A survey on Network Application Frameworks

Syed Usman Hassan
Helsinki University of Technology
shassan@cc.hut.fi

Abstract

Application frameworks are a key resource in today's ever-growing and rapid application development. Different frameworks for various development fields are always coming up with rapid innovations. The scope of this article is to analyze few such frameworks, from both the developer's and the user's point of view, that not only help, but also support more efficient and modernized web and network application development that utilizes latest programming paradigms, features, libraries and tools. We have chosen four application frameworks for review in this article. Twisted, a python based network application framework for writing multi-protocol internet applications by exploitation of same tool base for all protocols. RealPeer, a Java based framework for development of p2p applications, allowing the developer to first simulate and then deploy the same application in a real environment. Merb, a Ruby based framework for developing web applications different from Rails, focuses more on plugin-oriented approach for optimized and light weight development, and Fuego, a mobile middleware framework for mobile network events and messaging. This article analyzes the features these frameworks offer, and describes the advantages and disadvantages of these features and contrast with similar technologies, and then concluding with some future trends and needed improvements in current framework development.

KEYWORDS: Framework, Application, API, Network, Web, Development, Middleware

1 Introduction

Application frameworks can be described as "programming aids" specified to develop a particular type of software, may it be a stand-alone application, a network application; on the web or in a closed network or a mobile device application. Application frameworks provide interfaces and tools to implement such systems more efficiently providing flexibility to add better interfacing, security, usability and communication without revealing much about the low level implementation details, such as protocol management and communication details in case of network applications.

Application frameworks are always on an innovative race with rapid releases from various programming languages and environments, bringing out products which encompass many development areas and various end-user functionalities and requirements, for example a framework that implements both web development and network development features.

This article presents some application frameworks chosen to obtain a wider view of how various development areas are enhanced or aided by frameworks and how their underlying development technologies are suited for the features they provide.

Four different frameworks are analyzed in this paper:

- Twisted: A network application framework for development of web and network applications using the python programming language.
- RealPeer: A simulation based framework focused on to develop p2p applications starting from a simulation development and progressing to real time implementation.
- Merb: A web-application framework developed for Ruby focusing on plugin based architecture.
- Fuego: A mobile middleware framework for mobile network events and messaging

2 Frameworks

2.1 Twisted

Twisted is based on Event Driven Architecture (EDA), enabling the implementation of dynamic networking capabilities with run time flexibility, i.e. the triggering of events based on other events (state changes). It began as a small game project and since then has developed into a complete development platform for internet and network applications and games. It was initially was designed to add multiplayer support to games. It is also worth mentioning that python based frameworks are used for internal scripting of some popular games today as well.

2.1.1 Technology and Implementation

Twisted is entirely written in python, which complements the above mentioned usage in game scripting (Civilization 4 uses python scripting for most in-game functions).However, major use or goal of Twisted is to be used as a platform for developing internet and networking applications. Twisted has support for multiple protocols, and we can use the same programmatic functions and tools for developing an application for any of the supported protocols, as the framework has inbuilt support for using the requested protocol, i.e. the developer does not need to change the code to deploy it for a different protocol.

Twisted consists of various modules coordinating different tasks of an application and provide various functionalities or systems to develop client and server-based applications. Twisted uses "factory" notation or subclass for both client and server applications to create or instantiate various protocols. The factory class is used to create and manage multiple connections for a single protocol, i.e. one factory per protocol.

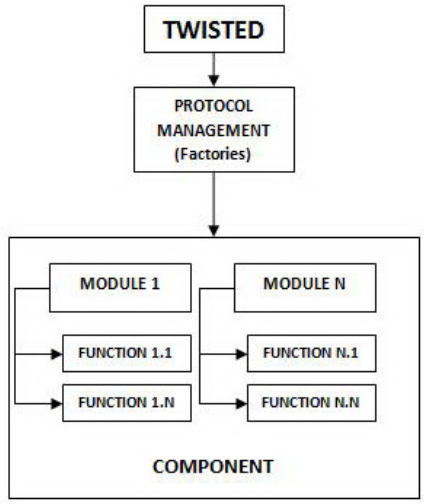

Figure 1: Twisted, a generic view

Twisted divides into modules or components based on role and functionality. Each of those modules then provides functions for the module's service area. For example, the GUI Integration module provides GUI features for various platforms. Starting from server development, Twisted has a number of server components such as, Twisted.mail, Twisted.words (chat server), Twisted.name (DNS server) and also Twisted.web (Web server). The last one uses the Nevow web application framework, which is also python based framework. It uses a component called Athena to implement Remote Procedural Calls for using python and JavaScript.

