

Mashups and the Enterprise

/// WHITE PAPER



Sunilkumar Peenikal
September, 2009

In web development, a mashup is a web page or application that combines data or functionality from two or more external sources to create a new service. The term mashup implies easy, fast integration, frequently using open APIs and data sources to produce results that were not the original reason for producing the raw source data.

Mashups are surely accelerating the focus on do-it-yourself (DIY) and self-service applications. Even a cursory examination of what people are doing every day on the web right now tells us that mashups are indeed happening in a large way within the consumer space, but inside our organizations/Enterprise, whether in the IT departments or in the lines of businesses, mashups are still a rarer occurrence.



Figure2: Mashup of Craig list website, Address Lookup and Yelp widget

Data Mashups combine similar types of media and information from multiple sources into a single representation. An example of data mashup is the Havarria Information Services' AlertMap, which combines data from over 200 sources related to severe weather conditions, biohazard threats, and seismic information.

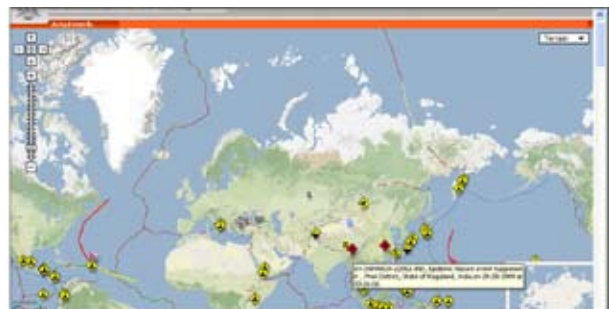


Figure 3: A Mashup of Google map with Data from various sources

Business Mashups focus data into a single presentation and allow for collaborative action among businesses and developers. This works well for an agile development project, which requires collaboration between the developers and customer proxy for defining and implementing the business requirements. Business mashups differ from consumer mashups in the level of integration with business computing environments, security and access control features, governance, and the sophistication of the programming tools (mashup editors) used. Another difference between business mashups and consumer mashups is a growing trend of using business mashups in commercial software as a service offering. The image below explains how an enterprise mashup environment will look from an end user perspective

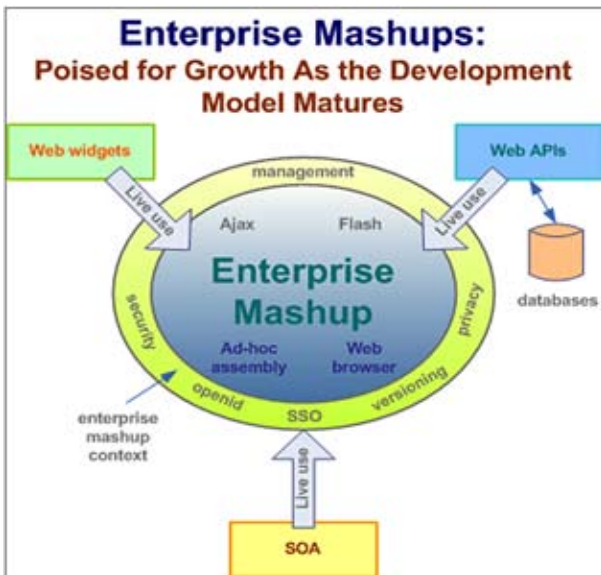


Figure 1: Enterprise Mashup Model (source: ZDNet.com)

There are many types of mashups, such as consumer mashups, data mashups, and business mashups.

A Consumer Mashup is aimed at the general public. Consumer mashups, when offered with web-based mashup facility can prove to be a very effective means for customer personalization of data/viewing, this is typically where the user's use their web browser to combine and reformat the data according to their needs. Figure 2 demonstrates an example where a mashup of Craig list website, Address locator widget and Yelp widgets is done using Intel Mash Maker software. As a consumer, I was able to change the craigslist website (The Home icon & Flower Icon against each listing is not a feature of craigslist site) to more suite my need of finding an apartment nearby my favourite restaurants.

