Mashup Technologies for Building End-user Enabled Business Portal

Shah Jahan Miah [1] Member, IEEE , John Gammack[1], and Geoffrey Greenfield [2]

[1] Institute for Integrated and Intelligent Systems, Griffith University, Brisbane, Australia
e-mail : s.miah@griffith.edu.au, j.gammack@griffith.edu.au
[2] UQ Business School, University of Queensland, Brisbane, Australia
e-mail: g.greenfield@business.uq.edu.au

Abstract— Mashup refers to an integrated Application Programming Interface (API) that combines data from different data destination or third party sources for web services. This web service provides a combined API that is technologically valid and compatible with other web applications. In recent years, web Mashups have been implemented for solving many issues in existing web applications, such as e-library. In this paper, we demonstrate a Mashup prototype by analysing a real service problem in an existing virtual organization to show current limitations of using a distributed architecture. This paper also presents an analysis of web service orientated architectures using the Mashup concept. Based on the case issues in classical web portals, we outline a generic architecture that offers a dynamic solution for providing services to better meet users' needs.

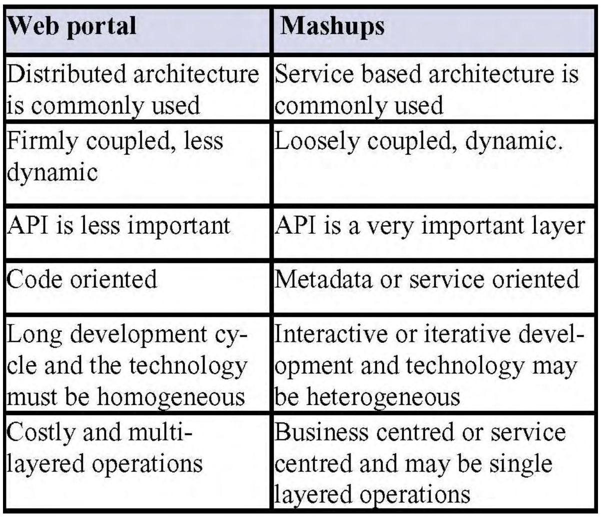
Index Terms— Mashup, web applications, web portals.

I. Introduction

Emerging technologies have been used for many end user enabled web-services and these emerging technologies have recently drawn significant interest in the web community. Digital library services (for examples see Curran, Murray and Christian, [1]; Pearce, [2]) provide examples where emerging technologies have been used for enhancing end user's active participation. Web applications driven by end-users may not necessarily use traditional requirement-and-build approaches especially in the area of web based service development. Business processes are evolving due to the potential for improved virtual operation and immediate increased value in human capital, through the use of emerging technologies. Therefore, developers have started using emerging technologies for web based user interface design, service composition design, as well as social or community based features. These are designed to create increasingly interactive applications.

As conventional technologies for web services suffer from potential weaknesses such as dynamicity, scalability and flexibility, the view of emerging technologies is that they offer an innovation to end users, business entities, and communities. The digital ecosystem concept reinforces an approach that complements such initiatives for a better way of solving current business problems and encouraging user driven activities through the use of latest Web 2.0 technologies such as Mashups, wikis, and blogs. As an extended version of Miah and Gammack [12], this paper proposes a framework, which is driven by users in a business context, for an end-user-enabled web portal, in which Mashup technologies are used for web service creation and composition. Table 1 illustrates the difference between the web portal and Mashups.

Web services provide applications offering 'platform independent' service components available on the Internet. Today many application services are aggregated from a combination of diverse sources that are suitable third-party web services. This structure has been named service composition by web-service communities [3]. This type of composition is no longer being created using fixed HTML code, rather from seamless, dynamic applications assembled using Mashup concepts into an integrated experience. HTML based solutions suffer from issues including dynamicity, scalability and customizability for end users who sustain businesses. Thus, it is a critical to business acceptance that end users participate in mashing up different existing services to leverage and integrate effective new service applications.



