CLASSIFICATION AND MODELLING OF WEB TECHNOLOGIES

Mark Wallis, Frans Henskens and Michael Hannaford
Distributed Computing Research Group, School of Electrical Engineering and Computer Science
University of Newcastle, Callaghan, NSW, Australia

Keywords: Web, Classification, Model.

Abstract: The World Wide Web is a constantly changing environment in which academia, industry and interest groups participate to innovate and design the next evolution of online user interaction. The ad-hoc nature in which new web-based systems and technologies have been developed has led to an increasingly diverse environment, with ill defined interactions and fuzzy classification systems. Recently, business pioneers in the industry have attempted to classify web applications into large groupings based on certain key characteristics. The high-level taxonomy presented in this paper provides a way to scientifically classify web applications. By classifying applications and studying the progression from one classification to the next, predictions can be made as to the direction of future web application development. After presenting a formal classification model this research discusses how this model can be used to compare existing web technologies and design the next generation of the World Wide Web.

1 INTRODUCTION

The World Wide Web is a constantly changing environment in which academia, industry and interest groups participate to design the next evolution of online user interaction. The ad-hoc nature by which new web-based systems and technologies have evolved has led to an increasingly diverse environment with ill defined interactions and fuzzy classification systems. Recently, business pioneers have attempted to classify web applications into large groupings based on certain key characteristics (O'Reilly, 2005). Various attempts have been made to firm up these classifications (Hinchcliffe, 2005), but ultimately, they have been critiqued as too high level and ill defined (Gilbertson, 2007).

This paper introduces a taxonomy based on the connections between the various components involved in a World Wide Web interaction. The taxonomy is then extended to address the next generation of web technologies such as Cloud Computing (Boss et al., 2007). The intent is that this system will be extendable to cater for future evolution of the World Wide Web model. The usefulness of such a taxonomy is apparent when making comparisons between web technologies, and designing changes to how web applications and users interact.

The paper begins with a review of the pre-existing classification systems and key web technologies. The proposed taxonomy is then presented and critiqued. Finally, future development concepts are discussed with comparisons to the presented models.

2 BACKGROUND

Business has, for some time, been a key driver behind the evolution of the World Wide Web (Cockburn and Wilson, 1995). The concept of web "generations" can be traced back to 2005 where O'Reilly first keyed the phrase "Web 2.0" (O'Reilly, 2005). At the time, Web 2.0 was identified as any web notion that matched the following criteria:

- The application represents a service offering and is not pre-packaged software.
- The application data evolves as the service is used by the end users. This is in contrast to applications where static data is generated solely by the application owner.
- A framework is provided that supports and encourages user submission of software enhancements. These submissions are generally in the form of plug-ins or extensions to the web application.
- Evolution of the application is driven by the end user as well as the application owner.

Wallis M., Henskens F. and Hannaford M..
CLASSIFICATION AND MODELLING OF WEB TECHNOLOGIES.
DOI: 10.5220/0003338801510156
In *Proceedings of the 7th International Conference on Web Information Systems and Technologies* (WEBIST-2011), pages 151-156
ISBN: 978-989-8425-51-5

- The application interface supports interaction from multiple client devices such as mobile phones and PDAs.
- The application provides a lightweight, yet dynamic, user interface.

A definition of Web 2.0 is useless without a formal definition of Web 1.0. Accordingly, Web 1.0 was retrospectively specified as providing the same basic Web Application characteristics, but that the applications and their data were largely static and not user driven. The client to server interface was also stricter in Web 1.0 (O'Reilly, 2005).

The more recent web application classification has been termed Web 3.0, or the 'semantic web' (Wainewright, 2005). This model places greater focus on the following:

- Server to server communication technologies such as Web Services.
- Applications that provide information as a service. For example, an application that provides currency exchange information via an API.
- Applications that aggregate information from multiple sources and present it to the user in a value-added fashion.
- Functionality provided by the web browser using either local or remote stored components that are provided and serviced by 3rd parties.

The definitions of Web 1.0, 2.0 and 3.0 have grown out of attempts by business to classify advances in technology. These classifications have been retrofitted to bracket convenient periods of that evolution, and as a result the classification of any individual system rarely falls solely into a single group. This is primarily due to the continuous nature of web development. The classification method presented in this paper relies on feature selection and interactions rather than on consideration of how to best provide solutions to problems found in any current generation of web application design.

2.1 Web Browsers

Web browser technology has been pivotal in the evolution of the web, acting as the main interface between the user and published web applications. As the web evolves, different expectations are placed on the web browser to provide the functionality required to support the various generations of web applications.