Another component is Twisted.Spread which manages remote object access, and includes modules for communication, serialization (marshalling). Moreover, there are modules for executables such as for server that create the previously mentioned servers and also testing executables which carry out unit testing functionality. There are also executables for document format conversion (lore) and for application deployment (twistd).

There is also a Misc component, which has a collection of utility modules such as Xish for XML and DOM manipulation, Cred for authentication and Enterprise for database management supporting various databases including Oracle and MySQL.

Furthermore there is Twisted.Internet which contains API's for managing event loops i.e. running events responding to other events or triggers and the inter and intra communication of these loops to deploy an event driven system on the internet. The Twisted.Protocols component manages all functionalities for all the protocols it supports, i.e. the developer can use the same functions and features for any protocol he wishes to utilize without having to alter the code if the same function is to be used for a different protocol.

2.1.2 Advantages

The advantages of Twisted can be seen first in its modular division for functionalities, but here we mean to say that twisted has "intelligently" modularized the functions. A second advantage would be the property to make all functionality available in the same way for all protocols. It is not implemented strictly for every protocol it supports, but it does for the majority and with time it is constantly being improved to achieve this goal. Another positive aspect is the extensible feature provided by the Twisted.plugin, which allows to extend the application by adding third party API or code enhancement to the application. The plugin feature is flexible in resource management i.e. it allows for these plugins to be loaded according to application requirements, for example a plugin can be loaded and saved on first program run, or re-discovered each time the program starts.

2.1.3 Disadvantages

There are also a couple of disadvantages in the framework, in our opinion one of them is that the framework can only be optimally used when there is no memory consuming tasks implemented. It doesn't mean that we can't use memory properly and efficiently with python but some implementation can have higher memory overheads. For example with reference to twisted, if we make a web application which uses some robot function (an automated script or function) for some task such as testing or checking bottlenecks, then memory intensive actions would be required. Another drawback that we noticed is the limitation of the framework to develop Rich Internet Applications. Such applications would require flexible graphical manipulation and interfaces for which there is no component present in the framework. The use of Athena in calling JavaScript code is not enough to make a Web 2.0 application. Support and implementation utilities for Web widgets and AJAX whether as a new component or new modules in Nevow would be a great addition and feature for Twisted.

2.2 RealPeer

The next framework included in the study is RealPeer. It is basically a framework that can simulate p2p networking and to develop a more realistic implementation . That is, the approach for this framework is to first simulate the p2p application, test that simulation under various definable parameters, and then use it either as a standalone real time application or an extension of an existing p2p engine. It is a very recent framework released in February 2008.

2.2.1 Technology and Implementation

RealPeer is written in JAVA, and implements various features required to design simulation and making a real time version of the p2p application, either as a plugin to an existing p2p application or a new standalone p2p application. The methodology followed is to start to design a model of a

p2p application. After modeling, the developer implements the features and functionality contained in that model. After modeling and designing, the developer uses the simulation engine to simulate the implementation. Not only can the developer simulate the main functions but can also define and simulate the network environment with virtual peers (as many as required). Then after the testing has been completed the application can be incorporated in real time systems if it is intended as a plugin or can be run as stand-alone application, then it can be tested practically.

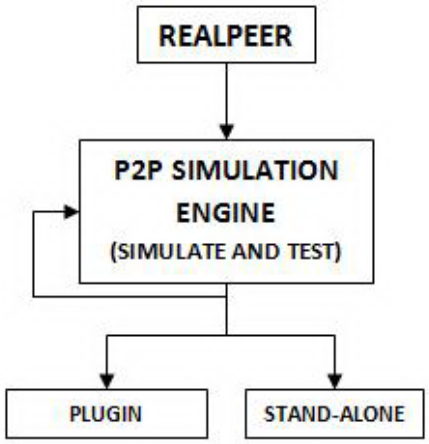

Figure 2: RealPeer

The framework, as we pointed out with twisted as well, offers modular implementation for the ease of programmer, with each functionality separated. The simulation system is designed so that it can offer some control on the simulation that runs and implements validity checks, such as making sure the input parameters are not erroneous and impractical as compared to a real environment. For example, a programmer cannot define a network bandwidth of 10 Tbps (Tera bits per second). These validity checks ensure that simulation results are near to practical feasibility.

