technicians and others started developing the websites for a choice of kinds of purposes. Talking about website development, quick turn time along with clean high quality is two of the most important factors as they directly influence the client services. This is somewhere Ruby on rails framework is a milestone in achieving this success. Ruby on Rails is a web application framework written in Ruby, a dynamically typed programming language. It is an open-source web framework that's optimized for programmer's happiness and sustainable productivity. It lets you write beautiful code by favoring convention over configuration. The amazing productivity claims of Rails is the current buzz in the web development community. If your job is to create or manage web applications that capture and manipulate relational database from a web-based user interface, then Ruby on Rails may be the solution you've been looking for to make your web development life easier.

[3] ADVANTAGES OF RoR FRAMEWORK:

It's all about productivity

1.) DEVELOPMENT TAKES LESS TIME

It has been calculated that the development time on Ruby on Rails is reduced by 40-50% in comparison with the development on other programming languages.

2.) FLEXIBILITY

Modular design of Ruby on Rails apps results not only in faster development, but also in flexibility of these solutions. They can be easily modified, improved or extended after the release. "The Rails Way" you'll probably discover a tremendous increase in productivity.

3.) HERITABILITY

"The Ruby Way" is a concept of consistency in the structure and methodology when writing code followed by RoR developers to make code management between developers easier.

Write code not configuration 1.)

Don't Repeat Yourself

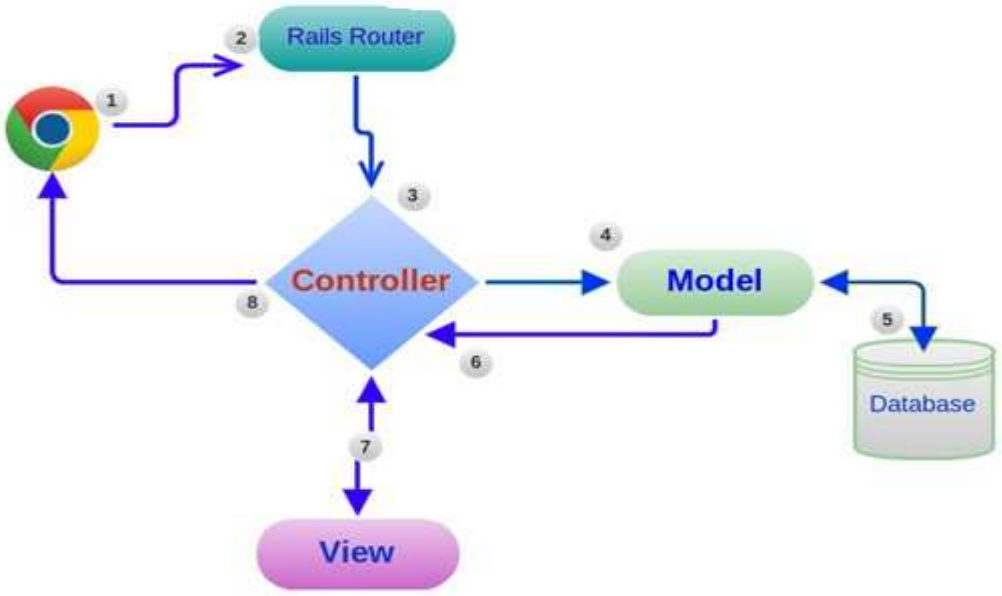
Ruby on rails framework follows a principle 'DRY' which states that "Every piece of knowledge must have a single, unambiguous, authoritative representation within a system." By not writing the same information over and over again, rails code becomes more maintainable, extensible and less buggy.

2.) Convention over configuration

Most web development frameworks force you to write pages of configuration code. If you follow naming conventions, rails doesn't need much configuration. Rails version of Active record discovers columns in a database schema and automatically attaches them to domain objects using metaprogramming.

[3] ARCHITECTURE:

Ruby on rails follows Model-View-Controller (MVC) framework. MVC pattern allows rapid change and evolution of the user interface and controller separate from the data model.



