The Web 2.0 Movement: MashUps Driven and Web Services

Chang-Chun Tsai, Associate Professor, Institute of Management, MingDao University
MingDao University, No. 369, Wen-Hua Rd., Peetow,ChangHua 52345, Taiwan, R.O.C.
Email: jimtsai@mdu.edu.tw

Cheng-Jung Lee, Ph. D. Candidate of Department of Information Management, National Yunlin University of Science and Technology, Taiwan R.O.C.
Email: rabbit66@ms61.url.com.tw

* Shung-Ming Tang, Professor, Department of Information Management,
National Yunlin University of Science and Technology, Taiwan R.O.C.
Email: tangsm@yuntech.edu.tw

Abstract: - Service-Oriented Architecture (SOA) can be viewed as a philosophy that drives the development of components by defining their interfaces clearly and in a way that relates to real needs. It is the key to IT and business flexibility and receives a lot of attention from academia and industry as a means to develop flexible and dynamic software solutions. Web 2.0 world is wide and rich. Although significant progress is being made in several fronts, many other researchers speak of Web 2.0 applications; they tend to focus on the technology aspects of the environment. However, the real impact of integrating Web 2.0 technologies is to tie the flexibility of Web 2.0 to service-oriented principles of loose coupling, encapsulation, and reuse that are the heart and soul of SOA. Today's sites are no longer limited to exchanging links and interacting via hypertext; instead, the interconnectedness of the Internet has become progressively more important with the rise of web services. This paper presents web 2.0 mashups remixing data and Web Services. The purpose of this paper is to propose a flexibility solution and new mashup platforms for e-applications and interactive services. Flexibility is the key driver of Web 2.0 success—the flexible delivery of data through the combination of services and disparate data sources through mash-ups, real-time data feeds, and rich interactions. In this paper we also explore the architectural basis, technologies, frameworks and tools considered necessary to face this novel vision of Web 2.0—all of which adds business value and helps companies utilizes the rich collaboration and communication of the Internet today.

Key-Words: - SOA, Web services, marhups, web 2.0, Service-Oriented, platform, blogs

1 Introduction

1.1 The trend toward networked applications

Researchers commonly referred to Web 1.0 as Electronic Commerce or E-Business. Whereas, web 1.0 focused on a read only web interface, Web 2.0 focuses on a read-write interface where value emerges from the contribution of a large volume of users. The Internet initially focused on the command and control of the information itself. Information was controlled by a relative small number of resources but distributed to a large number which spawned the massive growth of the web itself. [24]

In recent years, the entrepreneurial spirit of the Internet has been rekindled, and smart companies are taking small, gradual steps into new and unconventional business directions. Web developers have begun to use dynamic technologies and concepts like Ajax [16], Flash, PHP and content syndication in order to build more collaborative and rapid-publishing applications. [32] More and more sites are exposing their data through service oriented architecture (SOA), Web services and XML.

The Internet is now a hub for commerce and information. The technology necessary to produce complex behavior in web sites has also gotten increasingly complex. [33] The fruit of this innovation is the wealth of blogs, free APIs, comment-driven sites and wikis you enjoy on the web today. The role of a web site has evolved from the simple act of presenting a user with a view of information, to a Rich Internet Application (RIA) that can provide the user with a highly interactive experience. Applications woven into the Web are no longer software — they are services that people take for granted. This is a better thing in the RIA world than it is in the relationship world, because, in the case of your well-written RIA, it may get you the kind of notice that changes your career and lifestyle. Just ask the creators of YouTube what being taken for granted is worth (in their case it was over $1.6 billion). New classes of applications, such as Base Camp, Flickr, YouTube, and Picasa are useful specifically because of their networked context. The trend toward networked applications is accelerating. [18]

1.2 What is Web 2.0

Web 2.0 provides the basic technology for creating a network of customers who are passionate about the company’s product offering. There are many different opinions on this, making it difficult to pinpoint an exact definition; however, some of the features typically associated with Web 2.0 sites are as follows:

(1) **Using standards-compliant HTML and CSS**. This allows sites to work across many platforms and helps with accessibility. This includes the use of microformats to generate friendly HTML that can be used across a variety of platforms.

(2) **Using Ajax to provide a rich user interface.** By performing trivial operations in the background using XMLHttpRequest, web pages can be more functional and intuitive. XMLHttpRequest is a JavaScript API that allows a background HTTP request to occur while a user is viewing a web page. This means that the current page can be updated based on a response from the server without the user navigating to another page on the web site. The phrase “making an Ajax request” (or similar) typically refers to performing an HTTP request in the background using XMLHttpRequest.