Web portal	Mashups
Distributed architecture is commonly used	Service based architecture is commonly used
Firmly coupled, less dynamic	Loosely coupled, dynamic.
API is less important	API is a very important layer
Code oriented	Metadata or service oriented
Long development cycle and the technology must be homogeneous	Interactive or iterative development and technology may be heterogeneous
Costly and multi-layered operations	Business centred or service centred and may be single layered operations

Table 1: Differences between web portal and Mashups

Emerging technologies such as wikis, blogs, and Mashups offer improved user privileges in web service space. Wikis offer an open but shared web space for users where they can collaboratively contribute to complete a task. This type of web application provides basic options such as add new, modify, and remove content from the web. The acceptance of wikis in the business environment is not seen as a fad but a long-term investment in the future [4]. Businesses are finding more uses for wikis including collaboratively writing documents; or writing research papers, a central repository for project information to which all project team members can contribute, writing project management documents, creating service manuals, and creating notes of meetings on the fly [5]. Blog is a short form of web log. A

978-1-4244-2346-0/09/$25.00 ©2009 IEEE

blog contains journal type entries from user groups. A number of blog applications can be found these days including Blogger as a Google application. Kulathuramaiyer [6] stated that a blog enables anyone to become a publisher of their own content with the ability to modify content from any source.

Mashups can be described as a method of merging existing service content or applications from multiple web sources. Yee [7] describes Mashups as using "XML and web services to reuse or "remix" digital content and services. A Mashup is a web application that seamlessly combines web content from multiple sources; this is provided through relatively simple Web Application Programming Interfaces (APIs) [12]. Dillon, Wu and Chang [8] describe how Mashups enable virtual organizations to move from service level to user level, as the key to value creation. The main components of the Mashups are the data and an API for providing data access to the user. In this paper, we focus on the technological background of Mashup through outlining a combined web API for a specific case application and propose a generic architecture of a prototype for a web portal.

Mashups offer features for web service development for businesses. Other Web 2.0 technologies are not generally associated with direct business applications; rather they are used for social networking through enhancing user participation or collaboration [12]. Thus in this paper, we focus on Mashup technologies and how these can be used specifically in replacing a distributed structure for web portal systems.

Mashups represent both a concept and a technology for integrating web applications or services coherently [9]. While the source content is normally implicit on the web, Mashup technology helps process relevant content to be integrated, whereas existing web service architectures would have the potential for problems. Such problems include scalability, performance, flexibility, and ability to implement [8]. For example, a web service is expressed as relevant with a Web Services Description Languages (WSDL) that specifies only the syntax of messages that enter or leave a program [2] The order in which messages have to be exchanged between services must be described separately in one specification However, the composition of this type of service flow is still manually obtained though there are specification languages such as WSCI and BPEL4WS available. Semantic web application development researchers have addressed this problem by providing grid based solutions. For instance, GridSpace is described as a solution which incorporates soft aspects of the grid, service-orientation, transient service space, and web architecture [8]. In fact, GridSpace architecture provides infrastructure to enable service discovery and Mashup at various levels [12][8]. Others including Fox and Pierce [11] have suggested applications, infrastructures and technologies for e-Science environments. In general (enterprise and distributed environment) these authors claim that Web 2.0 can provide narrow grids for building web services that may provide a robust managed environment. Previous research for grid based solutions has focused on data rather than infrastructure [12].

Understanding these instances in relation to the explicitly relevant issues of developing web based service applications is required. We need effectively to address the web services composition issues and the general problem of user directed requirements that apply to numerous virtual business operations. We have selected a realistic web service application for case analysis leading to a model Mashup solution and design for rich Internet applications

The virtual business we analyse here (real but presented anonymously) is a wholesaler's portal based in Europe. The web business has been operating for more than ten years selling computer goods, hardware, electronics household goods etc. We investigated the businesses processes and found that they used a complicated distributed architecture. We propose a flexible alternative solution using Mashup which is suitable and easy to implement for online business.

This paper is organized as follows. Section II discusses the problem space and section III Mashup technologies. Sections IV and V presents the research case, and conclusion and discussion.