The model that the framework offers allows various features such as the ability to simulate a "physical network". It provides interface for various protocols, it is similar to the Java socket implementations. The developer can add quite a lot of detail depending upon the interfaces he wants to use and define, for example, he can define the protocols to use, the type of network that maybe present, the type and speed of network connections. Then there is also the feature to simulate raw data transfer between peers, i.e. large amounts of data which will closely simulate a real p2p session. Such as Gigabytes of data being transferred using full network resources and not just some small and simple text message transfers between peers, which will not use any significant network resources and will be hardly a couple of Kilobytes.

2.2.2 Advantages

One advantage is the "real" simulation. i.e. not just simple simulation to test whether we can share data or transfer it but actually to be able to generate an actual environment with real parameters. These parameters include validity checks as explained in previous section, access interfaces, and to be able to simulate proper data transfer and a large number of peers.[3]According to developer publication the test run was simulated with 20,000 peers. Another advantage is "Analysis" component, the role of which is to give the programmer analysis or decision aids and information on how the system is progressing and what variables and parameters are in which states.

2.2.3 Disadvantages

One of the drawbacks of the framework is that it is not a very abstract one i.e. it has a very limited scope to cater to implementation of only p2p systems, that is not a flaw but it seems limited when more could be achieved from the same, such as the extension on analysis component to compare with some standard p2p applications so that the simulation is not only tested in its own functionality but at the same time can be compared with other existing applications.

Also, importantly the framework should be able to support a GUI implementation so that programmer does not need to create separate GUI and link it with his functions. A GUI engine would be a very helpful tool in this framework. Also the framework currently has no support for parallel or distributing computing.

2.3 Merb

Merb is an MVC (Model View Controller) oriented framework i.e. the implementation and user interface are separated from each other and each can be independently altered without effecting the other. It is a developed using Ruby and is thought of as counterpart for Rails (also done in Ruby) and is faster and lighter than Rails. Merb is also a very recent framework released in November 2008.

2.3.1 Technology and Implementation

As already mentioned Merb is written in Ruby and depends a lot on JavaScript and its associated features for web 2.0 development such as AJAX or jQuery. A few general features of Merb are that it is plugin based as compared to Rails which was designed to load everything with the core load, i.e. Merb uses extended features whenever needed and does not load them when the application runs. To use many such functionalities, Merb uses plugins and frees the application from the hassle of loading all functions and code with each run regardless of the fact that it will be used or not. This also means that the core is quite limited, not feature wise but size wise, so that it is designed to interact with the Merb plugins and itself provides just the engine and interface but not any functionality.

The plugins in Merb are simple gems (gem is a Ruby package) and are distributed either manually, by a third party or incorporated as part of the Merb application. The plugins automatically get installed when we run the web application or set it up on a server. As with Twisted, the Merb plugins can be configured to be loaded and saved either one time on first run or to be fetched each time the application is loaded.

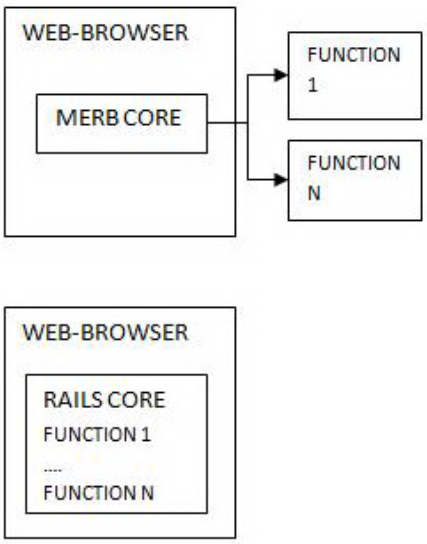

Figure 3: Merb and RAILS

Controllers, which are essentially the same components as mentioned earlier for the MVC model i.e. they control the UI and the implementation. Two controllers are used, one for semantics and layout and the other one for request/response management. The prominent feature in the controllers is the complete support of various and more importantly popularly used content types (such as both XML and HTML).