Web 2.0 saw a push to highly interactive web interfaces. Web browsers have addressed these requirements with advances in Javascript (Flanagan, 2002) and AJAX (Garrett, 2005). These technologies introduced dynamic content to the user, but to date have been viewed as sluggish and riddled with cross-browser compatibility problems (King, 2003) (Yank, 2006). In fact, it is not uncommon that web developers give up attempting to support all the major web browser platforms. Instead, developers force their users to use a specific subset of web browsers for which they have specifically catered.

Vision past Web 2.0 reveals a much greater focus placed on client-side processing. For example, the concept of a "Serviced client" is introduced by Wainewright (Wainewright, 2005). Extensions to this, introducing the concept that a web browser can take a much greater responsibility in client to web application interaction has also been further discussed in recent publications (Henskens, 2007) (Paul et al., 2008).

2.2 Previous Analysis Work

There have been various attempts in documented research to model the World Wide Web; these basically fall into two categories.

Initial attempts focused on modelling the interactions between network nodes on the Internet and how they relate to each other using such tools as Power Laws (Faloutsos et al., 1999). These laws model interactions on the Internet as a whole, as opposed to just focusing on web-based traffic. Various projects (Opte Group, 2008) (Dodge and Kitchin, 2000) have built quite detailed maps of the Internet based on these ideas.

Analysis has also been performed from the Software Engineering perspective, focusing on how to design and model the internals of a web application. These proposals tend to be based around extensions (Conallen, 1999) to UML (OMG Working Group, 2007) and detail how to model the objects that internally make up a web application, along with limited support for modelling service-level interactions.

Recent work in this field has seen the creation of the Web Science Research Initiative. Initial technical reports (Berners-Lee et al., 2007) suggest that this group will focus on modelling the world wide web using concepts deeply founded in the Sciences.

The work presented in this paper focuses on modelling World Wide Web applications themselves by concentrating on the interactions between the actors involved in a World Wide Web transaction. By modelling these actions we can determine where the current issues in the model lie and design enhancements to usher in the next generation of web application.

3 TAXONOMY

The taxonomy proposed in this paper aims to characterise web applications based on key functionality. To make this possible, a model of actors, and the interactions between these actors in the system has been established. These interactions are represented by directed graph edges labelled with the type of flow being described. This model provides a scientific system in contrast to that currently being proposed by industry and, we believe, provides a powerful tool for testing and profiling next generational enhancements to the world wide web.

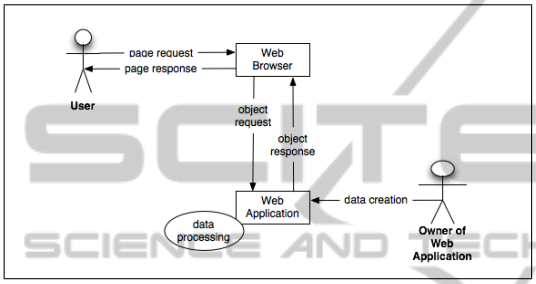

Figure 1: Web 1.0 model.

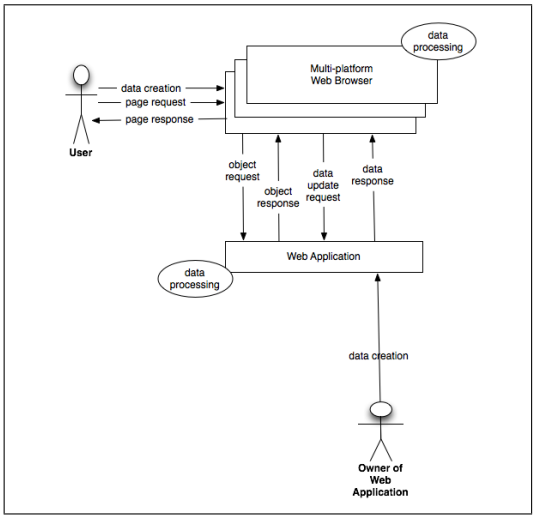

Figure 2: Web 2.0 model.

To build this model, the following key actors have been identified;

- User - the end user who interacts with the web browser to gain access to the information and services provided by the World Wide Web.
- Web Browser - the software interface between the user and the web application.
- Web Application - the software package, deployed as a service offering and hosted on one or more servers.
- Web Application owner - the entity that owns and manages the web application.
- Web Service - a software package that provides server-to-server APIs and services, generally as a source of information which the web application would transform and aggregate.