(3) **Sharing data using web feeds and web services**. Users like to aggregate many feeds to easily receive content updates from their favorite sites using web feeds (such as RSS or Atom). Additionally, web services can enable one site to use data from other sites (for instance, using Google Maps).

(4) **Incorporating social networking tools.** Blogs and forums can enable users to communicate with each other. [4]

1.3 The popularity of Web 2.0

Positive network effects created the Web 2.0 network platforms and contributed to the online hyper growth of networks such as Google, Yahoo!, eBay, Skype, Wikipedia, Craigslist, Flickers, and others. [19] The Web 2.0 platform has emerged as a model and standard for contemporary web applications [1, 20, 26], supported by the so-called Ajax technologies [6, 11].

Facebook, YouTube, Skype, MySpace, and Flickr show that a Web 2.0 company's business and financial valuation depends on the number of users and how quickly those users accept, adopt, and bring their positive network effects to a new online service. These enterprises have strategically combined different kinds of network effects—including direct, indirect, cross-network, and demand-side—to multiply the overall positive impact of network value creation. Web 2.0 turbocharges network effects because online users are no longer limited by how many things they can find, see, or download off the Web, but rather by how many things they can do, interact, combine, remix, upload, change, and customize for themselves. This online DIY self-expression benefits businesses and other users, not just individual uploaders. [19, 23]

Web 2.0 changes the rules of business, but it isn't a simple disintermediation play aimed at replacing earlier businesses with web-based businesses. New-style click-and-mortar [13], online-offline network partnerships focus on bridging and building new networks rather than replacing or disrupting the infrastructures of offline companies. There's plenty of opportunity for connecting innovations in the online business worlds, and potential competitors are also potential partners. There are many opportunities for Web 2.0-style online-offline cooperation besides competition in your relationships with customers, suppliers, competitors, and complementors. [19]

1.4 Importance

According to Forrester Research Inc. investments in Web 2.0 technologies will increase by approximately 43 percent per annum for the next 5 years [10]. It is predicted that in 2013 companies will invest 4.6 billion dollars in the Web 2.0 domain. Forrester further specifies that social networks, blogs, mishaps, RSS, podcasting, widgets and wikis will become primary applications. The “ARD/ZDF Online Studie 2006“, a study on online usage, states that 11 percent of all online people in Germany use a Web 2.0 application at least once a week and 9 percent on a daily basis. Additionally Web 2.0 applications are mainly used by people with higher education, e.g. “Abitur” certificate. It is also important to remark that 43 percent of Web 2.0 usage is passive and 57 percent active. This means that one of the main aspects of Web 2.0, the interactivity, is perceived at a high rate. According to an empirical study of Web use by Weinreich et al. [31] a strong growth in the proportion of submits events indicate an increase in dynamic Web pages and Web applications. In summation, Web 2.0 applications are already important but are predicted to increase in usage even further during the next few years. [34]

How can Web 2.0 redefine and transform the scope of business model innovation and collaboration in your industry, and across industries and geographies successfully? The communication mechanisms vary, ranging from podcasts, wikis, and feeds to social networking. Enabling Web 2.0 functionality requires increased infrastructure flexibility, which can be supported by a robust service oriented architecture (SOA) [8]. SOA is an approach that draws IT and Business together and drives a discipline toward flexibility. Flexibility is the key driver of Web 2.0 success—the flexible delivery of data through the combination of services and disparate data sources through mash-ups [30], real-time data feeds, and rich interactions.

SOA and Web 2.0 have received a lot of press in recent times [7, 9, 17, 14, 29], as Web 2.0 facilitates the collaboration aspects, and SOA enables the infrastructure for flexibility, however what about the linkage between Web 2.0 and SOA? Web 2.0 applications have much potential for the public sector in terms of interaction, participation and transparency. However, examples of website with MashUps remixing data and Web Services are rare. In this paper, we explore the linkage between Web 2.0 and SOA which is important for businesses. Mash-ups represent the practical bridge between SOA and Web 2.0 and represent the coming together of information from disparate sources, SOA infrastructure provides those information sources. Scalable, dynamic, and accessible over the Web, SOA services are the raw material for Web 2.0 mash-ups. Both AJAX and REST (Representational State Transfer) are enablers for SOA. REST can be used to expose services, and AJAX is used to build front ends. The purpose of this paper is to propose a flexibility solution and new mashup platforms for e-applications and interactive services. Companies can take this approach to extend their existing infrastructure and management capabilities in support of those projects.

