

Exploring Mashups for Business Benefit

Introduction

With the advent of Web 2.0, Mashups have become a familiar buzzword to both hard-core developers and casual users alike. Unlike some other web-centric technologies, mashups are relatively easy for almost anyone to understand from a conceptual level. A mashup is “*a web application that combines data from more than one source into a single integrated tool. The term Mashup implies easy, fast integration...*”¹. The concept of easy and fast integration has become increasingly important in business circles, since most software and systems development is not so-called “green field” -- completely new applications. Instead, much of the application delivery effort entails the integration of existing systems and services. Further, many of these existing applications are long-lived business critical solutions that are either too expensive to rewrite or too critical to rewrite and get **wrong**. Therefore, the ability to rapidly compose business applications utilizing existing assets is of primary importance to maintain competitiveness for businesses. This paper will examine mashups, and specifically how mashups are being positioned in business as an enabling technology for companies that have (or will) adopt the approach.

Mashups

Mashups: What

“*A Web mashup is a Web page or Web site that combines information and services from multiple sources on the Web... [It] lets you connect, collect, and mash up anything on the Web as well as data on some backend systems*”².

Mashups allow users to combine, mix, match and present data in a way that is useful to them, thereby creating a new solution for a given situation. Simmen et. al call these types of applications “situational applications” since they are not general purpose solutions; rather, they are targeted intranet applications delivered for a particular a user in a short timeframe³.

As with many of the Web 2.0 technologies, mashups were not initially used extensively, nor were promoted by, business users. Instead, mashups attained much of their “buzz” from non-business internet users and web-saavy technocrats who used them to quickly deliver innovative and useful web-centric solutions. In face, in 2006, van der Vlist and colleagues found that more than 40 percent of mashups were mapping mashups making use of Google or Yahoo Maps⁴. While geospatial and cartographic applications are certainly useful (particularly for non-business users), having a plethora of mashup applications identifying “jobs near UMass Lowell” only goes so far in terms of managing information for

¹ Wikipedia.org

² Murugesan, pg 36

³ Simmen et. al, pg 1171

⁴ Murugesan, pg 36

business benefit. This leads to one of the problems with Mashups: while mashups make it easy to create new solutions “on the fly” by mashing up existing data and services, the problem of application lifecycle management and governance still exists. Whether in business or in the general-purpose user space, the questions arise:

- How many versions of <situational application X> are there, and how many do we need?
- If we change service Y, how does that affect the consumer mashups that depend on it? And,
- If mashup P and mashup Q do just about the same thing, should we merge them into a single mashup to reduce effort/maintenance/overlap, or not?

These questions are less imperative for internet users creating mashups outside the business environment since the Web by nature is self moderating and self governing to a great extent. However, as mashups are adopted in business settings these questions are paramount to software delivery and I/T management organizations in the company since it has a direct affect on resource utilization and the bottom line.

Mashups: How

As mentioned earlier, mashups are essentially composite applications that utilize existing data and services. The art of how to deliver mashup applications has continued to evolve. Since mashups are applications, they can be coded and delivered by programmers and technical users. However, since situational applications are quite often delivered by non-technical people, the ability for this constituent to be able to deliver solutions via mashups has been touted greatly. Regardless of the “masher’s” technical skills, Murthy et. al describe the general lifecycle of mashup development:

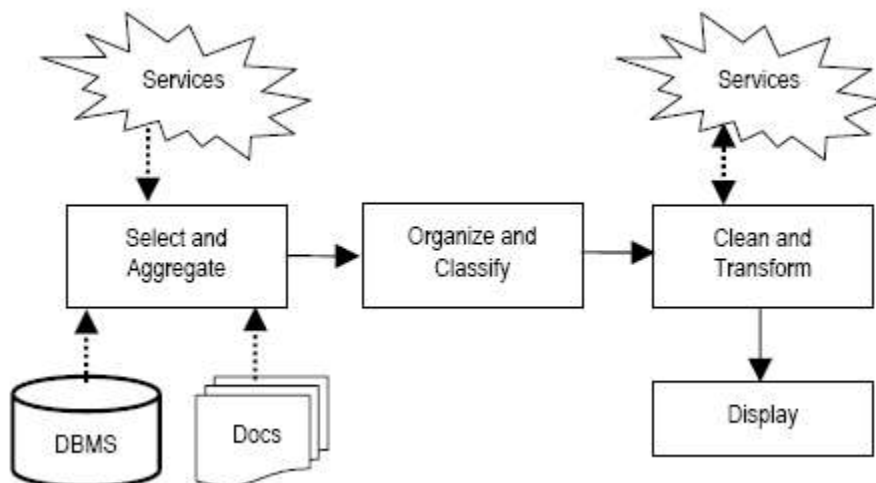


Figure 1: Mashup Development Lifecycle Steps⁵

The key steps for mashup developers of all kinds:

⁵ Murthy et. al, pg 206

1. **Select and Aggregate:** Determine the composite application, and select the information sources to aggregate for the solution.
2. **Organize and Classify:** Structure the fragmented information into a holistic view for usage
3. **Clean and Transform:** Query the information sources and extract the subset of data needed; transform the data into whatever format is applicable, and prepare for presentation.
4. **Display:** Present the information to end-users

Programmatically, some of the technology and techniques used to perform these tasks may be a combination of Javascript, XML, XSLT, HTML, and other scripting. A good example of this style of traditional programming (e.g. with a text editor) is the Google Mashup Editor. The Google Mashup Editor supports AJAX with declarative XML tags, JavaScript, CSS, and HTML. On their site, Google states: “*For advanced developers, Google Mashup Editor allows you to be more expressive by tapping into our JavaScript API*”⁶.

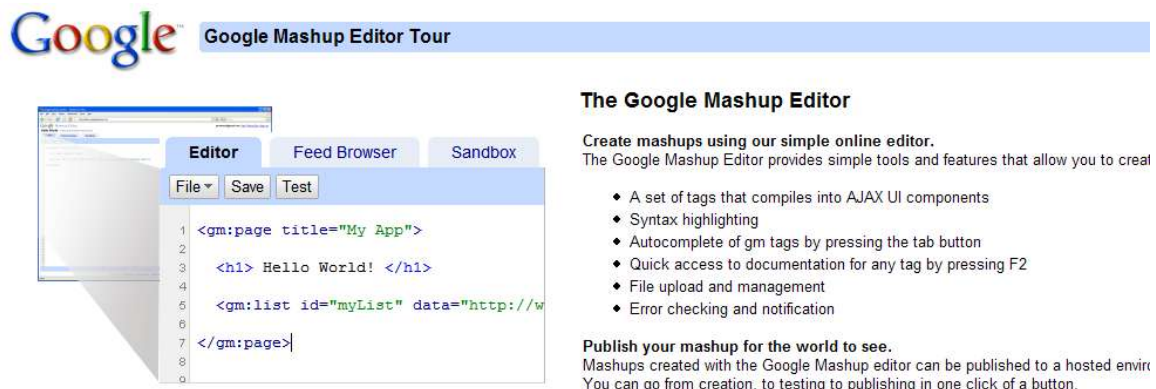


Figure 2: Google Mashup Editor

However, considering the previous discussions on accessibility to non-technical users, traditional programming approaches like the Google Mashup editor are not optimal for a number of potential users. To this end, a number of graphical programming solutions have appeared on the scene to simplify mashup development for all audience members – both technical and non-technical alike. Under the hood, the same core technologies are applied, but the complexity of scripting solutions is hidden behind easy-to-approach graphical construction kits. Essentially, users “wire together” solutions using graphic flow diagram editors to create solutions. IBM Lotus Mashups and Yahoo Pipes are two environments that support this type of graphical metaphor. Graphical environments like Yahoo Pipes allow users to use pre-built operator, and reuse components already delivered as “widgets” by other Pipes (or Lotus Mashups, if in that environment) users.