II. Problem Space

Traditional web portal systems are problematic as they are a closed and rigid platform with tight rules for exchange of data and services. Processes are rapidly changing in businesses and the processes are getting more complex, requiring multiple users from different management levels to participate and collaborate to address the complexity. Businesses need to explore the potentialities for enhanced virtual operation, through the use of emerging technologies such as Mashups. Examples of this use of emerging technologies such as Mashups has been demonstrated in personal learning environments [15] and course management [16]. Both of these present a better way for managing complex learning resources from multiple sources. In the business realm, customer demands are changing, as well as marketplaces, competition, and business partners. Therefore, advantage is conferred by the innovative use of new technologies rather than their mere presence. Such attributes imply business problems may be solved in a dynamic way however, the current web development technologies do not yet fully support such types of service solutions.

Existing technologies support the need to provide more design middleware and the replacement of old systems. Rather than being about reusing the old applications in a new way, it is about remixing the system solutions. This remixing of system solutions can derive services from businesses' existing services. Therefore, increasing its ability to cope with change through, empowering end-users and enabling innovative concepts. Mashups offer flexible composition of businesses existing services within an improved user interface environment-through improved API's suitable for end users in their context of use [12].

III. MASHUP TECHNOLOGIES

Web application Mashups generally offer more than simple merging of services and content. These sites typically value add through benefiting users in ways that are different and better than the individual services they leverage [17]. The technologies for integrating services retrieved from a variety of sources, such as newsfeeds, maps, and weather forecasts, can be classified in two primary styles, server-side Mashup and client-side Mashups [17] [9]. The server-side Mashup technologies integrate services and content, by acting as a proxy between a web application on the client, typically a browser, and the other website that takes part in the Mashup. In a server-side Mashup, all the requests from the client go to the server, which plays a role as a proxy to make calls to the other web site. The main task of the client in the server-side Mashup is to push from the web application client to the server [9].

The following steps describe the activities of server-sided technologies [9] and displayed in Fig 1. Where the web services or contents on the web server are mixed and transferred to the client throughout the HTTP protocol [17]. The technical description that follows in this section is taken from Ort et al. [17]

A. A user generates an event which triggers a JavaScript function in the client. This is typically a webpage.
B. Through the primary website the client makes a request to the server.
C. Then a servlet calls an encapsulating method of the code to connect and interact with the second website in the Mashup, after receiving the request.
D. Before opening a connection to the Mashup site, the proxy class processes the request and augments it as needed.
E. The Mashup returns data to the proxy class after receiving the request and processing.
F. The proxy class on receiving the response can transform it to a more suitable format of the data.
G. The servlet then returns the response to the client.
H. Updating the clients view through a callback function.

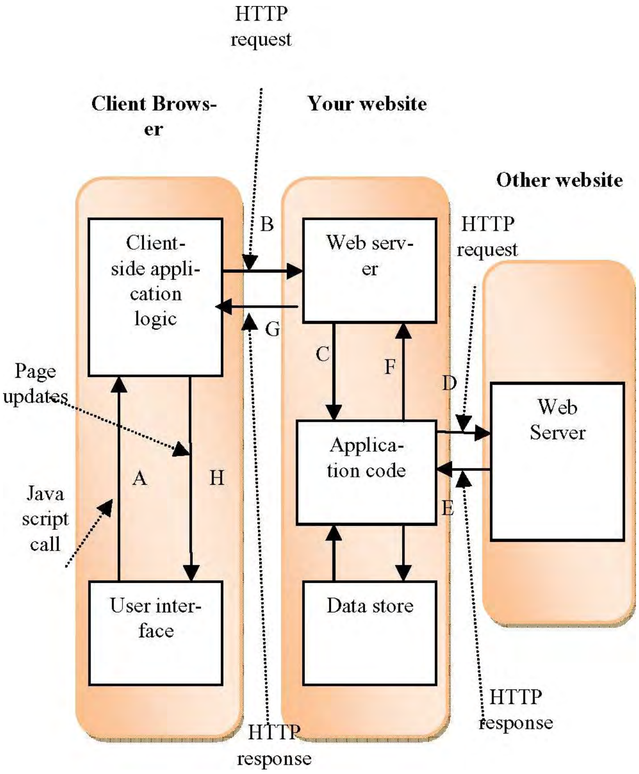


Fig 1 A Server-Side Mashup
[source: 17]

Unlike server-side Mashups, client-side Mashups integrate services and content on the client, directly combining with the other web site's data or functionality. According to Auinger [9], the following describes this process and is displayed in Fig 2. [17]:

A. The browser requests the server through your website for the webpage.
B. Your websites server loads the page into the client.
C. The browser page calls a function, through some action, using the JavaScript library provided by the Mashup site
D. ,Requesting the Mashup site load the script which is based on the <script> element
E. The script is loaded by the Mashup site.
F. Updating the client view of the page through the callback function.