Model (ActiveRecord):

1.) Contains data for the application (often linked to a database)
2.) Contains state of the application (e.g. what orders a customer has)
3.) Contains all business logic
4.) Notifies the View of state changes
5.) No knowledge of user interfaces, so it can be reused

View (ActiveView):

1.) Generates the user interface which presents data to the user
2.) Passive, i.e. doesn't do any processing
3.) Views work is done once the data is displayed to the user.
4.) Many views can access the same model for different reasons

Controller (ActionController):

1.) Receive events from the outside world (usually through views)
2.) Interact with the model
3.) Displays the appropriate view to the user

[4] DIRECTORY STRUCTURE:

1.) ***app/:*** Contains actual code (controllers, models, views, helpers, mailers, assets) necessary to execute your project.
2.) ***bin/:*** Contains the rails script that starts app and for setup, deploy or run application.
3.) ***config/:*** Configure application routes and database.
4.) ***db/:*** Contains database structure.
5.) ***log/:*** Contains daily users' activities on web application.
6.) ***public/:*** The only folder seen by others.
7.) ***test/:*** Contains unit test-cases for web application.
8.) ***tmp/:*** Temporary files such as reports.
9.) ***Gemfile***: These files allow you to specify what gem dependencies are needed for your Rails application.

[5] RUBY GEMS:

During your development in Rails, there will be times where you will want to provide some functionality which is required by you, but either you don't know how to do or you don't want to implement it on your own since a lot of work has been put into its development by talented developers. RubyGems is a package manager for the Ruby programming language that provides a standard format for distributing Ruby programs and libraries (in a self-contained format called a "gem"), a tool designed to easily manage the installation of gems.
These developments which you might need (user authentication, message system, asset handlers, geolocation, pagination system, linking to exterior services such as Amazon AWS, and last but not least Rails itself) are called Ruby Gems. These are ruby software packages. A gem is a collection of Ruby code that we can extract into a "collection" which we can call later.

[6] FEATURES:

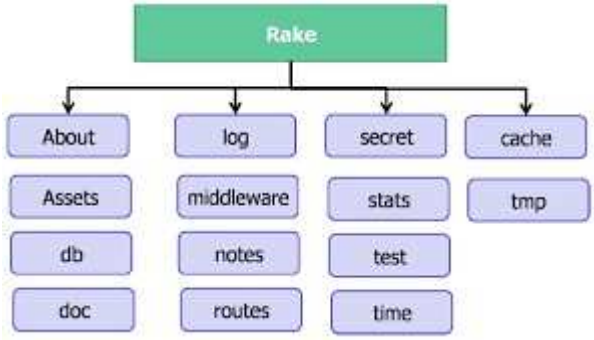
1. Associations:
What is importance of using an associations in rails? Because they make common operations simpler and easier in code. With active record associations, we can streamline two models by declaratively telling rails that there is connection between two models.
Types of associations:
1.) belongs_to
2.) has_one
3.) has_many
4.) has_many: through
5.) has_one: through
6.) has_and_belongs_to_many

2. Validations:
Validations are used to ensure that only valid data is saved in database. They are database agnostic, cannot be passed by end users, and are convenient to test and maintain.
Three ways to validate data in rails:
1.) Database constraints and/or stored procedures
2.) Client-side validations
3.) Controller-level validations

Methods create, save, update will trigger validations, and will save the object to database only if the object is valid.
Active Record offers many pre-defined validation helpers that you can directly inside your class definitions. They provide common validation rules. They include validations such as Acceptance, confirmation, exclusion, format, length, numericality, presence, absence, uniqueness etc.

3. Action mailer:
Action Mailer allows you to send emails from application using mailer classes and views.
Methods:
1.) Headers - Specifies any header on the email you want.
2.) Attachments - Allows you to add attachments to your email.
3.) Mail - Sends the actual email itself.

4. Rake tasks:



Rake is used for common administration tasks such as
1.) *About:* Information about Ruby version, RubyGems, Rails etc.
2.) *Assets:* Precompile the assets.
3.) *Database:* To migrate, create database and know current version of the database.
4.) *Doc:* Generate documentation for app, API, guides.

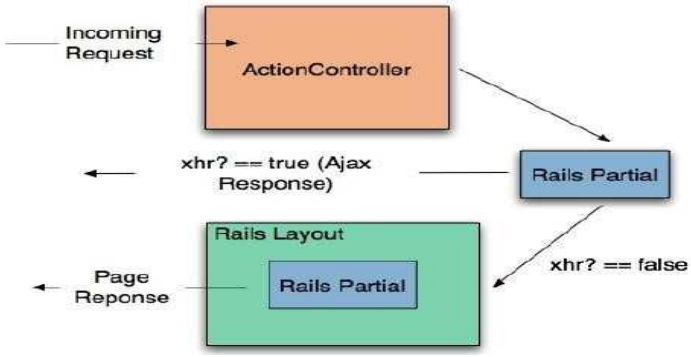
5. Working with Ajax:

1.) Without Ajax process:

When you type http://localhost:3000 into your browser's address bar and hit 'Go,' the browser (your 'client') makes a request to the server. It parses the response, then fetches all associated assets, like JavaScript files, stylesheets and images. It then assembles the page. If you click a link, it does the same process: fetch the page, fetch the assets, put it all together, and show you the results. This is called the 'request response cycle.'