2 The Web 2.0 Movement

Web 2.0 has given rise to a wave of new tools and techniques, below are innovative technologies proliferate around Web 2.0.

2.1 Web 2.0 Mashups

Web 2.0 is composed of many enabling technologies, such as PHP, AJAX, RSS, REST, and others. These Web 2.0 composite applications are often referred to as mash-ups. A mashup is a more user-facing and user-directed process, where an end user directs an application to combine and aggregate data from various sources (RSS feeds, Atom feeds, web pages, etc.) and combines the data with a presentation to provide a comprehensive view of the aggregation. Such a mashup provides great value to the end user. As a kind of ad-hoc composition, several data outputs of existing services or API’s can be combined in a new representation that is also called mashup (See figure 1). The main sources of mashups can be categorized as mapping, photo, video, news, weather, search, and e-commerce. Typically, data sources such as REST- or SOAP (Simple Object Access Protocol)-based services as well as lightweight protocols like RSS or ATOM build the foundation of a mashup. The user interaction and the rendering of the application is usually done within the browser. [28]

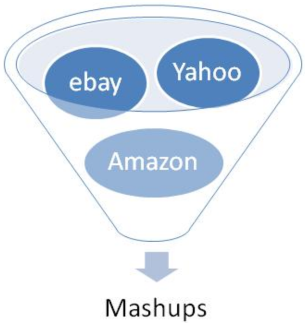

Figure 1: Mashups can be created using APIs from competing companies

A mashup usually involves two or more web applications or parts of web applications combined to make something new, just like with The Grey Album. The end result is a hybrid web application. [35] In many cases, this facilitation is made possible by the data owners providing application programming interfaces (API) to their data. These APIs follow standard web service protocols and can be implemented quickly and easily in a variety of programming languages, including PHP. [21] For example, Flickr is an up-and-coming photo-sharing service that is often considered a close sibling of del.icio.us. Many of the features of del.icio.us are directly replicated in Flickr, but with the purpose of organizing photos instead of bookmarks. For example, like del.icio.us, Flickr features tags as organizational tools. It also has RSS feeds for almost every page, and of course an API.

The emergence of new mashup platforms (e.g. Yahoo Pipes, Microsoft Popfly, IBM QedWiki) to aggregate and visualize heterogeneous web resources shows their potential as one key technology of the Web 2.0. [15] Before some of these technologies are explained, it is important to further categorize them into the roles they play in mashup development, which is done in Table 1 [25]

Table 1: Categorized technologies used by mashups

Foundation	HTTP	Web browser		
Presentation	CSS	HTML /XHTML		
Interactivity	JavaScript	Ajax		
WebServices API	XMLHTTP Request	XML-RPC	SOAP	REST
Data	XML	RSS /Atom	JSON	KML

2.2 Ajax: Asynchronous JavaScript and XML

Ajax is a JavaScript development technique that is so important to Web 2.0 development that it should be considered a separate technology. The name Ajax came from the bundling of its enabling technologies: an asynchronous communication channel between the browser and server, JavaScript, and XML. AJAX can be a useful tool in improving web application usability. It's spawning a new breed of web applications that can expand the possibilities of what users can accomplish inside a web browser. AJAX is not only improving upon stale and archaic web architectures, but it also enables web-based applications to rival or surpass the importance of desktop applications in terms of usability and user experience. AJAX even allows powerful new application workflows and visualizations that currently have no desktop software-based equivalent—not necessarily because of a technological shortfall on the part of desktop developers but certainly because AJAX has put Rich Internet Applications (RIA) within reach of most web developers. From that perspective, AJAX has already changed and will continue to change the way users view traditional web and desktop applications alike.[31]

JavaScript is the programming language Ajax function calls are made in. Data retrieved using the technique is commonly formatted using XML and JSON. Quoting the original definition, Ajax incorporates the following features: [2, 27]

(1) Standards-based presentation using Extensible HyperText Markup Language (XHTML) and Cascading Style Sheets (CSS)
(2) Dynamic display and interaction using the browser's Document Object Model (DOM)
(3) Data interchange and manipulation using Extensible Markup Language (XML) and Extensible Stylesheet Language Transformations (XSLT)
(4) Asynchronous data retrieval using XMLHttpRequest or XMLHTTP (from Microsoft)
(5) JavaScript to bind everything together

AJAX is a web development technique that allows you to directly connect your HTML presentation with XML data. This XML data can come from anywhere, such as a PHP script or web service, and can be processed via a JavaScript command or routinely by your web page, without the need for a form submit or page reload. [12, 35]

