

# AJAX Enabled Property Management

Author. Mallorie, J.

**Abstract**—This paper analyses different AJAX development technologies. It evaluates the feasibility of one in particular in relation to developing an AJAX enabled Web application for enhancing the management of properties and property details. The core functionality needed and the issues which arise in development are also analyzed. Features of AJAX in relation to specific benefits and limitations of the application are viewed critically for recommendations.

*What is AJAX, Evaluation, Rich Internet Applications, RIA's.*

## I. INTRODUCTION

Property sales and letting companies often find it difficult to keep up to date records of property, client and partner details. This is due to the large amount of data tracking and modification needed within this industry, and the pace which it is needed to be modified. Updating a corporate Web site with property data would certainly be a continuous job. However, the data *is* stored, and records *are* updated. Every company needs to know if a property is still for sale, or if a new letting has come up. The question is; why can we not use that *same data*, without duplicating for the Web? With the use of AJAX, the answer is, you can! More companies are turning to the Web to improve their business processes and to capitalize on new interactivity and business performance that AJAX offers, over the standard HTML model of on-line company brochures.

## II. WHAT IS AJAX?

Contrary to most computing abbreviations or acronyms, AJAX is not a specific technology or language, but a technique using multiple technologies together to deliver a more dynamic experience in standard browsers. Developers have been trying to find a way to get around the standard HTML model[4] and the problems that are associated with it, such as the constant annoyance of the *request-response* nature of the Web. Before AJAX, Web sites and applications have historically requested data from the server, and on the server response, re-render the entire page with the new data in place. The hyperlink architecture or, distributed Hypermedia architecture[8] is what the Internet was founded upon. This full page re-fresh causes many issues for the user, such as long waits for server response and the possibility of duplicating payment actions by navigating back to a payment screen and re-processing data[20].

The AJAX Web model attempts to solve the *request-response* problems by creating asynchronous[9] connections

with the server, processing and re-rendering only parts of the page that need to be modified. Asynchronous connections are the main characteristic of an AJAX enabled page. AJAX is an acronym for asynchronous JavaScript and XML[9], as the name suggests AJAX uses a combination of technologies to create a dynamic experience that cannot be achieved in the HTML Web model alone. Using technologies mostly developed in the 1990's[19] such as JavaScript, CSS, DHTML, DOM, XSLT, XPATH and XML you can create pages that have a highly interactive desk-top application interface, removing the 'blink' between full page re-rendering.

Rich Internet Applications (RIA's), or Rich Web Applications have lots to offer. There has been a boom in development IDE's, Frameworks, and AJAX 'all-in-one' applications which all claim to make the development of RIA's easier than the last.

## III. HOW IT WORKS

### *ECMA Scripts*

Triggers embedded in the page make function calls to client side code written in ECMA script. The functions then call the XMLHttpRequest object and connect to the server in the. The ECMA script used in most cases is JavaScript, however other scripts available[22].

### *XMLHttpRequests*

The reason this is possible using existing technologies is because of the introduction of the XMLHttpRequest object [12]. When an event is triggered on the page, the XMLHttpRequest object runs in the background enabling partial page updates without re-rendering the entire page.

### *Messages*

Due to the standardization of XML in 1998, we now have a standard cross platform messaging format that is used to transport messages from client to server and vice-versa. Even though the magic behind AJAX is called XMLHttpRequest there is no requirement that the messages sent between client and server are formatted this way and other options are becoming popular such as JavaScript Object Notation (JSON) [11].

### *Web services*

Adding a business logic layer[18], separating logical layers, or implementing Service-Oriented Architecture (SOA) [24] to a Web Application can be done using AJAX. Once the asynchronous connection has been made to the server, it is not restricted to just returning an evaluated string, or record from a database but can also initiate Web services.

