Advances in Computer Science and Information Technology (ACSIT)
p-ISSN: 2393-9907; e-ISSN: 2393-9915; Volume 4, Issue 5; October-December, 2017, pp. 327-330

http://www.krishisanskriti.org/Publication.html

Web Enabled Application Progression

Dr. Harpreet Kaur Sethi

Dept of Computer Science S.R. Govt. College (W), ASR
E-mail: harpreetsethi.27@gmail.com

Abstract—*A Web application is designed from the start to run in a Web-based environment. This means that the hypermedia aspect in terms of hypertext and multimedia in combination with traditional application logic must be taken into account throughout the application lifecycle, which makes it different with respect to a conventional application. The complexity of Web sites are increasing and transforming into Web applications that contain business logic, interactivity, transaction handling and states. This phenomenon forces the Web developers to adapt more traditional software engineering techniques to keep the Web applications error free, maintainable, reusable, well documented etc. Many Web developers do not use any engineering techniques. Web application is an application that was designed from the beginning to be executed in a Web-based environment.*

Keywords*: Hypertext, Hypermedia, Transaction Handling, Web Developers, Web-Based, Conventional.*

1. INTRODUCTION

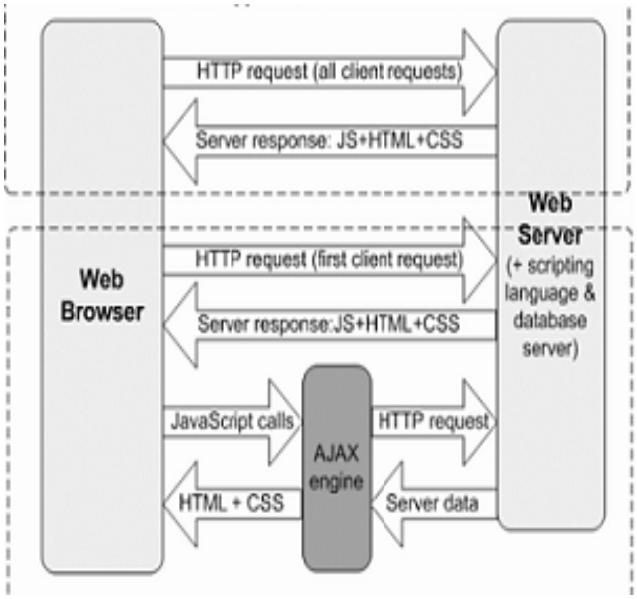
Web enabled application progression is the process and practice of developing web applications. Just as with a traditional desktop application, web applications have varying levels of risk. A personal home page is much less risky than, for example, a stock trading web site. For some projects security, software bugs, etc. are major issues.

If time to market, or technical complexity is a concern, documentation, test planning, change control, requirements analysis, architectural description and formal design and construction practices can mitigate risk. In the early 1990's analysis was a small part of the traditional (non-Web) system progression process .The traditional Web page has been developed to be as fashionable as possible and this approach has caused problems because of the lack of functionality analyzing. Another aspect is that reuse and site maintenance are not considered, leading to difficulties in making modifications but also takes away the possibility to reuse parts of the application, saving time and money in the development process.

There are goals that developers should try to fulfill with a page. It should be correct and error free, maintainable, reusable, robust and reliable, well documented etc. To achieve this you should use software engineering techniques because Web sites are becoming more and more like traditional software.

2. TERMINOLOGY OF WEB APPLICATION

Some common terminology in today's modern web world



Web Application Interface Architecture

2.1. Browsers

Browsers are the interpreters of the web. They request information and then when they receive it, they show us on the page in a format we can see and understand.

- Google Chrome - Currently, the most popular browser brought to you by Google
- Safari - Apple's web browser
- Firefox - Open-source browser supported by the Mozilla Foundation
- Internet Explorer - Microsoft's browser.

2.2 HTML

HTML is a markup language. It provides the structure of a website so that web browsers know what to show.

2.3 CSS

CSS is a Cascading Style Sheet. CSS let's web designers change colors, fonts, animations, and transitions on the web. They make the web look good.

- LESS - a CSS pre-compiler to make working with CSS easier and add functionality
- SASS - a CSS pre-compiler to make working with CSS easier and add functionality.