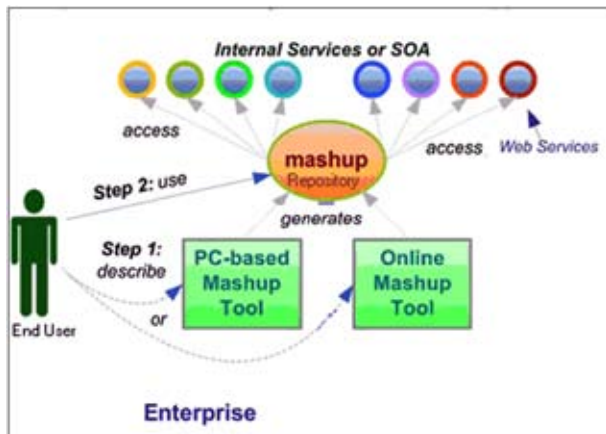


Figure 4: End User view of an Enterprise Mashup Environment

Architectural Aspects of Mashups Architecturally, there are two styles of mashups: web-based and server-based. Web-based mashups typically use the user's web browser to combine and reformat the data, server-based mashups analyze and reformat the data on a remote server and transmit the data to the user's browser in its final form.

Mashups can appear to be a variation of a Facade pattern, that is, it is a design pattern that provides a simplified interface to a larger aggregate of different feeds with different APIs. Mashups can be used with software provided as a service (SaaS).

After several years of standards development, mainstream businesses are starting to adopt Service-oriented Architectures (SOA) to integrate disparate data by making this data available as discrete Web services. These web services can be reused to provide completely new services and applications within and across organizations, providing business flexibility. This shift can help mashup gain acceleration and enterprises adopting it as a next step towards self-service and DIY.

Key Aspects and Benefits of the Mashup

- **Effective Leverage of Web Parts and the Enterprise SOA**

Mashups are generally built out of the bits, pieces, and services of other Web applications that already exist, or data that already exists but is scattered around in multiple applications. Additional coding is only required when data can't be sourced from internal or external suppliers or to provide integration "glue" between the parts. Enterprise can work this feature of mashup to its advantage as they can easily re-use their existing applications as a service for the mashing it up.

- **Simple, Lightweight Software Models and Services**

By focusing on the simplest possible techniques and formats, web mashups appear to be successful and widespread primarily because just about anyone can create them. Mashups are typically built using techniques like cutting and pasting snippets of JavaScript, using RSS feeds and XML to connect the various parts together, and even one-line JavaScript inclusions that can pull in and integrate a powerful external component, such as Google Maps or a YouTube video player, that originally required a massive investment from its creator. Google makes its AdWords and Maps widgets incredibly easy to install and deploy while YouTube even puts the hosting code next to each and every video on its site. Finally, mashups are 100% pure Software as a Service (SaaS) and require no installation, updates, plug-ins, admin rights, or anything but a garden variety Web browser and the mashup's URL to run.

- **A Focus on Self-Service and DIY**

As mentioned earlier, mashup development can be just as much for advanced web users as it is for professional software developers/ designers. Mashups have the potential to give end-users the ability to create real, useful software. Mashup also means that applications that could never have been justified on a build-vs.-buy perspective are now possible. These apps can now just be created by end users – and groups of collaborating users – on the fly as they need them. It could also reduce the application backlogs that continue to bedevil IT departments and their customers everywhere. For example just about anyone can now create the views, dashboards, and other software applications that let them get their work done better and faster. The developers' responsibility is reduced to creating appropriate widgets and a repository for the widgets, such that user can search for the widget with the required business function and they can present it the way they want it to be presented.

• **Accelerated Time to Market**

The componentization of business processes and the use of enterprise mashups within an organization will accelerate the speed at which new business services can be delivered by an order of magnitude. This represents a significant competitive strength; organizations that are not pioneers in their use of enterprise mashups will inevitably find themselves non-competitive.