Arguably, Web services can be used to add versatility and value to your business using RPC, SOA and REST [13]

### Document Object Model (DOM)

The DOM is the standard Object model for representing HTML and related documents [4]. It was never designed with the intention of partial re-rendering, however once the XMLHttpRequest has returned from the server with the data, the DOM is manipulated to partially re-render the page.

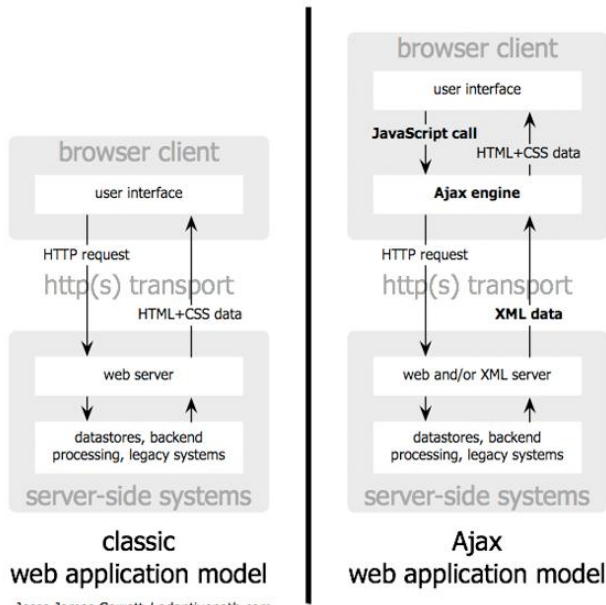


Fig 1. The classic and AJAX web models

## IV. BENEFITS OF AJAX

With the advent of AJAX, the developer can separate the data, from format, from style and functionality. More dynamic and user friendly applications are coming to the Web, and the benefits are plentiful:

**Real-time form data validation:** Form data such as user IDs, serial numbers, postal codes, or even special coupon codes that require server-side validation can be validated in a form before the user submits a form.

**Autocompletion:** A specific portion of form data such as an email address, name, or city name may be autocompleted as the user types.

**Load on demand:** Based on a client event, an HTML page can fetch more data in the background, allowing the browser to load pages faster.

**Sophisticated user interface controls and effects:** Controls such as trees, menus, data tables, rich text editors, calendars, and progress bars allow for better user interaction and interaction with HTML pages, generally without requiring the user to reload the page.

**Refreshing data and server push:** HTML pages may poll data from a server for up-to-date data such as scores, stock quotes, weather, or application-specific data.

**Partial submit:** An HTML page can submit form data as needed without requiring a full page refresh.

**Mashups:** An HTML page can obtain data using a server-side proxy or by including an external script to mix

external data with your application's or your service's data.

**Page as an application:** AJAX techniques can be made to create single-page applications that look and feel much like a desktop application[17].

## V. IS IT ALL GOOD NEWS?

### Complexity

Because of the complexity of developing AJAX aware applications, the skill set required for such development increases substantially. A Web developer who is capable of creating static pages for a company may now find that this kind of RIA development requires more training. It is also clear that due to the complexity and data accuracy needed in business, more *time* will be needed to plan, implement and test such applications, again another issue which will incur *cost*.

### Security

The AJAX framework relies on server-side programming, client-side programming and asynchronous connections. The complexity of this framework alone poses a threat to the security of the 'system' as a whole. A complex study has been carried out highlighting specific areas of risk. [21]

### Web usability Guidelines

Updating elements of a page without refreshing the page seems like a good idea, however enabling AJAX functionality on a Web-page breaks certain Web usability guidelines[2] such as; breaking the back button; changing state with GET requests; history caching and scrolling pages, disorienting the user. While many of these problems are slowly being solved there are still issues regarding usability in AJAX enabled pages.

Careful consideration should be made when deciding on the implementation of RIA's. Costs mount quickly for developer training, development and testing time, information safeguarding and user/employee analytics. The *return on investment* may now be less substantial than previously thought.