These actors can be used to represent any world wide web transaction directly involving users and web applications. In some cases, the way each actor is implemented may change from deployment to deployment. For example, not every Web Application may make use of Web services. Flows between these components can be generalised as either requests for data, or requests to create/alter data. The following will now demonstrate the use of these actors and interactions to formally define the Web 1.0, 2.0 and 3.0 concepts.

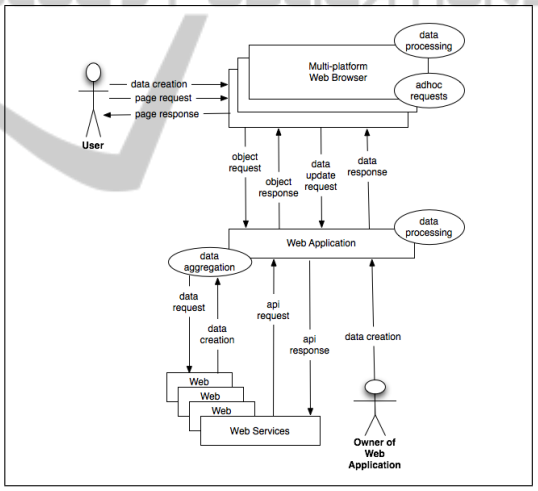

Figure 3: Web 3.0 model.

The Web 1.0 model depicted in Figure 1 represents a basic page/object request flow with the data being generated by the application owner. All of the data processing is performed by the web application with only basic web browser support required. The page request flow represents the user requesting a specific URL from the web browser. The web browser in turns discovers and connects to the web application, making the request for the specified URL. The web browser then continues to service the requests for the various objects that form the original page requested by the user. Data in the Web 1.0 model comes from the application owner. Generally, this data is in the form of static images and HTML content.

The Web 2.0 model in Figure 2 takes multiple steps forward from the Web 1.0 model. On the client side, multiple cross-browser platforms are supported, with varying degrees of protocol support. There is a much greater reliance on client-side processing to create dynamic interfaces and reduce overall application latency. Data can be created either by the application owner, or by the users. For example, user provided data is common-place in the popular social networking sites that fit this paradigm. Web browsers can also be instructed to make additional data requests independently of the user, allowing for real-time data refreshes. This is commonly implemented using AJAX technology and is a forerunner for web browsers themselves being able to directly take part in web services (Paul et al., 2008).

The Web 3.0 model depicted in Figure 3 takes the next step forward by introducing Web Services and API functionality to the back-end of the Web Application. The front-end of the model, which represents the user, the web browser, and its interaction with the web application, stays unchanged from Web 2.0. There are two sides to the web services enhancements to this model.

- In Web 3.0 the web application can expose services via an API or web service. This is represented in the model as the API request/response flows. An example is a 3rd party application making requests to the modelled web application for information on services.
- The web application itself is able to make outgoing requests to other 3rd party APIs and web services. Depicted as the data request/response flows, this information is pushed through an aggregator in the web application which can combine the data from multiple 3rd party sources. The combined data is often then represented, through the web application to the user, to provide value-added functionality. A classic example of such a service is a news aggregation website which takes multiple RSS (RSS Advisory Board, 2007) feeds from different sources and aggregates them into a single user interface.

To demonstrate further uses for this taxonomy we can use this design to model how advances in Cloud Computing affect the interactions between our actors. Figure 4 shows the model applied to a cloud computing design. Using a selection procedure and highlighting the external User and Web Browser actors we can show that cloud computing poses no change in the way that users themselves interact with the web. Experiments show that latency though will be affected due to the additional infrastructure elements of a standard infrastructure-as-a-service cloud computing design.

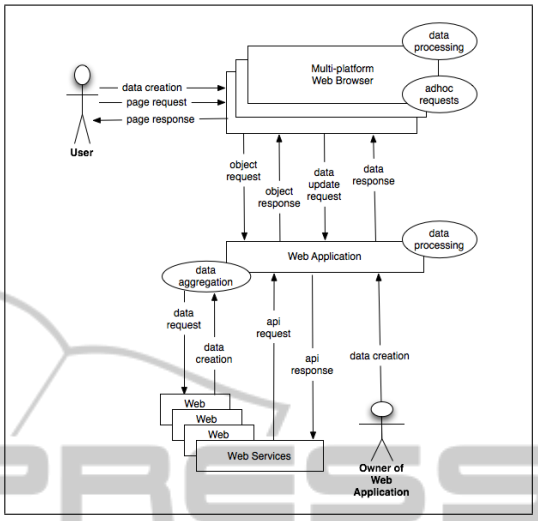

Figure 4: Cloud Computing model.