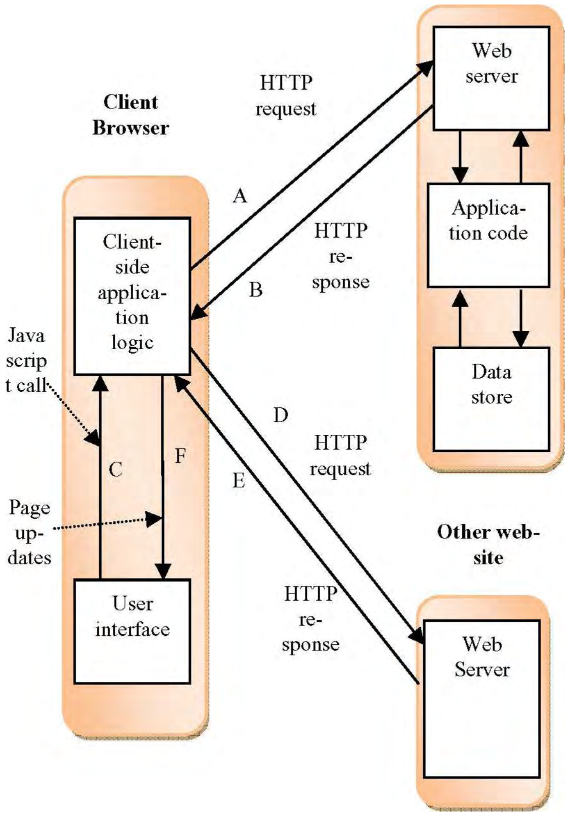


Fig 2 A Client-Side Mashup
[source: after 17]

IV. Research Case

This case uses an existing portal based in Central Europe selling software, hardware and household electronic accessories. This site “http://yannipes.eu” is a virtual business offering comparison pricing on products based on third party companies pricing and product data. This leveraging to a full service platform is through their business relationships with the enlisted companies. The portal incorporates the third party company’s data and acts as a wholesaler portal. Consumers can purchase product from the range of brands using the latest product information including price, product description and availability [12].

Although this type of aggregator business model is commonly found across the web it presents a mature model for widespread data sourcing. It has a number of issues for both the customer and a company point of view. One issue is that pricing and specification changes of items need to be reflected quickly in the marketplace, to maintain a comparative environment. Ensuring customers can make better informed decisions due to the currency of the information. However, the current system, which uses distributed database architecture, is less dynamic when the enlisted company changes/updates information. New business models making use of the improved and flexible developments, such as Web 2.0, are needed. Figure 3 outlines the current data structure of the Yannipes site.

In this context of distributed architecture, a Mashup could address the currency of the information by providing a more dynamic and evolving platform that adapts to the businesses requirements. As the enlisted companies update their own database, when they add/modify their existing product range, the portal can be dynamically updated. Through mashing up all the third parties data resources into a new API the customer can see their desired products in an environment where comparison is easy and the information is current.

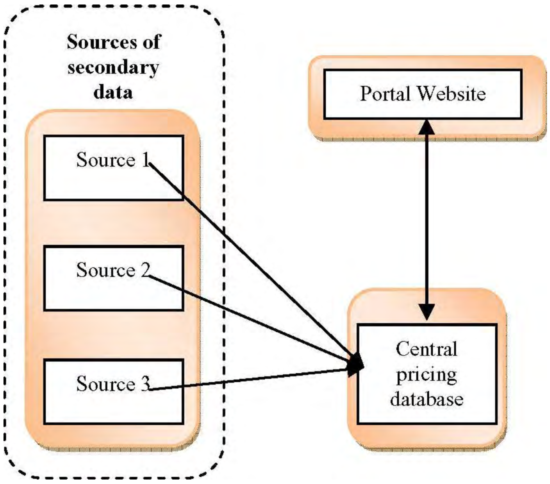


Fig 3 Yannipes general architecture where distributed database technologies are implemented.