You can think of the benefits in at least three ways:
(1)AJAX can improve and empower the user experience for end users, making them more effective and satisfied.
(2) AJAX can reduce the demands on network and server infrastructure, saving money by reducing maintenance and even bandwidth, and improve quality of service for all users.
(3) AJAX can create the possibility for new kinds of functionality not possible or practical in a traditional application model, giving users new tools to achieve their goals. [22]

2.3 XMLHttpRequest

Before we go into the various types of web services, it is important to call out the most important component of the emergence of web services as a way to create web APIs— XMLHttpRequest. XMLHttpRequest is not a web service technology but an API that is available in JavaScript, to send data to and from a web server using HTTP, by establishing an independent communication channel between a web page's client side and server side. The data returned from XMLHttpRequest calls are often provided by back-end databases. Besides XML, XMLHttpRequest can be used to fetch data in other formats such as HTML, JSON, or plain text. [25]

2.4 RESTful architectures

In RESTful architectures, the method information goes into the HTTP method. In Resource-Oriented Architectures, the scoping information goes into the URI. The combination is powerful.

A few well-known examples of RESTful, resource-oriented web services include:
(1) Services that expose the Atom Publishing Protocol (http://www.ietf.org/html.charters/atompub-charter.html) and its variants such as GData (http://code.google.com/apis/gdata)
(2) Amazon's Simple Storage Service (S3) (http://aws.amazon.com/s3)
(3) Most of Yahoo!'s web services (http://developer.yahoo.com/)
(3) Most other read-only web services that don't use SOAP

2.5 Web Applications as Web Services

Along with syndication, web services are the primary means of integrating web applications. Whereas syndication, especially RSS, provides an outgoing read-only information stream, web services allow other programs to interact by submitting data as well as requesting data. There are two types of web services: SOAP and REST.

SOAP Web Service:
This style of web service is a form of RPC (Remote Procedure Call), where both the request and response of the call is XML. Although the calls are primarily transmitted over HTTP, the transport can be JMS, SMTP, or even FTP. SOAP-style web services use WSDL (Web Service Description Language) to define the API, may use UDDI (Universal Description Discovery and Integration) registries to find services, and can use SYNDICATION AND INTEGRATION a number of other W3C standards for application-level cross-cutting concerns. All of this provides a complete and robust infrastructure for web services.

.

RESTful Web Service:
REST is an architectural guideline meant for the World Wide Web, Many websites follow this architectural style fairly closely. Almost all of them follow at least a few of REST ideas and principals. [35] RESTful web services are lightweight, primarily use HTTP, and focus around the concept of using URIs (Uniform Resource Identifiers) to identify integration points. To modify or access the state of the object, the HTTP methods POST, GET, PUT, and DELETE are used. REST is an architectural style rather than a specific implementation. [3]

2.6 Remixing data and services

Three MashUps examples are explored below to see how people are remixing data and services to make something new and useful, three of them here: [5]

(1). Housingmaps.com takes two already well-known web applications to create something new: Housingmaps.com brings together Google Maps and the housing and rental listings from Craigslist.com.

Where Is the Remixing Happening?
The remixing occurs on the server side on a web site (Housingmaps.com) that is distinct from both the source web site (Craigslist) and the destination application (Google Maps). Data is drawn from the source and transformed into a Google map, which is embedded in web pages at Housingmaps.com. Since the debut of Housingmaps.com, many other mashups have followed this pattern set of recasting data to make geographical location the organizing principle.

Comparable Mashups:
Many other mashups involve extracting geocoded data (location information) from one source to then place it on an online map (such as a Google map or Yahoo! map). Two prominent examples are listed below: 1) Adrian Holovaty's Chicago crime map, which is a database of crimes reported in Chicago fronted by a Google Map interface. 2) Weather Bonk, which is a mashup of weather data on a Google map

(2). Google Maps in Flickr (GMiF) brings together Flickr pictures, Google Maps, Google Earth, and the Firefox browser via Greasemonkey. The script was created to fill in that gap by letting you see a Flickr photo on a Google map. Today, even with Flickr's built-in map of geotagged photos, which uses Yahoo! Maps technology, GMiF remains a valuable mashup. GMiF allows users to use a Google map, which many prefer over Yahoo! Maps, to display their photos. Moreover, GMiF also integrates Google Earth, a feature not currently built into Flickr. GMiF provides an excellent case study of how you can extend an application such as Flickr to fit user preferences.

Comparable Mashups:

Mappr, “an interactive environment for exploring place based on the photos people take,” is a mashup of Flickr and a Flash-based map.