4 USING THE TAXONOMY

The usefulness of a taxonomy is based on the metrics that can be gathered through implementation of the model in a data modelling package. Examples of metrics that can be analysed using the presented design include, but are not limited to:

- Total number of messages passed between actors.
- Total bandwidth required between pairs of actors.
- Latency between request/response pairs.
- Load placed on each actor in the system.

While this data in a lone model might not provide much value, it is when comparing the information between models that the usefulness of the presented taxonomy becomes apparent. As new logical requirements emerge in web development these models can be used to compare various implementation styles and technologies to see which produces the best set of results. The taxonomy can then be seen as a key tool in the methodology of any future world wide web research.

Take, for example, the evolution from Web 1.0 and Web 2.0. One of the primary flaws identified in Web 1.0 was that it relied on the web site owner to generate content. Web 2.0 moved to a model where users themselves generated a significant proportion of the content. Of course, generating the content means that the upload bandwidth requirements between the User actor and the web application are greatly increased and

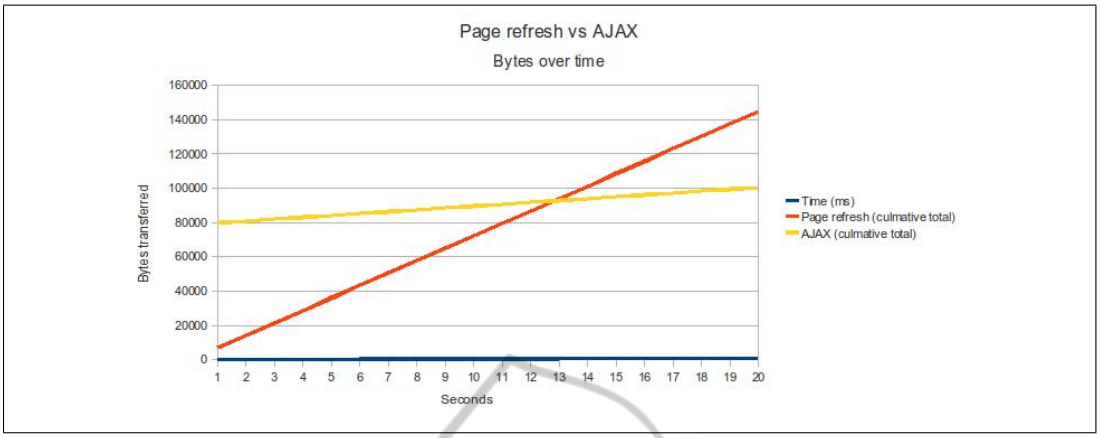

Figure 5: Page refresh vs AJAX - bytes.

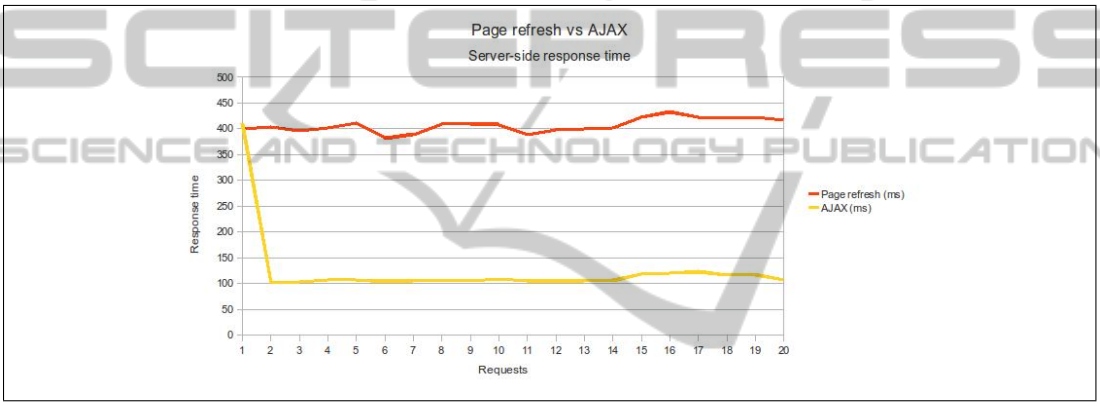

Figure 6: Page refresh vs AJAX - response time

the content itself is at risk during transit, especially if the content is sensitive. Taking the Web 1.0 model (Figure 1) and increasing the flow requirements on the graph edge between the Web Browser and Web Application has flow on effects to the other metrics presented by the model. Correct use of this taxonomy at the cusp of the Web 1.0 to Web 2.0 evolution would have highlighted the fact that the Web Browser would be spending a significantly larger portion of time stalled, waiting for data uploads to complete. Such problems would have been identified early and rectified before the user-generated content concept caught on. In fact, the solution to the large upload issue most likely would have been taken from concepts we are now seeing in Web 3.0, where the content would have been uploaded, perhaps in a compressed format, using a web service-based architecture. This effectively would have freed up the web browser to allow the user to continue using the application without having to wait for their upload to complete. In the end, proper use of such a taxonomy may have seen the Internet skip Web 2.0 all together and move from Web 1.0 to a hybrid 2.0/3.0 design.