We now discuss a new solution architecture that can address this issue. In particular our proposed approach replaces this distributed concept by utilizing Mashup concepts to provide a combined API for a business portal where customers can get a more dynamic service system, related to their specific needs

The illustrative application presents a Mashup architecture for the paradigmatic “technology product aggregator service” where end users can obtain dynamic and user-centrically directed information to make decisions about candidate accommodation purchases. This solution architecture is intended to be generic, interoperable and applicable in any similar problem domain such as for any online business portals.

V. Proposed Architecture

Figure 4 illustrates conceptual Mashup architecture for a solution to the product aggregator service problem. In this diagram each third party acts as a data island providing its data service by acting as a content server, that is to say it will receive the data in the form of a mediated request (via XHTTP, REST or SOAP) and return what in other circumstances might be a legible data report. In our example we shall consider data about the products prices, description, availability, and payment terms, but the principles applied can be generalised to any situation where the needs for such a system arise.

In this as in any Mashup style application, the critical thing to note is that there may be little if any commonality between the form or expression of data between the different providers (and indeed the required representation for the Mashup itself may be a fourth schema and style again. This is where the Mashup architecture comes into its own.

The designer of the Mashup must facilitate the user request into forms that are suitable for each of the content providers, and in turn, the responses (or lack thereof) must be merged back into a single data stream. The development of the API will specify the form in turn in which the requests can be made, and the manner of presentation.

APIs are only as good as their publication, and a final stage in the design deliberation will involve the creation of the specification. If the process is going to be treated as intellectual property then it may be prudent to conceal the inner workings of the data factoring and merging. The chief drawback of this is that of any closed-source system, that the more eyes on the design, the greater the chances of correction. With Mashups this may be far more critical than with ordinary coding or data design scenarios – the Mashup will only work if the content servers stay constant in their design, and the process of publication of the internals means that the content service owners as well as the API users can verify that the expectations of the system designers meet the initial requirements of the content servers, and that they continue to do so over time.

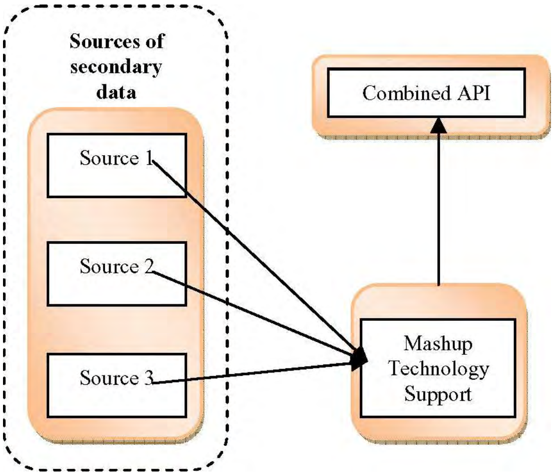


Fig 4 Conceptual model of generic mashup portal archecture.

We have outlined a generic Mashup architecture for business services based on this conceptual model for web end users. Figure 5 illustrates the architecture for a technology product aggregator service. It consists of four layers in line with the ideas presented by Dillon [8]. These are at least in part orthogonal to the states present in figure 3.

