



## Literature Synthesis

16 July 2007

Marc Pelteret  
Department of Computer Science  
University of Cape Town  
Rondebosch, 7701  
South Africa  
mpeltre@cs.uct.ac.za

Project Supervisor: Dr Hussein Suleman (Computer Science, UCT)

## ABSTRACT

The Cellphone Shopper project aims to make grocery shopping easier by using technology to aid the process. Key to the project is the management of a shopping list which is accessible by multiple people through two interfaces: a cellphone and a Web site. This paper looks at existing material that is relevant to the development of the Web interface. Related shopping systems and Web sites that maintain lists are discussed and compared to Cellphone Shopper. The AJAX Web development technique is studied, together with Web sites that make use of it. Several AJAX frameworks are also considered in order to determine which is appropriate for the Cellphone Shopper project. The Yahoo! User Interface Library was chosen to be used. Lastly, efficient communication is important for a Web application, particularly an AJAX one, so four light-weight communication protocols are examined for use in the system: XML-RPC, SOAP, REST and JSON-RPC.

## 1. INTRODUCTION

The key aim of the Cellphone Shopper project is to make grocery shopping easier by using technology to aid the process. At its core is the management of a shopping list stored on a central server which is accessible by multiple people through two interfaces: a cellphone and a Web site.

The typical use case is that of one user creating the shopping list for the current day or week and another person viewing that list on a cellphone while they are in the store doing the shopping.

This paper will look at the techniques and technologies that will be used in developing the Web end-user interface for the system.

The paper begins by looking at related systems, specifically two shopping systems and three Web sites that maintain lists and offer services based on those lists.

The aim is to create an interface that is interactive and simple and quick to use. To do this, the AJAX Web development technique will be used. In the Section 3, AJAX will be examined more thoroughly and examples of its uses given. A selection of AJAX frameworks also will be examined.

Linked to AJAX is the topic of efficient communication, as it directly affects interactivity. This will be studied in Section 4. Finally, Section 5 will conclude the paper.

## 2. RELATED SYSTEMS

In this section, two types of systems will be examined: shopping systems and Web sites that manage lists and offer services based on them. They will be compared to Cellphone Shopper to see if they have any features that could be used in the project.

## 2.1 Shopping systems

Two shopping systems will be looked at: a mobile shopping assistant, developed by Wu and Natchetoi [30], and Safeway UK and IBM's Easi-Order [4].

The mobile shopping assistant is a J2ME application that runs on a cellphone. It allows a customer to access Web Services published by a store while they are in that store. Example services include access to product descriptions and promotions and the ability to locate products within the store.

The customer begins their shopping by creating a shopping list of products or product categories. To help the customer find an item, the assistant displays a floor plan showing the user's current location, the location of the product and the shortest route.

Easi-Order is a PDA-based application that aims to extend Safeway UK's Collect & Go system, an ordering service that allows customers to place orders remotely via phone or fax and pick up their bagged groceries at the store the following day. Easi-Order extends this service by allowing customers to use a PDA to place orders.

Central to the system is a personalised shopping list which is a list of items a customer has previously bought. The system also lists items on special, items that customers can buy with Safeway loyalty points and products it recommends based on the customer's shopping habits. From these the customer draws up a list of items they want currently. They can also request items not on these lists.