## VI. BACKGROUND TECHNOLOGY

Now that dynamic pages and full RIA's can be developed there are added problems for the Web developer. Web pages now require more skills in more languages to produce this dynamic AJAX application, and the debugging of such applications is drastically different and much more challenging.

### Frameworks Toolkits & IDE's

The development of AJAX enabled applications can be developed outside of frameworks, Toolkits and IDE's, however due to the complexity of the design and languages needed to create such applications, development environments are being developed at a rate of up to 'one new framework per day' [15]

With each new development environment, offering a host of widgets, libraries, and built in modules to aid development and debugging time. It can be difficult to evaluate and decide on the best technology to use, for your company's needs. The different AJAX technologies available can be grouped into three main types:

**Direct AJAX Frameworks** – Small frameworks, for development of Web sites with functionality such as shopping carts. The Developer is expected to author pages directly in HTML.

**AJAX Component Frameworks** – Larger frameworks offering pre-built components. Generally used for Web applications instead of sites.

**Server-Driven AJAX Frameworks** – Server-side and client-side development, most of the functionality is programmed on the server and data and rendering information is then passed between the client and server[3].

There are many comparison sites and forums found across the Internet which can help decision making, however many of these sites may be *sponsored* and may not be giving totally impartial comparisons. I have taken a cross section of popular AJAX technologies available, which advertise benefits sufficient to satisfy the business needs.

### Google Web Toolkit

License: Open-Source

Platform: J2SE 1.4.2 or earlier

Language: JAVA – JavaScript

“Google Web Toolkit (GWT) is an open source Java development framework that lets you escape the matrix of technologies that make writing AJAX applications so difficult and error prone.” [10]

The GWT is a Java centric product, which means that development of the AJAX Web application can be done in any IDE with Java support. The Sun Java platform is also open source and after running a few command line Java commands, you can open the GWT project in a multi-language IDE. Once you develop the AJAX application in Java, the GWT compiles the Java to JavaScript.

GWT gets around debugging issues by allowing you to debug your application in your own IDE before the JAVA code is converted to JavaScript. The GWT framework allows two modes. The first, called *Hosted Mode*, allows you to build, run and debug your application using the Java Virtual Machine (JVM) locally. When you are ready to deploy the application for end-to-end testing or implementation, you run the GWT *Java-to-JavaScript compiler*, which converts the project from Java code to JavaScript *Web Mode*. The project can then be run in the browser with no need for the JVM or any other plug-ins. This conversion from JAVA to JavaScript enables any project built with the GWT to run on almost any browser.

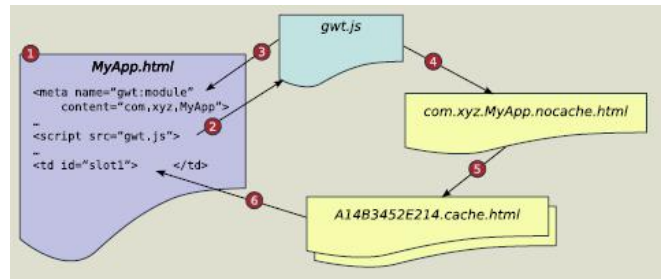


Fig 2. The GWT load sequence from Hosted to Web mode

The GWT has two main Libraries, the *JRE Emulation library* and the *GWT Web UI Class library*. The *JRE Emulation library* incorporates many of the Java classes such as *Java.lang* and *Java.util* however the rest of the standard libraries are not supported[7]. The *GWT Web UI Class library* is Google's own 'widget' toolbox, providing a set of custom classes to help you create Web UI's such as text inputs, buttons and images.

Unfortunately, all of the benefits of developing in Java and converting to JavaScript mean that the client will be processing more scripts. The compiler does strip whitespace and shortens variable names, but this still means lots of code to download[6]. Whilst GWT offers debugging functionality it is still very limited in the shadow of other such language native IDE's.