⁶ Google.com: Google Mashup Editor

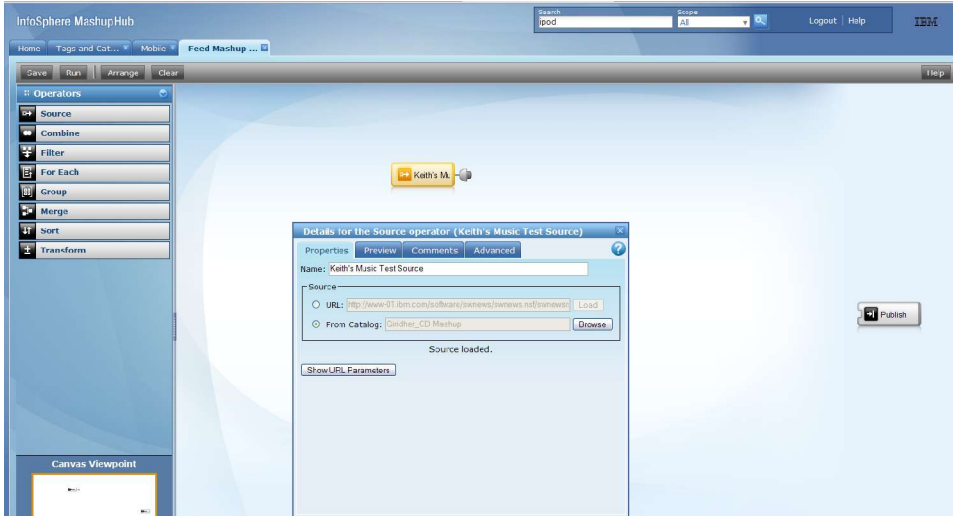


Figure 3: IBM Lotus Mashups

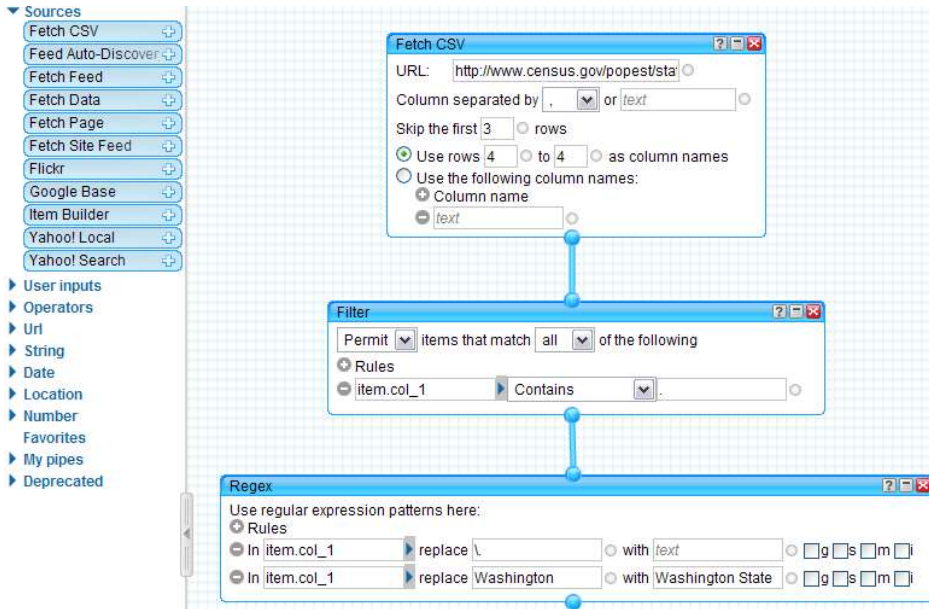


Figure 4: Yahoo Pipes

Business Mashups and Situational Applications

The previous sections established what mashups are, and their general-purpose usefulness. Now, let us focus on mashups specifically for business. For years, so-called “situational applications” have been created by enterprise business users as a by-product of solving day-to-day problems⁷. In the past, tools like MS Word, Excel, or even VBScript may have been used by non-technical business personnel to provide a small-scale, focused solution for business benefit without having to call on dedicated I/T resources. In fact, document-centric solutions approaches based on Word, Excel and the like are so prevalent in the business community