2.) With Ajax:

JavaScript can also make requests to the server, and parse the response. It also has the ability to update information on the page. Combining these two powers, a JavaScript writer can make a web page that can update just parts of itself, without needing to get the full page data from the server. This is a powerful technique that we call Ajax.



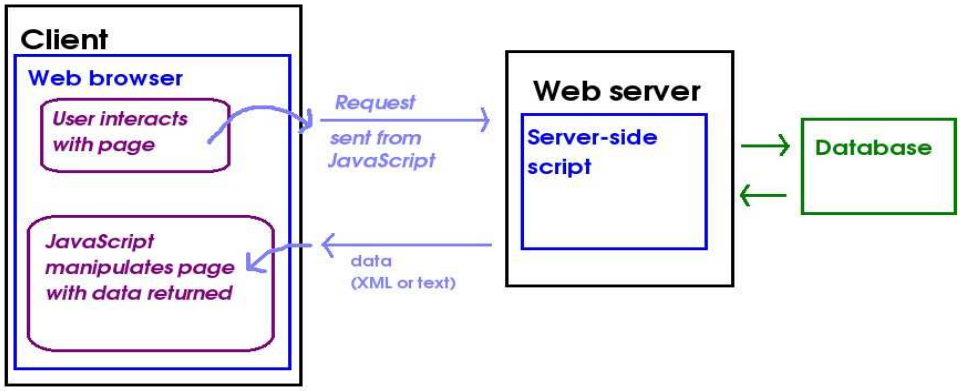
Rails has a simple, consistent model for how it implements Ajax operations. Once the browser has rendered and displayed the initial web page, different user actions cause it to display a new web page (like any traditional web application) or trigger an Ajax operation −

1.) *Some trigger fires* − This trigger could be the user clicking on a button or link, the user making changes to the data on a form or in a field, or just a periodic trigger (based on a timer).

2.) *The web client calls the server* − A JavaScript method, *XMLHttpRequest*, sends data associated with the trigger to an action handler on the server. The data might be the ID of a checkbox, the text in an entry field, or a whole form.

3.) *The server does processing* − the server-side action handler (Rails controller action) -- does something with the data and returns an HTML fragment to the web client.

4.) *The client receives the response* − The client-side JavaScript, which Rails creates automatically, receives the HTML fragment and uses it to update a specified part of the current page's HTML, often the content of a tag.



[7] WHEN TO USE RUBY ON RAILS:

In our experience, Rails is an ideal solution if your site falls into one or more of the following categories: *1.) E-COMMERCE*. Most e-commerce sites benefit tremendously from Rails' user-friendly features and modular approach to site development. We can also include features that you would normally only find in high-end e-commerce suites, such as bulk uploads and updates (for product descriptions and photos, extremely handy when you have thousands of products), custom pricing algorithms, and on-the-fly photo resizing/cropping (especially handy for making product thumbnails for browse pages). *2.) MEMBERSHIP SITES*. Membership and social networking options are pretty much "baked" into Rails. A variety of plugins are available to solve just about any social networking challenge you can think of.

3.) CONTENT MANAGEMENT. If the purpose of your site is to present thousands of articles, audio files, or other database-friendly content, Rails is a great solution because of the ease with which users will be able to navigate the site, and the ease with which you will be able to upload and manage the content.
4.) CUSTOM DATABASE SOLUTIONS. More and more of our new projects are custom solutions requiring a novel database structure to support a creative new business model. In most cases, Rails is an ideal way to realize build these solutions at a fraction of the usual time and expense. If site fits into one or more of those categories, generally Ruby on Rails is an ideal match.

[8] REAL TIME APPLICATIONS:

RUBY:

1. Simulations:
NASA Langley Research Center uses Ruby to conduct simulations. A research group in Motorola uses Ruby to script a simulator, both to generate scenarios and to post process the data.

2. 3D Modeling
Google SketchUp is a 3D modelling application that uses Ruby for its macro scripting API.
Business
Toronto Rehab uses a RubyWebDialogs based app to manage and track on-call and on-site support for the IT help desk and IT operations teams.