### Backbase

License: Commercial \$2,000

Platform: JavaServer Faces

Language: JavaScript / declarative XML

The Backbase framework boasts over 250 built in widgets and functions, which can be easily customized, to help with AJAX development. Unlike standard AJAX Web applications the Backbase framework requires a 'player' download called the Backbase AJAX Client Runtime.[5] The Backbase Client runtime is small however, and does not require an install so the user is unaware of such a download. The Backbase Client Runtime is responsible for page rendering, and the abstraction of the HTML DOM.

Programming the Backbase Client Runtime can be done with BXML, the Backbase declarative user interface language based on XML. Similar to XML this script can define user elements, apply styling and can even start asynchronous connection to the server. The Backbase framework supports the entire DHTML stack and like many AJAX framework, have solved some usability web guide line issues such as the broken back button.

Shown in *Figure 3*. even though the Backbase server is based on a Java technology called JavaServer Faces(JSF), it can still support a multitude of other languages. Backbase also support Real-time browser based debugging, a feature which has been called 360-degree debugging.

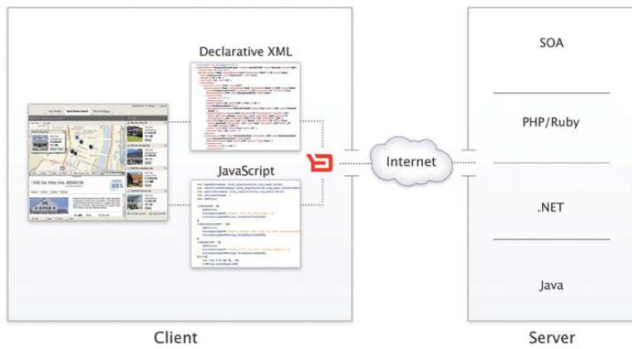


Fig 3. The Backbase server / client architecture

### Adobe Flex 2.0

License: Commercial \$249.00

Platform: ActionScript

Language: JavaScript / MXML

"Flex Builder provides a complete IDE for Flex development. Editors for MXML, ActionScript, and CSS support developers with syntax coloring, statement completion, code collapse, quick fixes, and more." [25]

Flex is a contained IDE from Adobe built on Eclipse[1], and is not multi-language compliant. Flex uses the ECMAScript Edition 3 [22] compliant script, ActionScript3.0 instead of the more common JavaScript used in AJAX programming.

Similar the BackBase architecture, Flex also runs a client side engine. Very few people may even realise as market penetration for the Adobe flash player has reached over 98% of hosts connected to the Internet. Another similarity between Flex and Backbase is the declarative user interface language, in flex called MXML[25].

## VII.PROPERTY MANAGEMENT WEB APPLICATION FEASIBILITY IN ASP.NET AJAX

This section explains important areas of concern relevant to developing a property management Web application, in brief; *libraries and languages, automated tools, asynchronous connections, Web services and data retrieval.* This section also explains the development approach, and evaluation of the technology, each area at a time.

Thanks to the availability of the free *lite* version of Visual Studio (Visual Web Developer) and the features and libraries available through the .NET Framework, a sample proof of concept application was made using the ASP.NET AJAX technology.

### Overview

ASP.NET AJAX is a technology based on the Microsoft(c) .NET Framework, commonly used to develop rich windows applications[16]. ASP.NET AJAX uses a sub-set of the .NET Framework called ASP.NET which incorporates Web specific design tools and a set of readymade features specific for AJAX application development.

With no real measure as to the potential database size in the property industry, and the extent of future development in a property management AJAX Web application, ASP.NET AJAX offers a variety of options. The AJAX Web toolkit is a free component to aid development of AJAX applications and the support site includes a plethora of installation, and development guides, documentation and video support.