Merb also has other features such as Mailer component for mails, again with the option to separate implementation from interface. Another one is the Parts component which is similar to Mailer in the separation of implementation, but used for various objects on a web application, such as a clock, to be simple, we can create a digital clock as a Part and can use that Part anywhere and any number of times on our site. It means that the "Part" is actually a function which can be called anywhere any number of times, but this component is more of an object rather than a function, i.e. it is an object that performs certain defined functions without having to call those functions.

The exception handling in Merb works in a way that it does not throw and catch exceptions simply but throws few standard exceptions in the UI, which the programmer can customize. For example, all form submission errors will be output to a predefined interface and the developer can customize the output of that error. Also there is the testing feature with the ability to mock objects and test the implemented features and UI, it will create dummy or random objects and will allow to simulate some functionality which the developer wants to test. Merb also contains generators which build up skeletons when we start building applications so we already know where to implement what. It will build up a file structure with appropriate folders for content e.g. layouts for design views or parts for object manipulation.

2.3.2 Advantages

An advantage of Merb is that it has light-weight core, which allows the web application to work efficiently without the need to load all the functions code, objects code every time, and just use the required function or object when called or needed from the plugins. An advantage of being an MVC framework is better usability for developers as it supports rapid application development. This can be scene in the following scenario; When a programmer works on one feature and implements both its function and layout, he will always have to make them sync with each other. But when they can independently be changed without affecting the other, the developer does not need to worry about testing both the functionality and design for one change that he makes.

2.3.3 Disadvantages

Merb is immature and by no means a replacement for Rails, so still it needs to mature in order to provide same developer standard as that of Rails. For example, it should have a module specifically for jQuery or AJAX, meaning that it is not necessary to implement jQuery by knowing the jQuery syntax and technical aspects, we can user Merb to hide that implementation and give the programmer the functionality.

Another drawback could be the plugins being incompatible with any browsers. For instance in Internet Explorer many JavaScript features do not work, the most prominent being the use of SVG graphics which IE does not support well.

2.4 Fuego

Fuego middleware facilitates mobile middleware to make mobile application development and deployment. Middleware is something that could provide either a platform for development or than an interface between the software and operating system. It was developed during 2002-2004.

2.4.1 Technology and Implementation

Fuego middleware is written in JAVA. It provides a number of implemented features. It includes a scalable distributed event framework, an XML-based messaging service, a mobile XML-aware file synchronizer, a mobile presence service and a gateway between SIP events and the Fuego event framework.

The event system is based upon subscribers who get notification according to definable filters. Publishers, who publish the notifications, and the fuego routers, which route the notifications to the client. The notifications can be in any supported format file (e.g. XML) and there are also HTML views available for browsing sessions and views for router activity. The middleware also has a visualization component to visually monitor the activity, e.g. of the router.

The XML based messaging system is split into three parts, namely the service API which provides the application interface, a protocol for sending and receiving messages (called AMME and is connection-oriented) and a Xebu system to serialize and parse SOAP messages.

The Synchronizing XML-aware file system (Syxaw)[1] is a hierarchical file system for mobile devices that provides easy and efficient device-to-device file sharing, simple sharing semantics, and built-in support for reconciliation of concurrent updates to the file system.

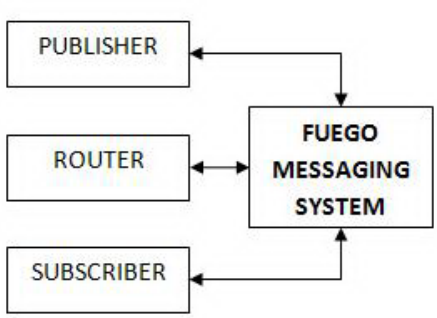

Figure 4: Fuego messaging

Furthermore, there is the presence and instant messaging service which provides features for presence and context information and instant messaging application interface. Users can publish their presence information from their devices and the peers can obtain this information as a notification. The publisher does not need to be aware how many peers are getting notified as long as he publishes his info on the presence channel.