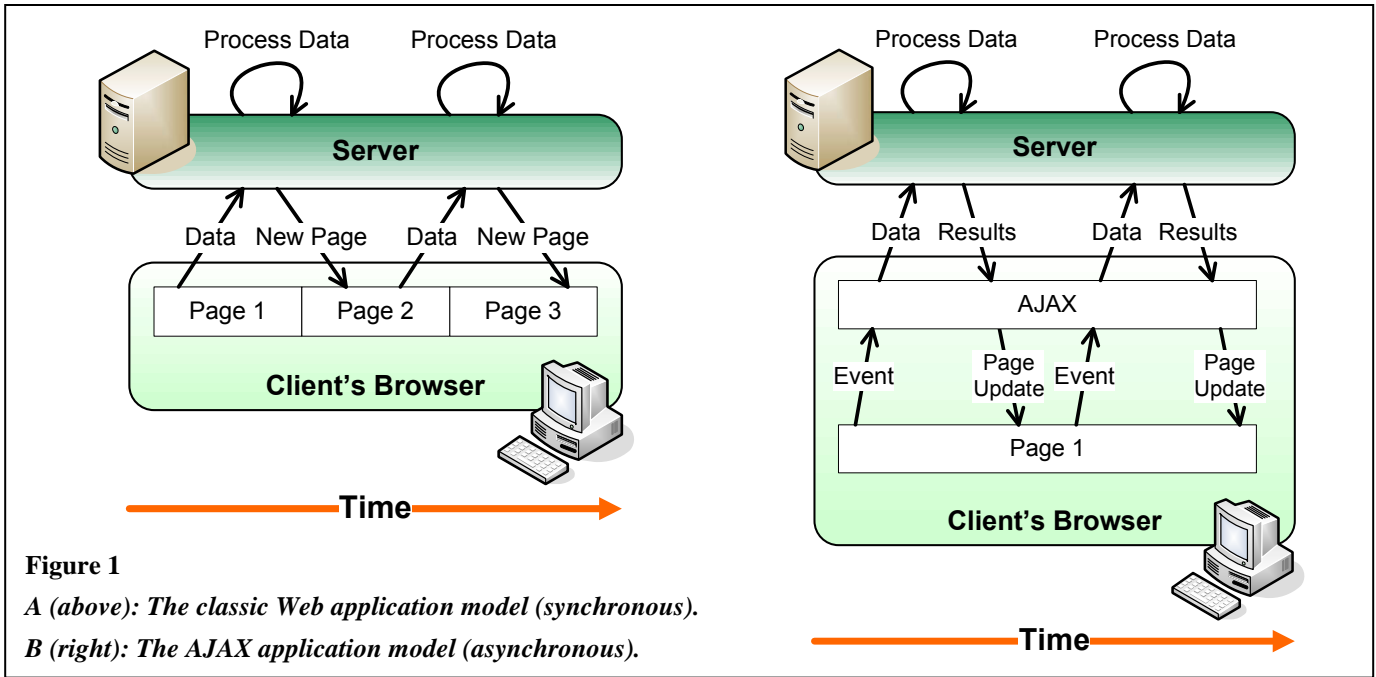
Like the mobile shopping assistant, Cellphone Shopper will also be a J2ME application and will also display floor plans and shopping routes. However, Cellphone Shopper will not be a store-based system – it will be accessible anywhere. While Cellphone Shopper will provide a history of previous shopping lists, this history will not be the focal point as it is with Easi-Order.

## 2.2 List management and services

Three Web sites will be examined in order to see what lists they maintain, how these lists are managed and what services are offered for or based on these lists.

Amazon.com [2] maintains some lists for a specific user and offers several services based on these lists. Here are just some of the lists it maintains:

- Recently Viewed Items
- Open & Recently Shipped Orders
- Shopping Cart
- Wish List
- Gift List
- Shopping List
- Wedding Registry
- Amazon Friends
- Interesting People



- Reminders
- Listmania! Lists

Except for the first two lists, all of these are user managed. Listmania! is a relatively new feature. It allows users to maintain lists of products they find interesting and share them with others. You can view a list and select items from it to add to your Shopping Cart or Wish List.

One service that Amazon offers is that of recommendations, where it recommends products that you might be interested in buying. The products are chosen by examining the items you have purchased, items you have told Amazon you own and items you have rated, and then comparing your activity on the site with that of other customers. The list of recommended items may change daily and can be quite extensive. Again, you can choose to add items to your Shopping Cart or to your Wish List.

Facebook [7] is based on lists. Users maintain a list of their friends and interests and the system tracks changes and additions to these lists, sharing these changes with the user's friends through what it calls a "news feed". This way, users can track what their friends are doing – what is new in their lives, what they like and so on.

The Woolworths online shopping store [29] allows users to maintain lists of products that they may want to buy regularly. The aim is to save users the tedious task of having to add the same items to their "shopping bag" every time. Instead, you can simply select that list and the items are added for you, and you can continue to add other items to the bag before checking out.

Providing a "news feed"-type feature in Cellphone Shopper would benefit users. It could inform them of recent changes to the list and other news. A regularly-bought-items list would also be appropriate and beneficial to users to save them time when they are compiling their shopping lists.

A product recommendation feature may be implemented in Cellphone Shopper if there is sufficient time to do so.

### 3. ASYNCHRONOUS JAVASCRIPT AND XML (AJAX)

AJAX is not a technology; it is, in fact, several technologies that have been grouped together to form an approach to Web application development. As Garrett [10] explains, it incorporates:

- standards-based presentation using XHTML and CSS;
- dynamic display and interaction using the Document Object Model;
- data interchange and manipulation using XML and XSLT;
- asynchronous data retrieval using XMLHttpRequest;
- and JavaScript binding everything together.