The development environment for ASP.NET AJAX can be chosen from the variety of Microsoft products such as VisualStudio 2005 or VisualStudio 2008, these environments are expensive. However to compete with other open source products Microsoft have released a free version of VisualStudio for AJAX development called Visual Web Developer. The VisualStudio environment makes the creation of simple Web services and AJAX aware Web Applications simple. With the use of ASP.Net AJAX Toolkit plug-in, the Visual Studio toolkit gives the developer a clear graphical representation of the tools, and AJAX widgets needed to build AJAX applications. All standard debugging features common with Visual Studio are available such as setting breakpoints, code validation and intellisense. VisualStudio also incorporates a localhost server, Internet Information Services (IIS) and local database management features for ODBC, SQL and more, to ensure compatibility and provide a full sandbox environment for project development.

### Libraries

Because ASP.NET AJAX is based on the .NET Framework, it supports almost any of the features available in the .NET class library [14]. The ASP.NET Web application runs in the Common Language Runtime (CLR) on the server. The CLR is the engine that supports all .NET Languages.

ASP.NET AJAX incorporates a client-side library called the 'Microsoft AJAX Library' which includes pre-built classes to help the developer with common AJAX client-side script, such as browser identification and compatibility, pre-defined controls and debugging services.

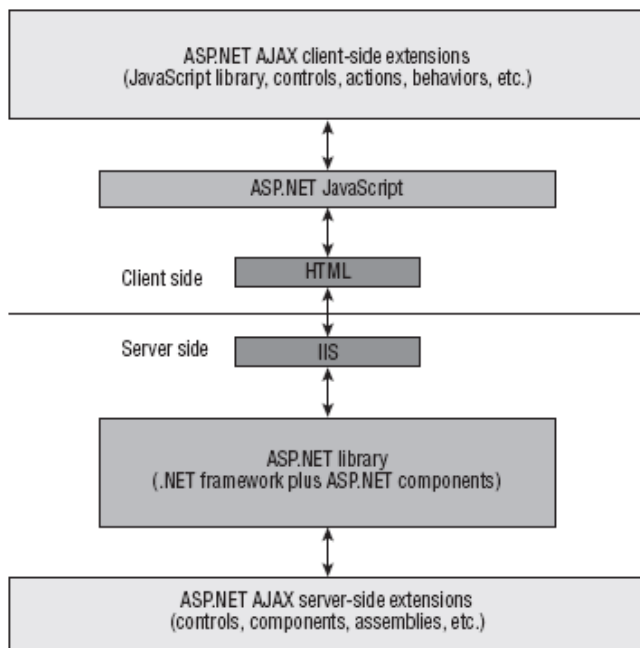


Fig 4.0. ASP.NET AJAX Architecture

### Languages

ASP.NET AJAX allows for server-side programming in any Microsoft language, Visual CSharp, Visual Basic or Visual Jsharp, as well as most client-side scripting languages creating the ability to develop fully heterogeneous systems. ASP.NET AJAX does not use XML messaging, instead JavaScript Object Notation (JSON) is used as the native message format, this is only the native format however and XML is fully supported.

To access the client-Side scripts the Default.aspx Web page must include a Script Manager Object. The Script Manager includes functionality, which acts as an intermediary between the client and the server, it talks to the server and automatically pulls down all the resources needed from the server such as the appropriate AJAX runtime's and cross browser compatibility scripts, saving the developer manual coding time. Once the Script Manager is in place, creating AJAX enabled Web pages is simply a case of dragging and dropping controls onto stage.

### Automated Tools

The extensive toolbox options inside VisualStudio gives you the flexibility and error prevention not obtainable when developing outside of an IDE or pre-defined framework. Placing and AJAX control on stage generates the code behind the visual representation automatically. The intuition of this environment is advanced. Dropping standard HTML controls to stage, create automated HTML code in the document, and opening the *click* action of a HTML control will automatically generate JavaScript and so on.

Installing the AJAX Web toolkit incorporates new AJAX ready controls, which can be dragged direct to stage. Since much AJAX functionality is an extension of the HTML model, so to are the controls found in the AJAX Web toolkit. Dragging an AJAX extender control over an existing HTML control will *extend* the functionality of the control,

which can then be manipulated by opening the properties dialog.