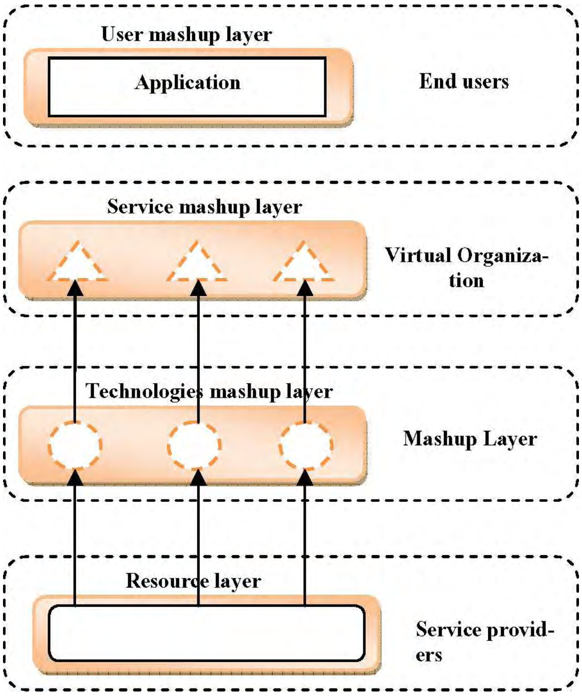
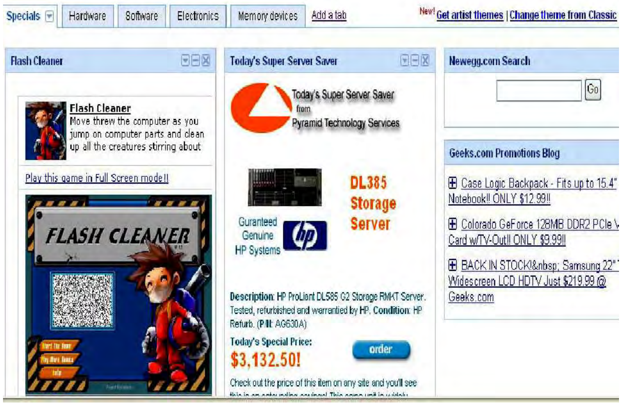


Fig 5 Mashup architecture for a technology product aggregator service.

These are firstly, the service providers-resource layer where the content servers are located. This is followed by the technologies Mashups transient layer where the Mashup technologies create a combined interaction for data from different resources. The virtual organizations-service Mashup layer is next where the Mashup services are prioritized for Mashing up data from content servers, and finally the user Mashup layer where the data is presented to the web end users according to their requirements.

Figure 6, illustrates a prototype demonstrating how a Mashup application can be built for a retail portal. This prototype contains a number of feeds that are used to represent different categories of the product. This can be outlined through the Mashup of existing content. In this case, the existing content is the price lists and the banners of the special offers displayed from the source websites. A combined API just re-presents this information using different tags in the new system. In an application such as in iGoogle, this type of personalised solution can be prototyped for any similar business. The business can add or modify the secondary content at anytime for the target customers. In addition their target customers, at the same time, can achieve greater uptake through the ability to compare between products from different brands, therefore, making customers buying decisions easier and better informed. Our aim here has been mainly to describe the architecture and its motivation, but now that a prototype has been built, claims can be tested directly for user cases in future.


Specials
Hardware
Software
Electronics
Memory devices
Add a tab
New! Get artist themes | Change theme from Classic
Flash Cleaner
Today's Super Server Saver
Newegg.com Search
Go
Geeks.com Promotions Blog
Today's Super Server Saver from Pyramid Technology Services
DL385 Storage Server
Guranteed Genuine HP Systems
hp
FLASH CLEANER
order

Flash Cleaner
Move threw the computer as you jump on computer parts and clean up all the creatures stirring about

Play this game in Full Screen mode!!

Description: HP ProLiant DL585 G2 Storage RMKT Server. Tested, refurbished and warrantied by HP. Condition: HP Refurb. (P/N: AG630A)

Today's Special Price: $3,132.50!

Check out the price of this item on any site and you'll see

Case Logic Backpack - Fits up to 15.4" Notebook!! ONLY $12.99!!

Colorado GeForce 128MB DDR2 PCIe Video Card w/TV-Out!! ONLY $9.99!!

BACK IN STOCK! Samsung 22" Widescreen LCD HDTV Just $219.99 @ Geeks.com

Fig 6: A prototype of Mashup portal applications developed using iGoogle Mashups.

VI. CONCLUSION AND DISCUSSION

This paper described how Mashups techniques can be used for replacing traditional web portal systems. We proposed a generic architecture using Mashup technologies based on processes from both the recognised standard client and server sided Mashups. This type of architecture can leverage and integrate the end user relevant information from the existing web applications in the web. Although this architecture may address issues in a traditional web portal system, implementing the concept would need consideration of issues such as intellectual property and organisational boundaries both of which can create implementation issues for these types of services. In addition, sensitive data required may need to use encryption when this data mashes up with data from other sources, to provide confidential handling. Finally, a third party content provider's unwillingness may interrupt the free flow of information.