2.4 AJAX

Ajax is not a single technology, but rather a group of technologies. HTML and CSS can be used in combination to mark up and style information. The webpage can then be modified by JavaScript to dynamically display – and allow the user to interact with — the new information. The built-in XMLHttpRequest object within JavaScript is commonly used to execute Ajax on WebPages allowing for websites loading content onto the screen without refreshing the page. Ajax is also not a new technology, or another different language, just existing technologies used in new ways.

2.5 Client (or Client-side)

A client is one user of an application. It's you and me when we visit http://google.com. Client's can be desktop computers, tablets, or mobile devices. There are typically multiple clients interacting with the same application stored on a server.

2.6 Server (or Server-side)

A server is where the application code is typically stored. Requests are made to the server from clients, and the server will gather the appropriate information and respond to those requests.

2.7 Front-end

The front-end is comprised of HTML, CSS, and JavaScript. This is how and where the website is shown to users.

2.8 Back-end

The back-end is comprised of your server and database. It's the place where functions, methods, and data manipulation happens that you don't want the clients to see.

3. WEB BASED ENABLED LANGUAGES INTERFACE

3.1 Programming Languages Interface

Programming language interface are ways to communicate to computers and tell them what to do. There are many different programming languages just like there are many different lingual languages (English, Spanish, French, Chinese, etc). One is not better than the other. Developers typically are just proficient at a couple so they promote those more than others. Below are just some of the languages and links to their homepages

- JavaScript - used by all web browsers, Meteor, and lots of other frameworks
- Coffeescript - is a kind of "dialect" of JavaScript. It is viewed as simpler and easier on your eyes as a developer but it complies (converts) back into JavaScript
- Python -used by the Django framework and used in a lot of mathematical calculations
- Ruby - used by the Ruby on Rails framework
- PHP - used by Wordpress
- Go - newer language, built for speed.
- Objective-C - the programming language behind iOS (your iPhone), lead by Apple
- Swift - Apple's newest programming language
- Java - Used by Android (Google) and a lot of desktop applications.

3.2 Application Programming Interface

An **API** is an application programming interface. It is created by the developer of an application to allow other developers to use some of the application's functionality without sharing code. Developers expose "end points" which are like inputs and outputs of the application. Using an API can control access with API keys. Examples of good API's are those created by Facebook, Twitter, and Google for their web services

4. FRAMEWORKS

Frameworks are built to make building and working with programming languages easier. Frameworks typically take all the difficult, repetitive tasks in setting up a new web application and either does them for you or make them very easy for you to do.

- Meteor - a full-stack (front and back end) javascript framework
- Node.js - a server-side JavaScript framework
- Ruby on Rails - a full-stack framework built using ruby
- Django - a full-stack framework built using python
- Ionic - a mobile framework
- Phonegap / Cordova - a mobile framework that exposes native api's of iOS and Android for use when writing javascript
- Bootstrap - a UI (user interface) framework for building with HTML/CSS/Javascript

- Foundation - a UI framework for building with HTML/CSS/Javascript
- Wordpress - a CMS (content management system) built on PHP. Currently, about 20% of all websites run on this framework
- Drupal - a CMS framework built using PHP.
- .NET - a full-stack framework built by Microsoft
- Angular.js - a front-end javascript framework.
- Ember.js - a front-end javascript framework.
- Backbone.js - a front-end javascript framework.

5. LIBRARIES

Libraries are groupings of code snippets to enable a large amount of functionality without having to write it all by yourself. Libraries typically also go through the trouble to make sure the code is efficient and works well across browsers and devices (not always the case, but typically they do).

- jQuery
- Underscore

6. DATABASES

Databases are where all your data is stored. It's like a bunch of filing cabinets with folders filled with files. Databases come mainly in two types: SQL and NoSQL. SQL provides more structure which helps with making sure all the data is correct and validated. NoSQL provides a lot of flexibility for building and maintaining applications.

- MongoDB - is an open-sourced NoSQL database and is currently the only database supported by Meteor.
- Redis - is the most popular key-value store. It is lighting fast for retrieving data but doesn't allow for much depth in the data storage.
- PostgreSQL - is a popular open-sourced SQL database.
- MySQL - is another popular open-sourced SQL database. MySQL is used in Wordpress websites.
- Oracle - is an enterprise SQL database.
- SQL Server - is an SQL server manager created by Microsoft.