You can use any event from controls on the page to trigger an event. For our purposes we could use a button to limit the results from a results set for the user, but a more user friendly way would be to trigger the event on the changing of a drop down list or on the input of data with a dynamic update field.

### Asynchronous connections

AJAX can be implemented in the company's Web site to create a richer user experience, and to meet company objectives. The proof of concept sample application shows the use of multiple *updatePanels*, which call the system time on the server and re-render parts of the page independently to the main page. In practice this functionality can be used to provide real time form validation of user input or auto complete town or county fields.

The most beneficial tool in the AJAX Web toolkit is the *updatePanel*. Using the *updatePanel*, part of the AJAX Web Toolkit allows you to define parts of the Web application you wish to update independently from the rest of the Web application in true asynchronous fashion. This partial page re-refresh increases network efficiency. A common misconception however is that the partial re-refresh in ASP.NET AJAX, sends only the data needed to re-render the page to the server, this is not the case. By default, a called event within the same *updatePanel*, posts all of the data from the client to the server, and the entire page life cycle on the server is then executed. Only the data needed to re-render the *updatePanel* is then sent back to the client for the partial page re-render. This does increase network efficiency, but sending lots of data to the server for a small page refresh could slow the network.

The *updatePanel's* default asynchronous postback is set as 'always' this means that if any action triggers a partial refresh, by default all of the *updatePanel's* in the Web application will be re-rendered. You can change this default action to allow the partial refresh to update on specific triggers only. This is useful, for example, if a user would like to update only the current list of properties, or the current location, without refreshing any other setting they have already made. Browsers, which support the *updatePanel*, include, Internet explorer, Mozilla / Firefox and Safari.

### Web services

Using ASP.NET AJAX to create Web services for customers and employee's, the company can turn stored data into valuable information, which could spin profits instead of collecting dust. To add a Web service to a Web application, you simply reference each Web service you wish to include from the server in the script manager. A local reference can be used if the service is contained within the same project, or in the same directory, however if you wish to invoke a Web service from another solution or application, you will need to change the path so that the Web application can resolve the path. The path should be the

location of the file name only, not the method name of the Web service. Once the reference is resolved, the Web service will then know to reply in JavaScript so the methods can run in the native browser language.

The Web services run on the server in a separate process in the CLR. Each Web service written ASP.NET AJAX implements the *System.Web.Services.WebService* from the .NET Library and therefore inherits all classes within for use in your specific Web Service. However, to manipulate and trigger these services on the client-side from JavaScript, you need to add references to *system.Web.Script.Services.ScriptService*.

Http is a stateless protocol so there is a risk of sending a request to a Web service that will not reply, for example if the server does not exist anymore. Adding a precaution such as a timeout to Web service calls is a good strategy.

The proof of concept application shows an example of a Web service called on the server, the web service processes a number and returns a result. The ability to incorporate Web services on the company site for customer analytics, and mobile property updates is a valuable feature.

#### *Data Retrieval*

The IIS and Database management features of VisualStudio are clear and straightforward to use. Incorporating existing data into the database is simple, and the creation of new databases, tables and queries is as expected with database query languages. Using the Data controls from the toolbox invokes a wizard to help with data connections, and automated display features such as paging and adding custom themes make the development process of data retrieval less complicated.

Pulling data from a file or database can be coded directly into the underlying HTML using the JavaScript XMLHttpRequest object. However, for the implementation of such business objectives as we have, scalability is an important issue and the creation of bespoke Web services for portability and re-usability would be wise.

The proof of concept application shows data dynamically populating the *datagrid* on the page. With each re-ordering of the information on the *datagrid*, the application re-renders the *datagrid* independently of the main page. The system timer on the top of the page indicates the last full page refresh. The ability to manipulate data asynchronously on the company Web site will avoid network delays and provide a better user experience for our customers.