What Is Being Combined?
GMiF brings together Flickr pictures, Google Maps, and Google Earth within the Firefox browser via a Greasemonkey script. As the Greasemonkey scripts are persistent, the changes made to the web pages are executed every time the page is opened, making them effectively permanent for the user running the script.” Greasemonkey scripts allow you—as the user of that web site and not as the author of the web site—to make customizations, all within the web browser.

(3). The Library Lookup bookmark: This is a JavaScript bookmarklet that connects Amazon.com and your local library catalog. It makes it easy to jump from the Amazon page to the corresponding catalog entry in your local library catalog. It shows another way to create browser-based integration. The user needs to do the following: 1) Configure a bookmarklet for the library of your choice. 2) Invoke that bookmarklet when you arrive on a web page for the book you want to look up in your library.

Comparable Mashups:
BookBurro, in the form of either a Firefox extension or a Greasemonkey script, displays the price of a corresponding book as a pop-up window.
LibraryThing is “an online service to help people catalog their books easily.” It is much more than a typical mashup but has elements that are mashup-like

2.7 Remixing Feeds

1. Remixing Feeds with Feedburner
Feedburner (http://feedburner.com) lets users remix feeds and offers intermediary services based on feeds (such as tracking usage and advertising). It thus provides a useful illustration of the ways some users and companies are reusing and repackaging feeds.

2. Remixing Feeds with Yahoo! Pipes
Yahoo! Pipes is a “an interactive data aggregator and manipulator that lets you mash up your favorite online data sources.” It is focused on enabling end users to filter and combine feeds into new feeds. You construct pipes through modules, entering parameters, and describing data flows through wiring these widgets together. Yahoo! Pipes is arguably more accessible to nonprogrammers because it does not involve typing code in a text editor.

2.8 Web services API: Amazon S3

The only interface to S3 provided by Amazon.com is a web services API. You can access S3 through its REST or SOAP interface directly or via third-party language specific API kits that use the REST or SOAP interface. The utility model lowers the barrier of entry to the level that a relatively poor individual can afford to create a Web 2.0 application. Besides, S3 is potentially cheaper than the alternative solutions.

3 Our Approach

3.1 The Proposed business model

Higher-level Web 2.0 is a cocktail of various new technology vectors. This is to reshape your business by using mashups through SOA combined with Web2.0 technologies. These technology vectors have given a fresh impetus to next-generation applications. Technology vectors can be divided in the following categories as shown in Figure 2.

Protocols :

Protocols such as SOAP, XML Remote Procedure Call (XML-RPC), REST are emerging technology vectors for these next-generation applications. Web 2.0 applications communicate with third-party Web Services and need XML envelopes running over traditional HTTP/HTTPS. Browsers are powered to access third domain applications using different calls.

Application Environment:

SOA is one of the key elements in the overall architecture. SOA provides various sets of Web services that can be consumed by the target browser or any other application.

Linkage:

The linkage between Web 2.0 and SOA is Mash-ups Mash-ups represent the practical bridge between SOA and Web 2.0 and represent the coming together of information from disparate sources, SOA infrastructure provides those information sources. Scalable, dynamic, and accessible over the Web, SOA services are the raw material for Web 2.0 mash-ups. Both AJAX and REST are enablers for SOA. REST can be used to expose services, and AJAX is used to build front ends. This is a flexibility solution and new mashup platforms for e-applications and interactive services— all of which adds business value and helps companies utilizes the rich collaboration and communication of the Internet today.