⁷ Simmens et. al, pg 1171

that Boyer researched delivering a “document-based mashup”⁸. His research on ‘Dual Forms’ was geared at mashing up characteristics of an intelligent document supporting web services with elements of business process while retaining the comfort and usability of a MS Word-like document.

This tie to document-centricity and desktop tooling is an extremely difficult one to break in businesses that rely heavily on these types of solutions. In my role as a technical solution provider, I have unfortunately run into many customers who a number of “situational” applications that were meant to be sort-term stop-gap measures that have either:

- a) turned into long-term solutions by accident or
- b) been duplicated over and over again because of a lack of governance

The promise of a Service Oriented Architectural approach (SOA) was sold to many companies as a solution to address these problems. SOA was the “silver bullet” that would speed development (by reusing services), and allow business users to build focused situational applications by composing services as needed. However, the plethora of backend technologies (e.g. Business Process Execution Language [BPEL], Various Web Security standards [WS-*], Process Modeling, etc.) required to reach this goal has remained daunting to the typical non-technical user. Further, when adopting SOA, Enterprise Architecture teams deem solutions should be “architecturally compliant”, otherwise companies end up with the same hodgepodge mess of technologies and approaches that they are typically trying to extract themselves from. These facts have typically made the reality of SOA less of a panacea than the promise was for many companies.

Mashups used for lightweight data and service integration bypass most of the problems outlined above. Further, the availability of easy-to-use graphical tooling described in the previous section of this paper puts mashups on almost equal footing with traditional desktop office tools like MS Excel from a usability perspective. Further productivity enhancements in mashup production is gained through tooling like the Mashup Advisor described by Elmeleegy et. al. in their research, which assists mashup composers by suggesting relevant outputs to augment partially constructed content and reduces the amount of time needed to create mashups⁹. Finally, by promoting an intranet-based business solution, the real possibility of reuse and decreased application redundancy improves. Rather than passing around copies of an MS Excel spreadsheet (with the latest version residing on Joe Employee’s desktop machine), users around the company can point to the singular intranet version that’s been published. If it becomes popular enough, the corporate I/T group may opt to take control, enhance and manage the solution for better performance and scalability. In line with what I have seen in the business environment, Simmens et. al. summarizes the situation thusly: *“In particular, we envision the corporate intranet steadily evolving into a platform of readily consumable resources, and lightweight integration technologies, which can be exploited by business users to create “enterprise*

⁸ Boyer, pg 100

⁹ Elmeleegy et. al, pg 338

mashups" in response to situational business needs. The lines between the intranet and Web will progressively blur as enterprise mashups reach outside the corporate firewall to exploit data and services on the Web”¹⁰

Despite the accessibility and approachability of mashups for business use, there still remain a number of challenges:

1. As previously mentioned, application development requires some type of application lifecycle management and governance. Simplifying the mashup construction process does not address this need at all. In fact, simplifying the construction process may exacerbate the problem by encouraging a plethora of uncontrolled and unmanaged applications to be constructed. Additionally, the question of if a mashup repository (and repository access) becomes of increasing importance in business environments.
2. The graphical tooling outlined in this paper makes mashup construction simpler, but in many cases, development is still not necessarily easy. Using pre-configured widgets still require users to understand the basics of the data they manipulate, and conceptual programmatic flow. In other words, one does not have to be a rocket scientist, but one still has to be a logical thinker¹¹.
3. Inertia. Businesses are typically slow-moving (like molasses), and will not typically throw their full weight behind something unless some other business has had success with it too. Online blogger Shaw states it this way: *“Remember when the web started to grow? At first it was full of sites with pretty pictures and cool graphics. Organizations created websites as experiments or as another avenue for advertising. It wasn’t until the web killer app came along, eCommerce if you were wondering, that we had the dot-com explosion. We can repeat this story with SaaS and Salesforce.com. When the business sees a killer app, the business wants the killer app. Once we find the Salesforce.com equivalent for mashups, we’ll have the business lining up to invest. Why hasn’t this happened yet? Because we’re too busy talking about how cool mashups are. While cool is cool, it isn’t a killer app until it solves a business problem. We can take maps, charts and videos, we can pull in data from multiple sources and we can mash them together at the glass into a visually exciting experience for the mashup user, but...if the data aren’t actionable, what’s the point? If I can’t reuse the business logic across the organization, then why invest?”¹²*

In summary, mashups are appealing for business use, but come must be managed like any other corporate process or asset to prevent ad-hoc mahem within the enterprise, and must be perceived as a valuable endeavor to move beyond “business niche”.

¹⁰ Simmens et. al, pg 1172

¹¹ Tuchinda et al, pg 139

¹² Shaw, web log

Examples

At IBM, there are a number of mashup applications available to (and from) employees. Besides the requisite “Travelocity-like” travel website that mashes mapping and rental-car data for a virtual trip advisor, there is a full-blown Situational Application Environment by which employees can share mashups.

Of course, there remain the obvious quasi-business mashups like Yahoo Finance and informational/geospatial mashups like the one at the Chicago Crimerate site.