#### VIII.ASP.NET AJAX EVALUATION SUMMARY

With good planning, creating a *new* AJAX Web Application using VisualStudio and ASP.NET AJAX can be very simple, and the creation of services and functionality that we require for our business objectives is heavily automated. However, modifying an existing HTML Web Model Application brings a new set of considerations. In ASP.NET, to add AJAX functionality to an existing Web application, many modifications need to be made to the Web.config file before

the ASP.NET AJAX extensions become activated. Once the Web.config file has been updated, all of the AJAX extensions available from the AJAX Toolkit can be used and added into the existing application.

#### *Installation and Setup*

Based on a system with no pre-requisite software for AJAX development, you can download, install and set-up Visual Web Developer, ASP.NET AJAX, and the extensions and toolkits free of charge. The Support for installation and setup is superb, and sample applications and source code snippets help with development. However there are some known issues with installation version control of the .NET Framework[23].

#### *Development Aspects*

For a developer, the environment is rich with reusable code in a clear visual display; however for a new user of the environment it could seem overwhelming. The integrated compiler, server browser, document browser and many other features make development in this environment straight forward. One of the many benefits of AJAX is network efficiency. It seems logical that sending less data will result in a shorter round trip. However, with ASP.NET AJAX it is not quite that simple. When a user triggers an event, the client-side data is sent in full to the server, not just one small variable. If the client has lots of stored data, or starts to make many server requests, it could reduce network performance not improve it.

#### *Fit For purpose*

Visual Web Developer and accompanying technologies are fit for purpose with a few minor considerations. There could be issues with waiting time for responses. If the runtime is running on the server when the user triggers another event, the current client-side trigger will not be processed.

#### IX.CONCLUSIONS AND RECOMMENDATIONS

Rich Internet Applications have proved to be good investments for companies in the past, and the increasing number suggest that more companies are willing to invest in such methods to achieve business goals. The implementation of such Ajax enabled Web pages is not a decision that is to be taken lightly however. Shear programming and language complexity creates an abundance of risks highlighted previously in the report. The overall value of the application and the benefits, which may be seen, must be clear. Ajax web application development should not be confused with standard web development. The skills, time and money required to develop an Ajax application must be justified by the overall goals, to ensure a return of investment. With this in mind choosing an Ajax technology from the virtual spaghetti of options is still difficult. The primary factor for this decision again falls back on the results wanted from the application. While the Adobe Flash platform (Flex) is highly interactive and visual the data layer is poor. The GWT is simple in design and excellent for browser compatibility, but can be prone to slow initial download speeds. Each Ajax technology has a unique set of characteristics, benefits and

limitations. ASP.NET AJAX is no different, where it lacks in visual diversity it makes up in data management. For the needs of this company and the set of specific business objectives, I believe there are few other technologies which can match the Visual Web Developer IDE with the Ajax plug-ins and tools.

The Web application needed is mainly data centric, and there are few frameworks which have the background knowledge in data management the Microsoft product. The set-up cost, using the Visual Web Developer is zero, as are all of the associated plug-ins and tools. The development support from download to implementation is detailed and abundant making this a suitable technology to pick up and use with no major outlay. The proof of concept application proves that the business goals are achievable in a short amount of time. The application can be extended to increase more functionality and the application is cross-browser compatible.

#### ACKNOWLEDGMENTS

All the companies/products/services names are used for identification only and may be the trademarks of their respective owners.