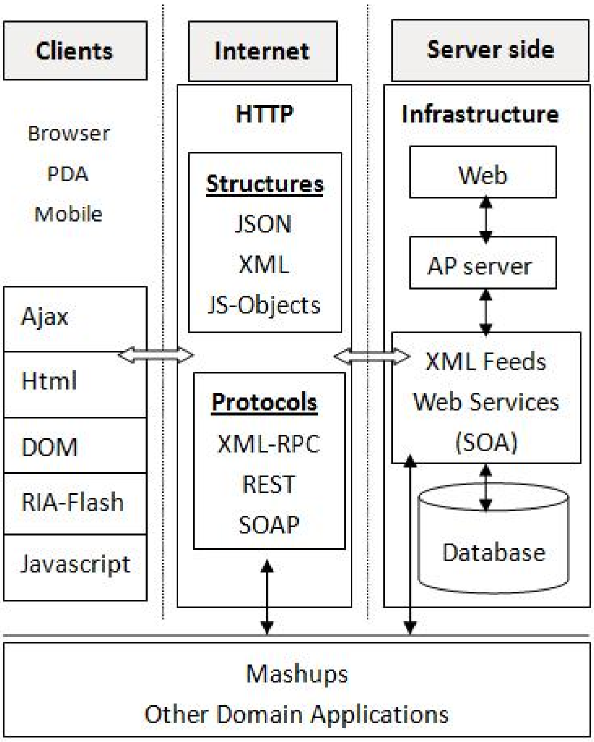

Figure 2: Web 2.0 higher-level view

The development of mashups follow the new principle of user-driven application design and bring flexibility and speed in delivering new valuable services to consumers. The developer can reuse existing resources and combine them to deliver value-added services or applications in a new context. Below is locus of value creation. (See Figure 3)

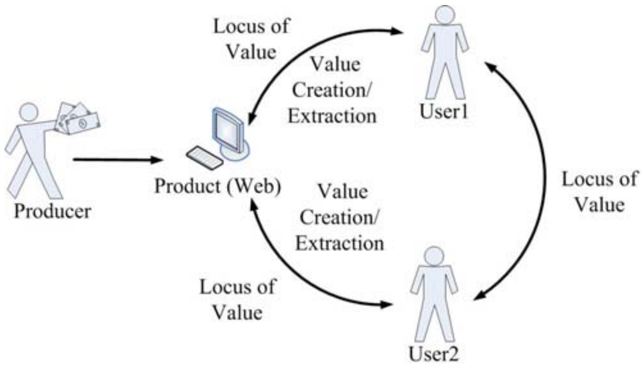

Figure 3: The Web 2.0 Value Creation System

3.2 Patterns in Mashups :

The pattern we present is the combination of four (1) Provide a meaningful context for creation. (2) Data is extracted from a source web site. (3) This data is translated into a form meaningful to the destination web site. (4) The repackaged data is sent to the destination site.

Mashups combine content from more than one source into a new integrated whole. Users can be invited to participate; they create a coherent, consistent context (see Figure 4). At a high level, this is community context; at a framework level, it includes mechanisms for identity, reputation, relationships among users, data identification (URLs, tags, etc.), and find ability of user and data search, aggregation, and personalization.

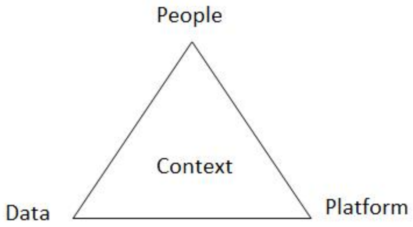

Figure 4: Context for creation

3.3 Ajax Interaction

We use Ajax (specifically, the XMLHttpRequest object) to request data asynchronously and then dynamically update a web page with the requested data. The XMLHttpRequest object is the basics of working with the heart of Ajax. Figure 5 shows the interaction paradigm in an Ajax application.

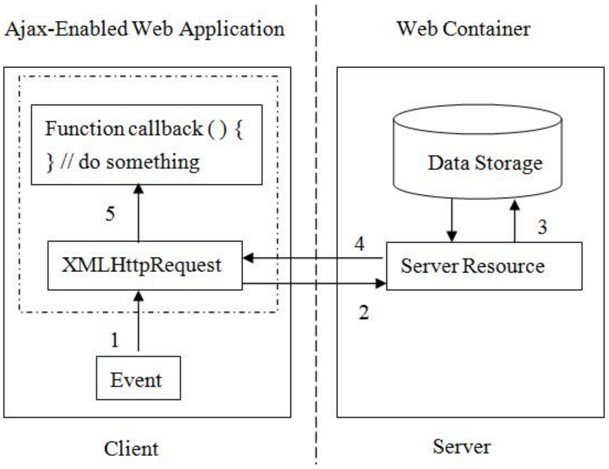

Figure 5: The interaction paradigm in an Ajax application.

The adoption of the XMLHttpRequest object by modern browsers has ushered in a new era of developing rich Web applications. In this paper, we established the basics of working with the heart of

Ajax, the XMLHttpRequest object. Using the XMLHttpRequest Object, an Ajax application does things as follows:

1. A client-side event triggers an Ajax event and then an instance of the XMLHttpRequest object is created.
2. A request is made to the server. This can be a call to a servlet or any server-side technique.
3. The server access a data storage or another system.
4. The request is returned to the browser.
5. Configure the XMLHttpRequest object to call the function callback( ) when the processing returns.