• **Mashup Versus Portals**

More and more organizations are moving towards a portal environment which is definitely a good sign, as portals have proven, the value, that it can bring into enterprise with its demonstrated strength in the software functional areas of source aggregation, content analytics and classification &, individual and collaborative filtering etc.,. What must be understood is that mashups and portals both share content aggregation technologies. Listed are few differences between mashup & portals:

Attributes	Portal	Mashup
Classification	Older technology, extension to traditional Web server model using well defined approach	Using newer, loosely defined "Web 2.0" techniques
Philosophy/ Approach	Approaches aggregation by splitting role of Web server into two phases - markup generation and aggregation of markup fragments	Uses APIs provided by different content sites to aggregate and reuse the content in another way
Content dependencies	Aggregates presentation-oriented markup fragments (HTML, WML, XML, VoiceXML, etc.)	Can operate on pure XML content and also on presentation-oriented content (e.g., HTML)
Location dependencies	Traditionally content aggregation takes place on the server	Content aggregation can take place either on the server or on the client
Aggregation style	"Salad bar" style - Aggregated content is presented 'side-by-side' without overlaps	"Melting Pot" style - Individual content may be combined in any manner, resulting in arbitrarily structured hybrid content
Event model	Read and update event models are defined through a specific portlet API	CRUD operations are based on REST architectural principles, but no formal API exists
Relevant standards	Portlet behaviour is governed by standards JSR 168, JSR 286 and WSRP, although portal page layout and portal functionality are undefined and vendor-specific	Base standards are XML interchanged as REST or Web Services. RSS and Atom are commonly used. More specific mashup standards are expected to emerge

As in any new technology, mashups come with a set of challenges and a few drawbacks to fit into an enterprise arena. Let us try and explore some of the key challenges.

Key Challenges of Mashups

• **Two Major Mashup Models At War**

In general, there are two mashup models that are quite different from each other. First is the Consumer Web Model, where we have the natural, emergent mashup phenomenon as numerous web parts suppliers and their consumers try different strategies out and sometimes hit upon the models that work. The second being the vendor Model, enterprise mashup tools try to add missing enterprise context like security, support for local SOAs, a mashup development model, usually by choosing a particular Web component standard and providing a visual IDE that makes it easier for the non-HTML savvy to create mashups. These commercial mashup application models try to figure out a priori what will work best, and in this they may be making the mistake of ignoring the vast laboratory of the Web that is already proving out highly effective models for mashup parts and integration strategies on a large scale. There is also a skill and open/closed barrier. The consumer web model thrives well in an open source environment there by enjoying a large skill pool, their major focus being on ease of usage, with very little attention to aspects like security, deployment issues, manageability etc.,. On the other hand the vendor model prefer a confined environment to safe guard its business interest, this leads to limited skill pool, also this model need to invest heavily in addressing major concerns like security, deployment, upgrades, managing etc. These differences between the two models tend to prevent the movement or migration from one model to the other.

• **Absence of a Standardized Widget Format**

Both Microsoft and Google have their own gadget models, NetVibes has the compelling Universal Widget Architecture (UWA). The W3C has a draft of the Widgets 1.0 specification. A consistent widget format is required to build a visual tool that can fully enable mashup development and realize the productivity potential. The enterprises are not yet anxious to start figuring out which widget models meet their needs and, unfortunately, no obvious solution is on the horizon.

• **Shortage of Web Services within Enterprises**

There just aren't enough web services within enterprise available to supply the data and back-end functionality to mashups. Most data resides on legacy systems and/or standalone web based intranet applications that are still in static web pages in the form of HTML. A complete SOA based enterprise architecture will take years and years. In the meantime we need to quickly service-enable our silos of web-based information if we are to fully exploit the mashups value. Fortunately, this challenge can now be partially mitigated through offerings of companies like Kapow, Intel and Yahoo Pipes. For example Kapow's RoboMaker can be

used to transform Web 1.0 content into a Web 2.0 friendly service format like RSS in a few minutes using visual tools. Thus widgets are ideal for consumption by providing an easy-to-use service.