3. Robotics
At MORPHA project, Ruby was used to implement the reactive control part for the Siemens service robot.

4. Networking
Open Domain Server uses Ruby to allow people using Dynamic DNS clients to update in real time their IP configuration so that it can be mapped to static domains.

5. Telephony
Ruby is being used within Lucent on a 3G wireless telephony product.

6. System Administration
Ruby was used to write the central data collection portion of Level 3 Communications UNIX Capacity and Planning system that gathers performance statistics from over 1700 UNIX (Solaris and Linux) servers scattered around the globe.

7. Web Applications
Basecamp, a web-based project management application developed by37signals, is programmed entirely in Ruby.

8. Others
A List Apart, a magazine for people who make websites that has been around since 1997, has recently been revamped and uses a custom application built with Ruby on Rails.
Blue Sequence, a sophisticated mission-critical application which forms part of Toyota Motor Manufacturing's own "sequence-in-time" production process, has recently been selected as finalist the British Computer (BCS) Information Management Awards.

RAILS:

1. Redmine:
Redmine is a flexible project management web application. Written using the Ruby on Rails framework, it is cross-platform and cross-database. Redmine is open-source.
Some of the main features of Redmine:
1.) Multiple projects support
2.) Flexible role based access control
3.) Flexible issue tracking system
4.) Gantt chart and calendar
5.) News, documents & files management
6.) Feeds & email notifications
7.) Per project wiki
8.) Per project forums
9.) Time tracking

2. Diaspora:
Diaspora is a free personal web server that implements a distributed social networking service. Diaspora is intended to address privacy concerns related to centralized social networks by allowing users set up their own server (or "pod") to host content; pods can then interact to share status updates, photographs, and other social data. It allows its users to host their data with a traditional web host, a cloud-based host, an ISP, or a friend. The framework, which is being built on Ruby on Rails, is free software and can be experimented with by external developers.

3. Spree-commerce:
The Spree storefront offers a full feature set and is built on common standards, so you don't have to compromise speed to market, efficiency or innovation. Rails framework provides modular platform that allows you to easily configure, supplement or replace any functionality you need, so that you can build the exact storefront that you want.
Rails framework allows Spree-commerce to support following characteristics:
1.) Fully featured and modular
2.) Powerful customization
3.) Comprehensive API
4.) Own the solution
5.) Mature and trusted platform
6.) Robust open source community

[9] SECURITY:

Ruby on Rails takes web security very seriously. This means including features to protect application makers from common issues like CRSF, Script Injection, SQL Injection, and the like.

1. *Preventing Cross-Site Request Forgery:*
Modern versions of Rails protect against CSRF attacks by default by including a token named authenticity token within HTML responses. This token is also stored within the user's session cookie - when a request is received by Rails it checks one against the other. If they do not match, an error is raised.

2. *Preventing SQL injection:*

3.
Rails uses an Object Relational Mapping (ORM) framework called ActiveRecord to abstract interactions with a database. ActiveRecord, in most cases, protects against SQL Injection by default, however, there are ways in which it can be used insecurely which can lead to SQL Injection.

4. *Security Gems:*
Rails offers many built in security features to help protect applications, data and users from web based attacks by providing Gems.

1.) devise
Devise is a popular authentication and authorisation Gem for Rails. It offers secure password storage using bcrypt to hash salted passwords. User lockouts, user registration, forgot password functionality and more.

2.) brakeman
Brakeman is a Static Code Analysis tool for Rails applications. It searches your application's source code for potential vulnerabilities.

3.) secure_headers
Developed by Twitter, SecureHeaders is a Gem that implements security related HTTP headers into your application's HTTP responses. Headers such as Content Security Policy to help protect against Cross-Site Scripting (XSS) attacks, HTTP Strict Transport Security (HSTS) to ensure your site is only accessible over secure HTTPS, X-Frame-Options and others.