Classic Web applications are based on a multi-page interface model [10, 21]. In this model (Figure 1A), a user action on a Web page triggers an HTTP request, which is sent back to the server. The server performs some actions based on this request and then returns a Web page to the client, which replaces the original page with the one it was sent.

AJAX uses a different model (Figure 1B), namely that of a single-page interface [10, 21]. When the user triggers an event that leads to communication with the server, rather than re-loading the entire page to display the results of that interaction, changes are made to elements *within* the page. In this way the user's interaction with the Web application is asynchronous – it is independent of communication with the server.

The core reason for AJAX coming into being is interactivity, which is closely associated with usability [8]. Teo et al. [26] investigated the effects of the level of

interactivity of a Web site on a user's attitude towards the site. Their results suggest that an increased level of interactivity has positive effects on a user's perceived satisfaction, effectiveness, efficiency, value and overall attitude towards a Web site.

Directly connected to interactivity is responsiveness, in particular, user-perceived latency. User-perceived latency is the period between the moment a user issues a request and the first indication of a response from the system [21]. It is desirable for a Web application to be responsive.

There are two primary ways in which user-perceived latency can be decreased: (1) decrease the round-trip time and (2) allow the user to interact with the system asynchronously. AJAX does both.

The core part of AJAX that allows communication without the need for the Web page to be completely refreshed is the XMLHttpRequest element [25]. It allows for a *delta-communication* style of interaction [21], where only state changes are communicated between the client and server, rather than HTTP requests with full-page responses. Hence the round-trip time of client-server interaction is decreased.

As mentioned above, the user's interaction with an AJAX Web application is asynchronous: the client browser can make requests to the server without making the user wait before they can interact with the application again. In addition to this, the use of JavaScript, in particular, means that many of the application's responses can be handled on the client side without involving the server – for example, form validation and various forms of data.

There are two issues worth noting with AJAX [21]. The first is that the initial download of the AJAX code, often referred to as the *engine*, introduces some latency for the user. However, data transfers (while using the system) are smaller and compensate for this. Second, it is possible to burden the client by over-using client-side functionality.

### 3.1 Example Web sites

There are many Web sites that make use of AJAX. The following examples illustrate the wide variety of uses for it.

**Google Suggest** [14] – as you type into the search box, the system offers suggestions in real-time on how to complete what you are typing. Many other Google sites make use of AJAX, including Google Maps [13] (its pan and zoom functions adjust your view using AJAX), Google Mail [12] (which provides, among other things, real-time chat using AJAX) and Google Calendar [11] (which uses AJAX extensively – for example, to provide the user with the ability to drag events from one timeslot to another).

**Panic Goods** [23] – this Web site sells t-shirts. You select the t-shirts you like by dragging them to your shopping cart at the bottom of the page.

**LiveMarks** [20] – this site allows you to watch bookmarks in real-time as they are added to del.icio.us, a

social bookmark Web site where users can maintain their bookmarks and share them with others.

**AjaxTrans** [1] – this system translates from one language to another as you type, sentence by sentence.

### 3.2 Frameworks

Web development is a complex task. Developers have to deal with issues such as content presentation, loading times, aesthetics, navigation, interactivity and security [5]. One of the biggest problems is browser compatibility, which affects, among other things, the way CSS displays and JavaScript operates. Quite often, developers have to use non-standard tricks – “hacks” – to get around this issue, or they simply have to accept the differences.

Frameworks aim to simplify the task of development. More specifically, the goals of these frameworks are to [21]:

- hide the complexity of developing AJAX applications – which is a tedious, difficult, and error-prone task,
- hide the incompatibilities between different Web browsers and platforms,
- hide the client/server communication complexities, and
- achieve rich interactivity and portability for end users, and ease of development for developers.

To achieve these goals, the frameworks provide a library of user interface components and a development environment to create re-usable custom components.

There are numerous AJAX frameworks: AJAX Patterns [9] lists more than 200 of them at the time of writing. Because there are so many, only a small selection will be looked at here. Specifically, the following frameworks listed in [28, 27]: Direct Web Reporting, Dojo, Google Web Toolkit, Microsoft Atlas, Open Rico and Prototype, Prototype and Scriptaculous, Yahoo! User Interface Library and Zimbra's Kabuki AJAX Toolkit.

A framework that is pure JavaScript may be best for the project, so Direct Web Reporting and Google Web Toolkit, which are both Java-based, are not appropriate. Microsoft Atlas is also not appropriate because it is deeply integrated with Microsoft's .NET environment and Cellphone Shopper will not be run in a solely Microsoft environment.