In summation, a user, controlling a browser, makes a request for the main URI of an application. Web 2.0 applications use several protocols over Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS).The server serves a web page that contains an embedded script. The browser renders the web page and either runs the script, or waits for the user to trigger one of the script's actions. The script makes an asynchronous HTTP request to some URI on the server. Most user actions in the interface trigger an HTTP request back to a web server. The server does some processing —retrieving data, crunching numbers, talking to various legacy systems — and then returns an HTML page to the client. It's a model adapted from the Web's original use as a hypertext medium. The script parses the HTTP response and uses the data to modify the user's view. This might mean using DOM methods to change the tag structure of the original HTML page. It might mean modifying what's displayed inside a Flash application or Java applet.

With XMLHttpRequest you are no longer limited to complete page refreshes and synchronous conversations with your server. The server doesn't necessarily need to send the response in XML format. It's perfectly legal to send the response as simple text as long as the Content-Type response header is set to text/plain. Combined with JavaScript and some basic DOM manipulation, Ajax allows for a level of interactivity previously unmatched on the Web. You can combine your existing expertise of server-side technologies with the unique capabilities of XMLHttpRequest to provide highly interactive Web applications. You can use Ajax techniques to send requests to the server and the client can parse the server's response. With AJAX, only the necessary data is transferred back and forth between the client and the web server without having to waste time reloading the entire page again. Ajax increases a web page's interactivity, speed, functionality, and usability. This minimizes the network utilization and processing on the client.

3.4 Roadmap to create Remix:

The following is map to create remix, we (1) use XML to Structure Data (2) use RSS and Atom to Receive Data Automatically (3) use JavaScript to Script the Mashup Page (4) use PHP to Perform Server-Side Scripting (5) use XMLHttpRequest, XML-RPC, REST, and JSON to Retrieve Data (6) use XHTML to Structure the Mashup Pages (7) Use REST and AJAX which are enablers for SOA.

REST can be used to expose services, and AJAX is used to build front ends. AJAX and SOA complement each other. AJAX takes care of the client side user interaction and is agnostic about what happens on the server. On the other hand, SOA encapsulates the server side functionality. AJAX applications can interact with services to fetch data from server and send modified data back to server.

3.5 Mashup Tools for web services:

Many tools are designed to help in creating mashups. We focus on tools for mashing up web services: (1) Data Mashups Online Service: use Browser -based GUI to create custom business applications, especially for mashing up data and web services (2) StrikeIron SOA Express for Excel: An add-on to enable easy access web services from within Microsoft Excel (3) WSO2 Mashup Server: An open source platform for creating and deploying "web services mashups"

4 Conclusion

The paper describes the concept, the flexibility and the technology associated open innovation approaches and business models. Data and content exposed through SOA combined with Web2.0 technologies change the application-development economics. Applications should take advantage of the Web as a platform rather than simply providing a presence on the Web. By working symbiotically with the openness and connectedness of the Web, services can reach out to all users. And in doing so, will get better as more people use the service. We presente the main ideas and patterns behind the concepts of web2.0. Some examples of mashups and web services are presented, the supporting technology is summarized. The mashup pattern, roadmap to create Remix and its key tools were introduced and the explanations of the fragments were provided. This calls for more research that can improve our knowledge on the strategies, business models and relevant technologies involved.

References:

[1] Anderson, C. The Long Tail: Why the Future of Business is Selling Less of More. New York: Hyperion, 2006.

[2] Apress (2006), Christian Gross, *Ajax and REST Recipes: A Problem-Solution Approach*, ISBN-13 (pbk): 978-1-59059-734-7

[3] Apress Inc. (2007), Ian Roughley, *Practical Apache Struts2*, *Web 2.0 Projects,* ISBN-10 (pbk): 1-59059-903-9.

[4] Apress (2008), Quentin Zervaas, Practical Web 2.0 Applications with PHP, ISBN-13 (pbk): 978-1-59059-906-8

[5] Apress Inc.(2008), Raymond Yee, *Pro Web 2.0 Mashups Remixing Data and Web Services*, ISBN-10 (pbk): 1-59059-858-X.

[6] Springer-Verlag, Apress (2007), Babin, L., *Beginning Ajax with PHP: From Novice to Professional*.

[7] C. Schroth and T. Janner. Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services, volume 9. In: IEEE IT Professional, June 2007.

[8] Frank Tsui, Andy Wang, Kai Qian, A discussion on security typing and measurement for SOA, May 2009, SIGSOFT Software Engineering Notes , Volume 34 Issue 3

[9] Federico Kereki, Web 2.0 development with the Google web toolkit, Feb.2009 Linux Journal, Volume 2009 Issue 178

[10] Forrester Research Inc. (Ed., 2008). Forrester: Global Enterprise Web 2.0 Market To Reach $4.6 Billion By 2013.