2.4.2 Advantages

Fuego is advantageous because it provides a number of features in one middleware implementation, the ability to have a network of mobile devices with routers using the middleware features is good way to implement routing and notifications. This means that using its features various customized networks and messaging systems can be developed according to need, as it does not place any restriction on what type of network to form or what kind of messages to use. With the added provision for monitoring all the activity of end-devices as well as intermediate devices(routers), it also covers security aspects.

2.4.3 Disadvantages

Fuego is only applicable to closed or local networks, and perhaps an added API for location based services can have a very positive impact on the framework's utility. For example, dynamically joining a network and get notifications based on the user's location, upon moving near a shopping center, user gets notified about any sales or other shopping information.

3 Comparisons and Contrasts

Looking at them generally, Merb, twisted and fuego are implemented in a way to encompass or provide a more diverse range of features enabling developers to develop various and different applications, whereas RealPeer has been designed to cater to specific target (i.e. p2p).

Twisted and Merb are not that different from each other in the way that both support building of web applications. However it should be noted that twisted is more technically versatile than Merb in providing much more features and many lower level implementations than Merb. On the other hand, although young, Merb offers features to implement Web 2.0 applications easily whereas Twisted has no such support, such as being able to make graphical, interactive objects is a feature of the Merb framework.

Merb is not as feature enriched and widely used as Rails. Usage is not a differing factor as it is not even a year old, but Merb will need more features. Another thing to compare with Rails is that Merb does not load everything with the core (application), which is advantageous and efficient, but browser support may not be as much as that of Rails.

Fuego is a positive effort to encounter fragmentation in mobile application development as it uses Java to provide a set of various API's for application development and does not depend on the underlying operating system of the device.

4 Future and Evolution

What we can expect from newer frameworks after this study is to obtain more in a single framework than currently being offered. The evolution should be towards designing frameworks that provide maximum features under one programming language, i.e. we should not always have one feature better with python framework and another feature more efficient in a Java or Ruby framework.

Moreover, it is seen that frameworks are moving towards hiding the lower level details from the programmer. This can be considered both a positive and negative aspect as it depends upon the type of service needed. A developer probably needs lower level accessibility and manipulation for a framework like Twisted, but he may not need so for Merb, because Merb only offers web application development whereas twisted caters to general network applications as well, which may not be interface oriented at all and may just offer a network service.

Looking at the p2p application development, there is much work being carried out for mobile p2p applications, frameworks such as Peer2Me and PeerNet provide this functionality for mobile devices, and the next step for frameworks like Realpeer would be to provide the same simulation tool for mobile devices, as p2p applications for these devices are still a question mark to the developer community as to whether they are essential or not. So having an analyses tool is always useful when development is at an early stage.

5 Conclusion

Summarizing the discussion we can conclude that all four frameworks offer attractive features for programmers expert in different languages and looking for different application areas. Twisted is more suited for developers working on network systems with events and triggers and deploying multi-protocol applications. RealPeer is more suited for developers and especially research developers looking for p2p optimization and better resource management. Merb is suited for developing Rich Internet Applications (RIA's). It is growing to support more complex web systems, at the same time releiving the browser from excessive overhead. Fuego gives useful mobile device network messaging and analyzing portal for the developers.

References

[1] Fuego Core. *Documentation of Fuego Middleware and FCK*, 1.0 edition.

[2] E. Greenberg. *Network Application Frameworks - Design and Architecture*. Addison-Wesley, 1999.

[3] Hildebrandt, Dieter, Ludger Bischofs und Wilhelm Hasselbring. RealPeer - A Framework for Simulation-based Development of Peer-to-Peer Systems. In *15th Euromicro Conference on Parallel, Distributed and Network-based Processing (PDP 2007)*, February 2007.

[4] Sasu Tarkoma, Jaakko Kangasharju, Tancred Lindholm and Kimmo Raatikainen. Fuego: Experiences with Mobile Data Communication and Synchronization. In *17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, September 2006.

[5] Twistedmatrix Labs. *Twisted Matrix User Manual*, 8.2 edition, 2008.