While all the remaining frameworks are adequate, the Yahoo! User Interface Library may be the best choice. Dojo has a broad collection of widgets, but its documentation has gaps in it [28]. Zimbra's collection of widgets is basic and its documentation is also lacking. The combination of Open Rico and Prototype lacks some pragmatic tools, such as a tree. Scriptaculous also lacks complete documentation. The Yahoo! Library has good documentation (that contains many examples and a great deal of code), contains many of the standard tools (such as an animation library and a tree collection) and is easy to work with [28].

It is possible to use multiple libraries, either in part or in whole. So should the chosen framework lack something that another has, it can be augmented.

AJAX is an appropriate approach to take for creating the Cellphone Shopper Web interface. It offers interactivity and speed, two elements that are important for a service that will be used on a regular basis. The availability of various GUI “widgets” in the Yahoo! framework will hopefully facilitate the building of an interface that is easy for the novice user and quick for the experienced user.

Linked to the speed of an AJAX approach is the efficiency of the underlying communication protocol. This will be examined in the next section.

#### 4. EFFICIENT COMMUNICATION

Efficient communication is important for a Web application, particularly an AJAX one. In this section, four light-weight communication protocols will be looked at: XML-RPC, SOAP, REST and JSON-RPC.

XML-RPC is remote procedure calling using HTTP as the transport and XML as the encoding [31]. It is designed to be simple while allowing complex data structures to be transmitted.

SOAP is intended for exchanging structured information in a decentralized, distributed environment [15]. Its key aims are simplicity and extensibility. SOAP messages are encoded in XML and can be transmitted over a variety of underlying transport protocols.

SOAP is the successor to XML-RPC. It works with objects rather than remote procedure calls, but has a greater overhead than XML-RPC [19].

Both XML-RPC and SOAP are widely supported, having been implemented in many languages for many operating systems and environments, both open source and commercial.

Representational State Transfer (REST) is an architectural style, not a protocol. However, it is very often referred to in protocol discussions [32, 22, 24]. In this context, it is used loosely and refers to the sending of data over HTTP without the use of a particular message encoding (you can use any you please, including an XML-based format) [16].

JSON-RPC is an RPC based on JavaScript Object Notation (JSON), which is used to encode its messages [3]. HTTP is used to transmit the messages. JSON is data-orientated, is based on a subset of JavaScript and is easy for humans to read and write [17]. It is also simpler and smaller than XML [18].

JSON is (a) a native format for JavaScript, which AJAX uses, and (b) smaller than XML. However, it is not as widely supported as XML. While this lack of support is not a problem for a Web interface, particularly one based on AJAX, it is a problem for other interfaces, such as a cellphone. XML has far wider support.

The online discussion on XML-RPC vs SOAP vs REST has been a long-running one that appears to have no definitive winner [32, 22, 24]. It appears that SOAP is somewhat verbose, but XML-RPC may be no better. Some people argue that XML-RPC has widespread support because it has existed for a number of years, but SOAP appears to be just as widely supported. Many think that a REST approach is enough (using XML, or maybe even SOAP, for encoding the message).

With no definitive best-choice, it is up to the developers of a particular project to decide what they want to implement. It is possible to support multiple protocols, so it is not the case that one has to be chosen over all the others.

#### 5. CONCLUSIONS

This paper looked at the techniques and technologies that will be used in developing the Web end-user interface for the Cellphone Shopper system.

Studying some shopping systems and Web sites that maintain lists provides ideas for features that could be implemented in the project to aid and inform users. In particular, the system will include some sort of “news feed” and a list of regularly bought items.

The Web development technique known as AJAX was considered. In particular, its core aims of interactivity and speed, both of which are important to Cellphone Shopper, were discussed. A number of Web sites that use AJAX were mentioned and a selection of AJAX frameworks were studied, with the Yahoo! User Interface Library being chosen for use in the project.

Linked to AJAX is the topic of efficient communication. Four protocols were examined: XML-RPC, SOAP, REST and JSON-RPC. No decision has been made as to which one should be used due to the fact that none of the protocols have been definitively defined as being better than the others. It is noted, however, that it is possible to support all of them.