#### REFERENCES

- [1] Adobe Systems Incorporated Anon, 2006. *Flex2, technical overview*. San Jose, CA.
- [2] Anglin, T., *The Ajax Papers*. Available from: [http://www.code-magazine.com/downloads/Telerik\\_AJAX.pdf](http://www.code-magazine.com/downloads/Telerik_AJAX.pdf) [Accessed 21 January 2008].
- [3] Anon, 2008, *Ajax Framework*. Available from: [http://en.wikipedia.org/wiki/AJAX\\_framework](http://en.wikipedia.org/wiki/AJAX_framework) [Accessed 21 January 2008]
- [4] Apparao, V., et al. 1998. *Document Object Model (DOM) Level 1 Specification*. (Version 1.0). Available from: <http://www.w3.org/TR/REC-DOM-Level-1/> [Accessed 21 January 2008].
- [5] Backbase, 2007. *Backbase Enterprise Ajax 4 Evaluator's Guide*. Available from: <http://docs.backbase.com/docs/Backbase-Ajax-Evaluators-Guide.pdf> [Accessed 21 January 2008].
- [6] Becker, D., 2007. *Tech Review: Google Web Toolkit*. Available from: <http://devblog.glowday.com/2007/05/tech-review-google-web-toolkit.html> [Accessed 21 January 2008].
- [7] Burnette, E., 2006. *Google Web Toolkit. Taking the pain out of Ajax*. United States of America: The Pragmatic Programmers LLC.
- [8] Fielding, R.T., Taylor, R.N., 2002. Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology*, 2, 115-150.
- [9] Garrett, J.J., 2005. *AJAX: A new approach to Web applications*. Available from: <http://www.adaptivepath.com/ideas/essays/archives/000385.php> [Accessed 21 January 2008].
- [10] Google., 2008. *Google Web Toolkit*. Available from: <http://code.google.com/webtoolkit/overview.html> [Accessed 21 January 2008]
- [11] Kelly, S., *Speeding Up AJAX with JSON*. Available from: <http://www.developer.com/lang/jscript/article.php/3596836> [Accessed 21 January 2008].
- [12] Kesteren, A.V., 2007. *The XMLHttpRequest Object*. (Version 20071026). Available from: <http://www.w3.org/TR/XMLHttpRequest/> [Accessed 21 January 2008].
- [13] Korotkiy, M., Top, J., *Onto,SOA: From Ontology-enabled SOA to Service-enabled Ontologies*. Available from: <http://www.gartner.com/resources/114300/114358/114358.pdf> [Accessed 21 January 2008].
- [14] MacDonald, M., 2007. *Beginning ASP.NET 3.5 in VB 2008*. 2<sup>nd</sup> ed. New York: Springer-Verlag
- [15] Mesbah, A., Deursen, A.V., 2006. *An Architectural Style for Ajax*. Delft University of Technology Software Engineering Research Group Technical Report Series, 2628 CD Delft The Netherlands: Software Engineering Research Group.
- [16] Microsoft Anon, 2007. *ASP.NET AJAX Overview*. Available from: <http://www.asp.net/AJAX/Documentation/Live/overview/default.aspx> [Accessed 21 January 2008].
- [17] Murray, G., 2006. *Asynchronous JavaScript Technology and XML (Ajax) With the Java Platform*. Available from: <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/> [Accessed 21 January 2008].
- [18] Papazoglou, M.D., Georgakopoulos, D., 2003. Service-Oriented Computing. *Communications Of The ACM*. (46), 10, 25-28.
- [19] Paulson, L.D., 2005. Building RichWeb Applications with Ajax. *IEEE Computer*, 14-17.
- [20] Smith, K. 2006. Simplifying Ajax-Style Web Development. *IEEE Computer*, 98-101.
- [21] Stefano, P.D., Giorgio F., 2006. *Subverting Ajax*. 23rd CCC Conference.
- [22] The ECMA General Assembly, 1999. *ECMAScript Language Specification 3rd Edition, Standard ECMA-262*.
- [23] Various, 2008. *Installation and Setup*. Available from: <http://forums.asp.net/t/1208539.aspx> [Accessed 21 January 2008]
- [24] Woods, D. 2003. *Enterprise Services Architecture*. Sebastopol, CA 95472: O'Reilly Media Inc.
- [25] Zalez, T., 2007. *Flex2 Development Guide*. Available from: [http://download.macromedia.com/pub/documentation/en/flex2/flex2\\_devguide.pdf](http://download.macromedia.com/pub/documentation/en/flex2/flex2_devguide.pdf) [Accessed 21 January 2008].