- **Issues Related to Security and Identity**

This shortcoming is the most important challenge. If not addressed in time it can make mashups just another web consumer play tool and not of real importance and value to the enterprise.

User Identity & Security. The most useful enterprise mashups will have access to a user's individual data and other corporate information protected by security and identity systems. Identity management is crucial to the governance of mashups, enabling business management to keep control over sensitive information.

Governance. & Version Management. Governance issues, including version management, quality of service, security, backup, disaster recovery, and customer privacy are essential considerations for any enterprise looking to adopt mashups as a way to bolster productivity. Since mashups can be rapidly recombined on the fly, it is important that a mashup platform rigorously maintain version control, so that users have the most up-to-date software available, and new versions can quickly be tested and fixed.

- **Addressing the Tension Between the Two Major Styles of Integration**

Most integration today is done up front with lots of testing and configuration control and baselined code from your external suppliers. On the other hand, mashups rely on live pulls of code from your supplier and are a much more extreme form of combining systems together. This new live model of integration has security, testing, and version control issues written all over it. Google and some of the larger suppliers are getting a handle on some of these, but market leadership on all of these issues is a long way ahead.

- **Other Enterprise Contexts**

To get a foot in the door of the enterprise, mashup tools need to have solid stories around single sign-on (SSO), LDAP, JSR168 (portals/portlets), legacy integration, management, monitoring, RSS strategy, governance etc. Most mashup tools are still falling short in these categories and will likely not get broad adoption until these issues are addressed.

Summary & Conclusion

Mashups have the potential to create significant competitive advantage by allowing for rapid innovation of business processes through the co-creation of new systems which harness the collective intelligence of an organization, they can also prove very effective and bring in value to enterprise with its unique aspects & benefits. Mashups help move Web 2.0 into the enterprise and can reduce application development costs, improve time-to-market, increase productivity, leverage SOA investments, and spur innovation. But enterprises are currently in a wait and watch mode, as some of the hard pressing challenges like governance, security, manageability, standardization are yet to be addressed.

References

Programmable Web now lists over 3000 registered mashups:

<http://www.programmableweb.com/mashups>

Make a mashup of any website using RoboMaker's

<http://openkapow.com/>

Personalizing websites using Intel MashMaker

<http://mashmaker.intel.com/web/>

Mashup Product Suite for Business & Data Mashups

<http://www-10.lotus.com/ldd/mashupswiki.nsf>

<http://pipes.yahoo.com/pipes/>

Contact us

USA

MphasiS
460 Park Avenue South
Suite # 1101, New York
NY 10016, U.S.A.
Tel: +1 212 686 6655
Fax: +1 212 686 2422

UK

MphasiS
88 Wood Street
London EC2V 7RS
Tel: 0208 528 1066
Fax: 0208 528 1001

MphasiS
Edinburgh House
43-51 Windsor Road
Slough SL1 2EE, UK
Tel: +44 0 1753 217 700
Fax: +44 0 1753 217 701

INDIA

MphasiS
Bagmane Technology Park
Byrasandra
C.V. Raman Nagar
Bangalore 560 093, India
Tel: +91 80 4042 6000
Fax: +91 80 2534 6760

About MphasiS

MphasiS, an HP company, delivers Applications Services, Infrastructure Services, BPO and KPO Services through a combination of technology know-how, domain and process expertise. We service clients in the Manufacturing, Financial Services & Insurance, Healthcare, Communications, Media & Entertainment, Transportation & Logistics, Consumer & Retail industries and to Governments around the world. We are certified with ISO 9001:2000, ISO/IEC 27001:2005 (formerly known as BS 7799), assessed at CMMI v 1.2 Level 5 and are undergoing SAS 70 certification. We also provide SEI CMMI, ISO and Six Sigma related services support.

MphasiS is a performance based company, dedicated to outstanding customer service. We offer capabilities to provide innovative solutions by sustainable cost savings and improved business performance through flexible engagement models. Customer centricity, transparency in operations, result-oriented activity and flexibility are the values on which we build long-term relationships with our clients.