Another example of the comparative nature of this taxonomy can be seen when using it to compare traditional Web 1.0 page-refresh approaches with Web 2.0 AJAX techniques. In Web 1.0, if the web application developer wanted to constantly update a page with data automatically, then the only option was to refresh the entire page. AJAX alternativly provides the developer with a means of providing the updated information via an XML stream that is processed behind the scenes by a Javascript AJAX library. Figure 5 shows an experiment that compares the bytes transferred between a web browser and web application over a period of 10 minutes with a 30 second refresh rate. The first series shows a traditional whole-page refresh approach where each request/response pair is the same size. The second series shows an AJAX approach where the initial response is much larger as it incorporates the AJAX library itself, but all susequent request/responses are much smaller due to the

XML stream. The results are shown as culmative bytes transferred.

The same page refresh vs AJAX situation can be measured from a latency point of view. By comparing the message round-trip processing time at the application server side of the model we can draw conclusions around how changes in the model can affect CPU utilisation on the web server. Figure 6 shows the difference in processing times for each HTTP request/response pair in the page-refresh and AJAX examples described above.

5 CONCLUSIONS

This research analysed the current state-of-play with regard to the World Wide Web and the technologies it promotes. A review was presented of the business-oriented classification of web generations, and how web browsers currently interact. A taxonomy was presented that supports further study of Web 1.0 through Web 3.0. This taxonomy provides a way to compare and contrast web generations, supporting technical modelling and analysis of past, current and future generations. The taxonomy allows us to model the interactions and analyse issues that need addressing in the current model, providing a way for future web development projects to compare and contrast different designs and technologies. Enhancements presented in other concurrent research (Wallis et al., 2010) will rely on this model to prove that changes in interactions between actors will not adversly affect the performance of the World Wide Web as a whole.

REFERENCES

Berners-Lee, T., Hall, W., Hendler, J. A., O'Hara, K., Shadbolt, N., and Weitzner, D. J. (2007). A framework for web science. *Foundations and Trends in Web Science*, 1:1–130.

Boss, G., Malladi, P., Quan, D., Legregni, L., and Hall, H. (2007). Cloud computing. http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/Cloud_computing_wp_final_8Oct.pdf.

Cockburn, C. and Wilson, T. (1995). Business use of the world-wide web. *Information Research*.

Conallen, J. (1999). Modeling web application architectures with uml. *Communications of the ACM*, 43(10):63–70.

Dodge, M. and Kitchin, R. (2000). *Mapping Cyberspace*. Routledge.

Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On power-law relationships of the internet topology. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*. SIGCOMM: ACM Special Interest Group on Data Communication, ACM.

Flanagan, D. (2002). *Javascript: the definitive guide*. O'Reilly.

Garrett, J. J. (2005). Ajax: A new approach to web applications. *Adaptive Path 2005*.

Gilbertson, S. (2007). Jakob nielsen on web 2.0: "glossy, but useless". *Wired Magazine*.

Henskens, F. (2007). Web service transaction management. *International Conference on Software and Data Technologies (ICSOFT)*.

Hinchcliffe, D. (2005). Visualizing web 2.0. *Social Computing Magazine*.

King, A. (2003). Optimizing javascript for execution speed. *devnewz*.

OMG Working Group (2007). Omg unified modeling language (omg uml), omg unified modeling language (omg uml), infrastructure, v2.1.2. Technical report, Object Management Group.

Opte Group (2008). Opte. http://www.opte.org/.

O'Reilly, T. (2005). What is web 2.0. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

Paul, D., Wallis, M., Henskens, F., and Hannaford, M. (2008). Transaction support for interactive web applications. In *4th International Conference on Web Information Systems and Technologies (WEBIST-2008)*, volume 4th of *WEBIST*. INSTICC.

RSS Advisory Board (2007). *RSS 2.0 Specification*, 2.0 edition.

Wainewright, P. (2005). What to expect from web 3.0. *ZDNet Business*.

Wallis, M., Henskens, F., and Hannaford, M. (2010). Expanding the cloud: A component-based architecture to application deployment on the internet. In *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*.

Yank, K. (2006). Is ajax cross-browser? *Sitepoint*.