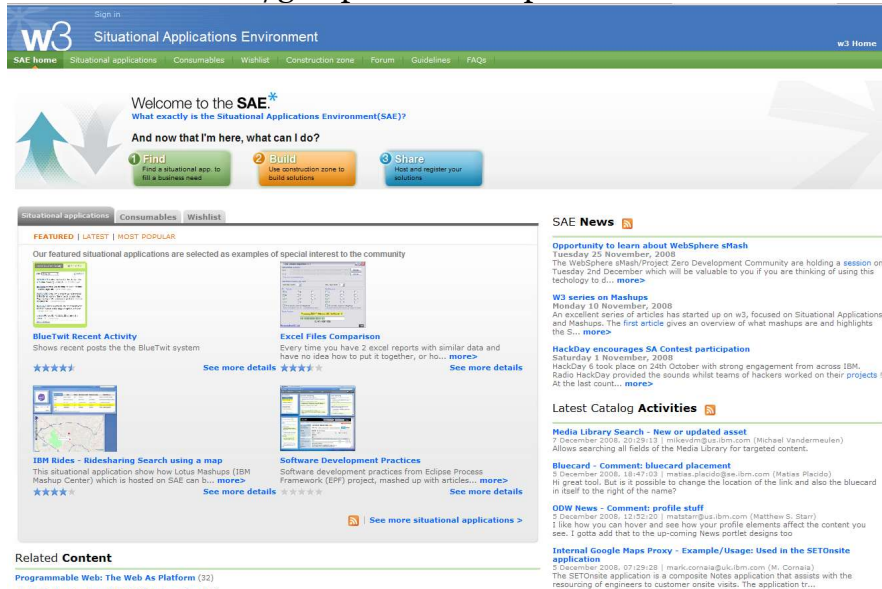


Figure 5: IBM's Situational Application Environment

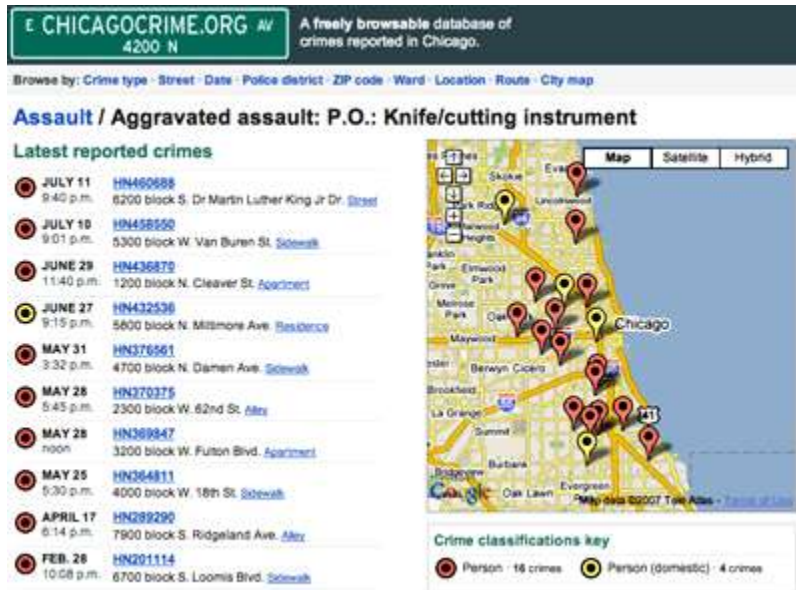


Figure 6: Chicago Crimerate Mashup: <http://chicago.everyblock.com/crime/>

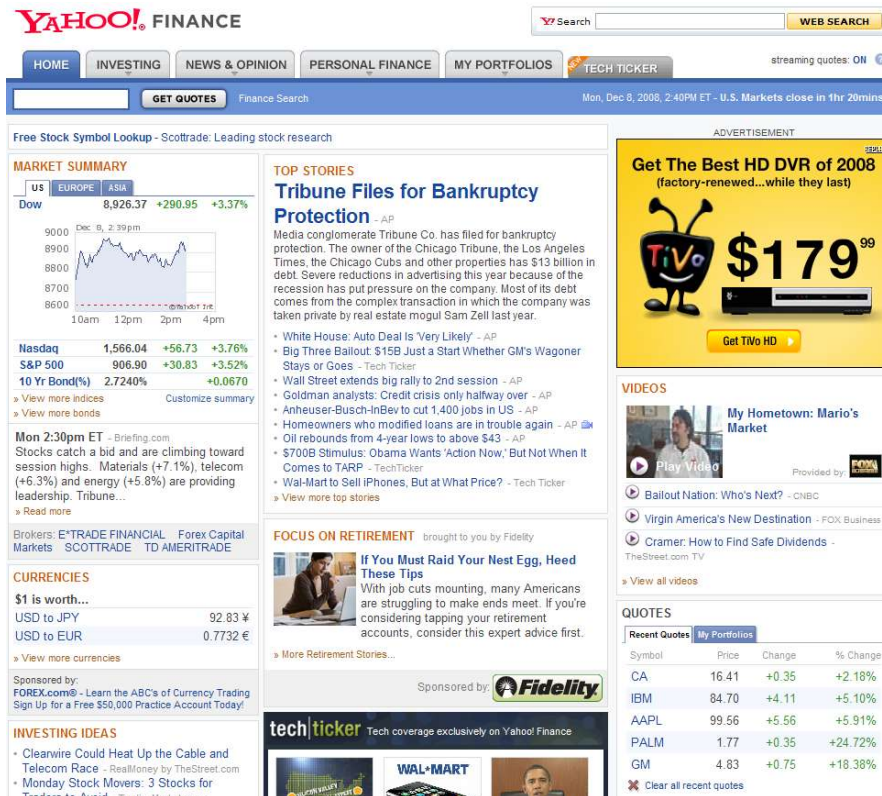


Figure 7: Yahoo Finance

Conclusion

This paper examined Mashups, first surveying the technology for general purpose situational application use, then as an enabler for business benefit. The notion of business driven development has been promoted for a number of years, but truly putting enterprise-level development capabilities in the hands of business users for situational applications has often proven difficult, if not impossible. Further, the solution that was partially successful: traditional desktop office tooling like Word and Excel, suffered from lack of governance, lack of scalability, and redundancy. Mashups for business finally provide the ability to rapidly compose business applications while utilizing existing enterprise assets. Further, with appropriate governance put in place, a coherent migration path could be applied to move situational applications to fully maintained enterprise applications managed by I/T organizations on a case-by-case basis.