#### 6. REFERENCES

- [1] AjaxTrans, <http://www.ajaxtrans.com/>.
- [2] Amazon.com, <http://www.amazon.com/>.
- [3] Aziz, A., and Kollhof, J. 2006. JSON-RPC 1.1 Specification. Working draft. August. Web site: <http://json-rpc.org>.
- [4] Bellamy, R., Brezin, J., Kellogg, W.A., Richards, J. and Swart, C. Designing an E-Grocery Application for a Palm Computer: Usability and Interface Issues. *IEEE Personal Communications*, 8, 4, 60-64, August 2000.
- [5] Critical Issues for Web site Development. Retrieved 8 July 2007 from Net Access:

- [http://www.netxs.com.pk/web/critical\\_issue\\_for\\_web\\_develop.html](http://www.netxs.com.pk/web/critical_issue_for_web_develop.html).
- [6] del.icio.us, <http://del.icio.us/>
- [7] Facebook, <http://www.facebook.com/>
- [8] Folmer, E. 2005. *Software Architecture analysis of Usability*. PhD thesis, University of Groningen, Mathematics and Computer Science.
- [9] Frameworks. Retrieved 8 July 2007 from Ajax Patterns: <http://www.ajaxpatterns.org/Frameworks>.
- [10] Garrett, J. J. AJAX: A new approach to web applications. 2005. Retrieved 8 July 2007 from Adaptive Path: <http://www.adaptivepath.com/publications/essays/archives/000385.php>.
- [11] Google Calendar, <http://calendar.google.com/>.
- [12] Google Mail, <http://www.gmail.com/>.
- [13] Google Maps, <http://maps.google.com/>.
- [14] Google Suggest, <http://www.google.com/webhp?complete=1&hl=en>.
- [15] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., Nielsen, H., Karmarkar, A., and Lafon, Y. 2007. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). W3C Recommendation. April. World Wide Web Consortium (W3C). Web site: [www.w3.org](http://www.w3.org).
- [16] Hixie, I. Call an apple an apple. 2005. Retrieved 10 July 2007 from Hixie's Natural Log: <http://ln.hixie.ch/?start=1111339822>.
- [17] Introducing JSON. Retrieved 10 July 2007: <http://json.org/>.
- [18] JSON: The Fat-Free Alternative to XML. Retrieved 10 July 2007: <http://www.json.org/xml.html>.
- [19] Lerner, R. M. 2001. At the Forge: Introducing SOAP. *Linux Journal*. 2001, 83es (Mar. 2001), 11.
- [20] LiveMarks, <http://sandbox.sourcelabs.com/livemarks/>.
- [21] Mesbah, A. and van Deursen, A. 2007. An Architectural Style for Ajax. In *Proceedings of the Sixth Working IEEE/IFIP Conference on Software Architecture* (January 06 - 09, 2007). WICSA. IEEE Computer Society, Washington, DC, 9.
- [22] Morin, R. SOAP, REST and XML-RPC. 2006. Retrieved 10 July 2007 from The RSS Blog: <http://www.kbcafe.com/rss/?guid=20060704042846>.
- [23] Panic Goods, <http://www.panic.com/goods/>.
- [24] Ruby, S. 2003. XML-RPC, SOAP, and/or REST. Retrieved 10 July 2007 from Intertwingly: <http://intertwingly.net/blog/1507.html>.
- [25] Stamey, J. and Richardson, T. 2006. Middleware development with AJAX. *Journal of Computing Sciences in Colleges*. 22, 2 (Dec. 2006), 281-287.
- [26] Teo, H., Oh, L., Liu, C., and Wei, K. 2003. An empirical study of the effects of interactivity on web user attitude. *International Journal of Human-Computer Studies*. 58, 3 (Mar. 2003), 281-305.
- [27] Turner, A., and Wang, C. AJAX: Selecting the Framework that Fits. 2007. Retrieved 9 July 2007 from Dr. Dobb's Portal: <http://www.ddj.com/dept/webservices/199203087?pgno=1>.
- [28] Wayner, P. Surveying open-source AJAX toolkits. 2006. Retrieved 9 July 2007 from InfoWorld: [http://www.infoworld.com/article/06/07/31/31FEajax\\_1.html](http://www.infoworld.com/article/06/07/31/31FEajax_1.html).
- [29] Woolworths, [www.woolworths.co.za](http://www.woolworths.co.za).
- [30] Wu, H. and Natchetoi, Y. 2007. Mobile shopping assistant: integration of mobile applications and web services. In *Proceedings of the 16th international Conference on World Wide Web* (Banff, Alberta, Canada, May 08 - 12, 2007). WWW '07. ACM Press, New York, NY, 1259-1260.
- [31] XML-RPC Home Page. 2003. Retrieved 10 July 2007: <http://www.xmlrpc.com/>.
- [32] XmlRpcDiscussion. 2006. Retrieved 10 July 2007 from Intertwingly: <http://intertwingly.net/wiki/pie/XmlRpcDiscussion>.