[11] Garrett, J. J. *Ajax: A New Approach to Web Applications. Adaptive Path.* Feb., 2005.

[12] Hsuan-Che Yang, Yuan-Tsung Chang, Timothy K. Shih, Using AJAX to build an on-line QTI based assessment system, Proceedings of the 2007 WSEAS International Conference on Computer Engineering and Applications, Gold Coast, Australia, January 17-19, 2007

[13] Jiangping Du, A tentative study on the model of the campus e-commerce, 2005, ICEC '05: Proceedings of the 7th international conference on Electronic commerce.

[14] John Vrettaros, Katerina Argri, WEB2.0 Tools in Education, 12th WSEAS International Conference on COMMUNICATIONS, Heraklion, Greece, July 23-25, 2008

[15] J. Zou and C. J. Pavlovski, Towards Accountable Enterprise Mashup Services. IEEE Conference on e-Business Engineering, 2007.

[16] Kanakalata N., U. Banerjee, S. Kumar ,2009, ISEC '09: Proceeding of the 2nd annual conference on India software engineering conference

[17] Ming-Chih Hsieh, Yung-Wei Kao, Sheau-Ling Hsieh, Shyan-Ming Yuan, Web 2.0 Proxy: A New Framework for Web 2.0 Website Development 7th WSEAS Int. Conf. on TELECOMMUNICATIONS and INFORMATICS (TELE-INFO '08), Istanbul, Turkey, May 27-30, 2008

[18] O'Reilly (2006), Tim, *Web 2.0 Principles and Best Practices*, O'Reilly Media Inc., ISBN-10: 0-596-52769-1.

[19] O'Reilly (2008) , Amy Shuen,*Web 2.0: A Strategy Guide*, ISBN-13: 978-0-596-52996-3.

[20] O'Reilly (2005), Tim. *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Softwar*, O'Reilly.

[21] PACKT publishing, Shu-Wai Chow, *PHP Web 2.0 Mashup Projects*, 2007

[22] Prentice Hall (2008), Dave Johnson, Alexei White, Andre Charland, *Enterprise AJAX: Strategies for Building High Performance Web Applications*, ISBN-13: 978-0-13-224206-6

[23] R. van Zwol, Flickr. Who is Looking? 2007,WI'07: Proceedings of the IEEE /WIC/ACM International Conference on Web Intelligence.

[24] Springer Science+Business Media (2008), Miltiadis D. Lytras, Ernesto Damiani, Patricia Ordóñez de Pablos, *Web 2.0 The Business Model*,LLC, ISBN-13: 978-0-387-85894-4

[25] Springer-Verlag New York, Inc., (2008), Eric Griffin, *Foundations of Popfly: Rapid Mashup Development*, Eric Griffin, ISBN-13 (pbk): 978-1-59059-951-8

[26] Tapscott, D., and Williams, A.D. Wikinomics: How Mass Collaboration Changes Everything. New York: Penguin Group, Portfolio, 2006.

[27] The Pragmatic Programmers LLC (2005), Pragmatic Ajax-A Web 2.0 Primer, Justin Gehtland, Ben Galbraith, Dion Almaer, ISBN 0-9766940-8-5

[28] Tobias Nestler, Towards a Mashup-driven End-User Programming of SOA-based Applications, MDC 2008, Proceedings of iiWAS2008, Page 552

[29] T. V. Raman, Toward 2W, beyond web 2.0, Feb.2009 Communications of the ACM, Volume 52 Issue 2

[30] X. Liu, Y. Hui, W. Sun, and H. Liang. Towards Service Composition Based on Mashup. In IEEE Congress on Services, 2007.

[31] Weinreich, H., Obendorf, H., Herder, E., and Mayer, M. (2008). Not quite the average: An empirical study of Web use, in ACM Transactions on the Web, vol.2 , no.1, pp. 1–31.

[32] Wiley Publishing (2007), C. Freedman, *Yahoo! Maps Mashups*, ISBN: 978-0-470-09778-6.

[33] Wiley Publishing (2007), D. Moore, R. Budd, W. Wright, *Python Frameworks Web 2.0 Programming with Django and TurboGears* ISBN-13: 978-0-470-13809-0.

[34] W. Kern, International Journal of Public Information Systems, Vol. 2008:2

[35] Wiley Publishing (2007), B. O'Connor, *del.icio.us Mashups*, ISBN: 978-0-470-09776-2