Companies like IBM are already beginning to promote mashups internally for addressing business problems. While not completely there yet, in the coming years, one can imagine mashups are being positioned in a number of companies as enabling technology for business benefit and/or competitive advantage. Finally, the adoption of mashups is a blueprint for summarizing the success of any technology. When slow moving companies adopt (and see value in) approaches that techies, consumers or hackers have used and generated buzz with, that approach has “arrived” in terms of success. To that end, mashups for business are circling the airport; they haven’t fully arrived yet, but their arrival is imminent.

References

Damia: Data Mashups for Intranet Applications, David E. Simmen, Mehmet Altinel, Volker Markl, Sriram Padmanabhan, Ashutosh Singh, SIGMOD'08, June 9–12, 2008, Vancouver, BC, Canada., Copyright 2008 ACM
978-1-60558-102-6/08/06

An Office Document Mashup for Document-Centric Business Processes, John M. Boyer, DocEng'08, September 16–19, 2008, São Paulo, Brazil.,
ACM 978-1-60558-081-4/08/09

Mash-O-Matic, Sudarshan Murthy, David Maier, Lois Delcambre
DocEng'06, October 10–13, 2006, Amsterdam, The Netherlands.,
Copyright 2006 ACM 1-59593-515-0/06/0010

Services Mashups: The New Generation of Web Applications, Djamel Benslimane, Schahram Dustdar, Amit Sheth, IEEE Internet Computing,
SEPTEMBER/OCTOBER 2008 1089-7801/08 © 2008 IEEE

Story Mashup: Design and Evaluation of Novel Interactive Storytelling Game for Mobile and Web Users, Jürgen Scheible, Ville H. Tuulos, Timo Ojala,
MUM'07, December 12-14, 2007, Oulu, Finland.
Copyright 2007 ACM 978-1-59593-916-6/07/0012

Understanding Web 2.0, San Murugesan, IT Pro July August 2007 Published by
the IEEE Computer Society 1520-9202/07/ © 2007

MashupAdvisor: A Recommendation Tool for Mashup Development, Hazem Elmeleegy, Anca Ivan, Rama Akkiraju, Richard Goodwin, 2008 IEEE
International Conference on Web Services, 978-0-7695-3310-0/08 © 2008
IEEE, DOI 10.1109/ICWS.2008.128

Mashups, Wikipedia.org

Web Mashup Scripting Language, Marwan Sabbouh, Jeff Higginson, Salim Semy, Danny Gagne, WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005

Google Mashup Editor, Google Corp, <http://code.google.com/gme/>

Yahoo Pipes, Yahoo Corp, <http://pipes.yahoo.com/pipes/>

Lotus Mashups, IBM Corp, <https://greenhouse.lotus.com/home/login.jsp>

What's in a mashup? And why? Studying the perceptions of web-active end users, Nan Zang and Mary Beth Rosson, 2008 IEEE Symposium on Visual

Languages and Human-Centric Computing (VL/HCC), 1-4244-2528-0/08
©2008 IEEE

Building Mashups By Example, Rattapoom Tuchinda, Pedro Szekely, and Craig A. Knoblock, IUI'08, January 13-16, 2008, Maspalomas, Gran Canaria, Spain, Copyright 2008 ACM 978-1-59593-987-6/ 08/ 0001

Business Mashups - mashup applications for and by the business, Shaw, <http://businessmashup.blogspot.com/2007/10/how-close-are-we-to-overcoming-10.html>