The main contribution of this paper is the linking of the emerging technology in the form of Mashups to better support the requirements of users for up to date and interactive websites. We have sketched how a Mashup approach can be applied to overcome limitations of current virtual organisations based only on a distributed architecture and without provision for directed data sourcing at the user layer. Detailed specification for this type of application would be problem specific, but the architecture itself is considered an advance in allowing more dynamic inputs from different data sources, directed by specific context requirements of the user.

VII. REFERENCES

[1] K. Curran, M. Murray and M. Christian. "Taking the information to the public through Library 2.0" 2007. Library Hi Tech, 25(2), 288-297.

[2] J. Pearce User collaboration in websites. 2006. Available at http://www.nla.gov.au/nla/staffpaper/2006/jpearce.html. Retrieved July 2, 2008

[3] B. Srivastava, and J. Koehler, "Web Service Composition - Current Solutions and Open Problems", IBM India Research Laboratory. Block 1, IIT, New Delhi 110016, India. 2006, Available at http://www.zurich.ibm.com/pdf/ebizz/icaps-ws.pdf

[4] L. Bean, and D. Hott "Wiki: A speedy new tool to manage projects".Journal of Corporate Accounting & Finance, 2005, Vol. 16 No 5, pp3-8.

[5] J. Cronin. "Upgrading to Web 2.0. An Experiential Project to Build a Marketing Wiki". Journal of Marketing Education, 2009, Vol. 31 No.1, pp66-75.

[6] N. Kulathuramaiyer. "Mashups: Emerging Application Development Paradigm for a Digital Journal". Journal of Universal Computer Science, 2007 Vol 13, p 531-542

[7] R. Yee. Mashups, IST-Data Services, 2006, Available at http://dret.net/lectures/services-fall06/Mashups.pdf

[8] T.S. Dillon, C. Wu and E. Chang. "GRIDSpace: Semantic Grid Services on the Web:Evolution towards a SoftGrid", In proceedings of the 3rd International Conference on Semantics, Knowledge and Grid, Oct.29-31, 2007. Xian, China.

[9] A. Auinger, M. Martin, D. Nedbal, and A. Holzinger. "Mixing Content and Endless Collaboration – MashUps: Towards Future Personal Learning Environments". 2008, Available at https://online.tu-graz.ac.at/tug_online/voe_main2.getVollText?pDocumentNr=95304 &pCurrPk=42744, Retrieved on 23 March, 2009

[10] Christensen & others, "The web services description language WSDL" Available at http://www-4.ibm.com/software/splutions/webservices/resources.html, Retrieved on 23 March, 2009

[11] G. Fox, and M. Pierce. "Web 2.0 and Grids", In proceedings of the 3rd International Conference on Semantics, Knowledge and Grid, Oct.29-31, 2007. Xi'an, China.

[12] S. J. Miah and J. Gammack. A Mashup Architecture for Web End-user Application Designs, Second IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2008), 2008. p 432-537

[13] S.L. Chan. "End-user directed requirements: a case in medication ordering In Advanced topics in end user computing" ed M.A. Mahmood IGI Publishing Hershey, PA, USA, 2002

[14] J.G. Gammack, "Constructive design environments: implementing end-user systems development", Journal of End User Computing, 1998, vol.11 n.1, pp.15-23,

[15] I.H. Hsiao, Q. Li, Y. L. Lin. Educational social linking in example authoring. Proceedings of the 19. ACM conference on Hypertext and hypermedia, (2008) 229-230

[16] N. Hardy, M. Pinto, and H. Wei. "The impact of collaborative technology in it and computer science education: harnessing the power of web 2.0". Proceedings of the 9th ACM SIGITE conference on Information technology education, (2008) 63-64

[17] E. Ort, S. Brydon, M. Basler. Mashups Styles, Part 1: Server-Side Mashups (Sun Microsystems). 2007 Available at: http://java.sun.com/developer/technicalArticles/J2EE/mashup_1/, Retrieved on 23 March, 2009