7. PROTOCOLS

Protocols are standardized instructions for how to pass information back and forth between computers and devices.

- HTTP - This protocol is how each website gets to your browser. Whenever you type a website like "http://google.com" this protocol requests the website from google's server and then receives a response with the HTML, CSS, and javascript of the website.
- DDP - is a new protocol created in connection with Meteor. The DDP protocol uses websockets to create a consistent connection between the client and the server. This constant connection lets websites and data on those websites update in real-time without refreshing your browser.
- REST - is a protocol mainly used for API's. It has standard methods like GET, POST, and PUT that let information be exchanged between applications.

8. DATA FORMATS

Data formats are the structure of how data is stored.

- JSON - is quickly becoming the most popular data format
- XML - was the main data format early in the web days and predominantly used by Microsoft systems
- CSV - is data formatted by commas. Excel data is typically formatted this way

9. CONCLUSIONS

Some techniques new protocols structure and found something in here that gave you a new way to think about or talk about web technologies. This was not meant to be an all-encompassing list, but rather a way to talk about all the great technologies we have at our finger tips. In this paper I have presented different techniques ,terminology developing web applications in a competitive and rush-to-market environment. Those findings suggest that development insuch a environment is communication intensive and work related these practises tothe success criteria identified by the websites for layman and professionals and pointed out some future work directions.In the future I will to look more at the decision making processes applied in web application progression more enhanced and security techniques with users.

REFERENCES

[1] Greene, Patrick. Available online 2002-03-06 at http://www.xgenapplications.com/pro-con.htm published in January 2001

[2] Jarke, Matthias Ph.D. Available online 2002-05-12 at http://portal.acm.org/citation.cfm?doid=290133.290145

[3] Published December 1998 in Communications of the ACM (Volume 41, Issue 12) Kafura, Dennis Ph.D. Available online 2002-04-26 http://people.cs.vt.edu/~kafura/cs2704/oop.swe.html Published in June 1996

[4] Royce, Walker. Available online 2002-04-15 at http://www.therationaledge.com/content/feb_02/f_conventionalToModern_wr.html

[5] A. Ginige, D. B. Lowe, and J. Robertson, "Hypermedia Authoring", IEEE Multimedia, Vol. 2, No. 41995.

[6] M. D. Good, J. A. Whiteside, D. R. Wixon, S. J. Jones,"Building a User-Derived Interface", Communications of the ACM (CACM), Vol. 27, No. 10, October 1984.

[7] M. Jackson, "The World and the Machine", Proc. of the17th International Conference on Software Engineering,Seattle, Washington, USA, April 1995.

[8] G. Kappel, W. Retschitzegger, and B. Schröder, "EnablingTechnologies for Electronic Commerce", Proc. of the XV.IFIP World Computer Congress, Vienna/Austria andBudapest/Hungary, August/September 1998.

[9] G. Kappel, W. Retschitzegger, "The TriGS Active Object-Oriented Database System - An Overview", ACM SIGMOD Record, Vol. 27, No. 3, September 1998

[10] G. Kappel, W. Retschitzegger, W. Schwinger, "Modeling Customizable Web Applications - A Requirement'Perspective", International Conference on DigitalLibraries: Research and Practice (ICDL), Koyoto, Japan,November 2000.

[11] P. Maes, "Concepts and Experiments in Computational Reflection", Proc. of OOPSLA87, Sigplan Notices, ACM,Oct. 1987.

[12] B. Mobasher, R. Cooley, and J. Srivastava, "Creating Adaptive Web Sites Through Usage-Based Clustering of URLs", Proc. of the 1999 IEEE Knowledge and DataEngineering Exchange Workshop (KDEX), November1999.

[13] R. Oppermann, and M. Specht, "A Nomadic Information System for Adaptive ExhibitionGuidance", Proc. of the

[14] International Conference on Hypermedia and Interactivityin Museums (ICHIM), D. Bearman and J. Trant (eds.)Washington, September 1999

[15] T. Powell, Web Site Engineering, Prentice Hall, 1998.

[16] B. Pröll, W. Retschitzegger, R. R. Wagner, and A. Ebner,"Beyond Traditional Tourism Information Systems - TIScover", Journal of Information Technology andTourism, Vol. 1, Inaugural Volume, 1998.