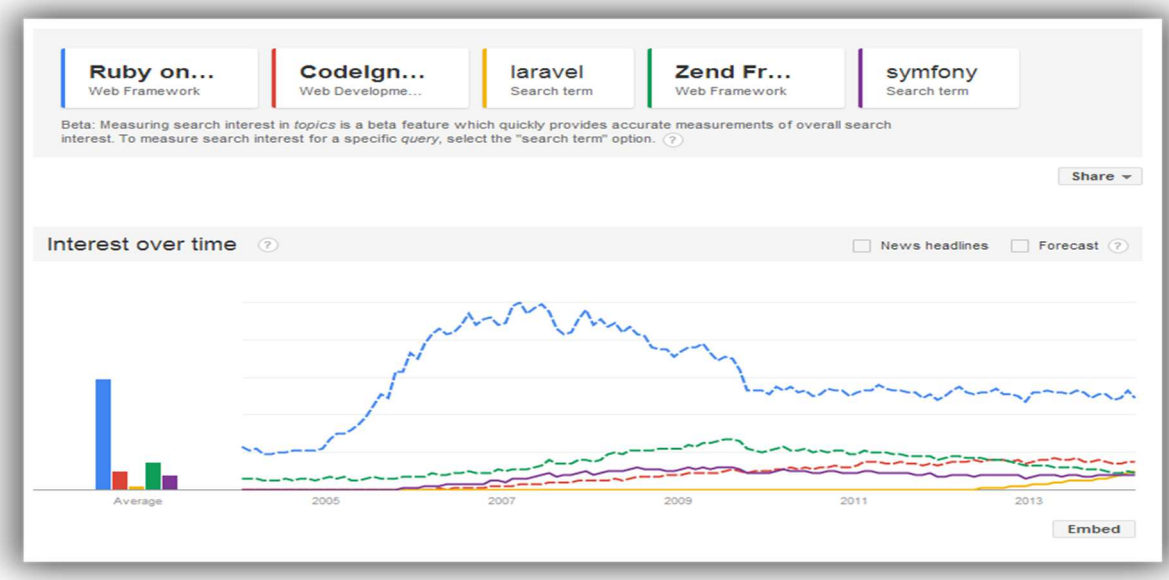
4.) rack-attack
Developed by Kickstarter, Rack::Attack is a Gem for blocking & throttling abusive requests.

Security real time applications:

1.) The Metasploit Framework, a community open source project managed by Rapid7, is a free penetration testing platform that helps IT professionals assess the security of their networks and applications. The Metasploit Project consists of over 700,000 lines of code and has been downloaded over a million times in 2010. The commercial editions developed by Rapid7 are also based on Ruby.
2.) The Arachni Web Application Security Scanner is a free, modular, high-performance Ruby framework aimed towards helping penetration testers and administrators evaluate the security of modern web applications.

[10] FUTURE SCOPE:

[11]



There are a number of advantages to using Ruby on rails platform instead of PHP or other. RoR combines Ruby along with CSS, JavaScript as well as HTML. This full-stack framework covers both back as well as front end design. Many popular sites such as BaseCamp, Twitter etc. have used this language as its base. Its popularity has made it important to start learning the language. Many popular sites are already using the framework and many are planning to use it effectively in future. As it is said "A different language is a different vision of life"…So definitely Ruby on rails framework will be a different era of programming in future.

[12] CONCLUSION:

Ruby on rails is definitely a milestone in achieving productivity, quick turn time, and high quality. No doubt it is rapidly gaining acceptance among Web programmers and designers. Also the rails framework and methodology has started and encouraged other similar projects based on other programming languages such as cakePHP which is a php-based approach implementing MVC functionality. However there is room for improvement but RoR is a welcome web standard that is and will be treated as new Gem of web development.

REFERENCES:

[1] Nicholas Belotti **Presentation on "Ruby & Rails"** [2] Manoj Kumar **"An introduction to Ruby on Rails"** Slideshare.net
[3] **"Ruby on rails guide"** http://guides.rubyonrails.org/
[4] **"Tutorials point"** http://www.tutorialspoint.com/ruby/
[5] **"Rails real time applications"** https://en.wikipedia.org/wiki/Diaspora_(software), http://www.redmine.org/, https://spreecommerce.com/storefront

Author[s] brief Introduction:

Anagha Kelkar
B.E. Information Technology, Department of Information Technology, V.E.S Institute of Technology, India

Komal Mukadam
B.E. Information Technology, Department of Information Technology, V.E.S Institute of Technology, India

Corresponding Address-

Anagha Kelkar
K/51, Palm Acres Society, Gavanpada, Mahatma Phule Road, Mulund East, Mumbai 400081.
Maharashtra. India
Contact No.: +91-9920427691

Komal Mukadam
AmiJharna, 12, Saras Baug, Sion Trombay Road,
Deonar, Mumbai 400088
Maharashtra. India
Contact No.: +91-9004535099