

# Mashup (web application hybrid)

From Wikipedia, the free encyclopedia

A **mashup**, in web development, is a web page, or web application, that uses and combines data, presentation or functionality from two or more sources to create new services. The term implies easy, fast integration, frequently using open application programming interfaces (API) and data sources to produce enriched results that were not necessarily the original reason for producing the raw source data.

The main characteristics of a mashup are combination, visualization, and aggregation. It is important to make existing data more useful, for personal and professional use. To be able to permanently access the data of other services, mashups are generally client applications or hosted online.

In the past years, more and more Web applications have published APIs that enable software developers to easily integrate data and functions instead of building them by themselves. Mashups can be considered to have an active role in the evolution of social software and Web 2.0. Mashup composition tools are usually simple enough to be used by end-users. They generally do not require programming skills and rather support visual wiring of GUI widgets, services and components together. Therefore, these tools contribute to a new vision of the Web, where users are able to contribute.

## Contents

- 1 History
- 2 Types of mashup
  - 2.1 By API type
    - 2.1.1 Data types
    - 2.1.2 Functions
- 3 Mashup enabler
  - 3.1 History
  - 3.2 Web resources
- 4 Data integration challenges
  - 4.1 Text–data mismatch
  - 4.2 Object identity and separate schemata
  - 4.3 Abstraction levels
  - 4.4 Data quality
- 5 Mashups versus portals
- 6 Business mashups
- 7 Architectural aspects of mashups
- 8 See also
- 9 Notes
- 10 References
- 11 External links

## History

The history of mashup can be backtracked by first understanding the broader context of the history of the Web. For Web 1.0 business model, companies stored consumer data on portals and updated them regularly. They controlled all the consumer data, and the consumer had to use their products and services to get the information.

With the advent of Web 2.0 a new proposition was created, using Web standards that were commonly and widely adopted across traditional competitors and unlocked the consumer data. At the same time, mashups emerged allowing mixing and matching competitor's API to create new services.

The term isn't formally defined by any standard-setting body.<sup>[1]</sup>

The first mashups used mapping services or photo services to combine these services with data of any kind and therefore create visualizations of the data.<sup>[2]</sup> In the beginning, most mashups were consumer-based, but recently the mashup is to be seen as an interesting concept useful also to enterprises. Business mashups can combine existing internal data with external services to create new views on the data.

Mashups are in the ascendant. As a statistic from Programmable Web found out in 2009 that three new mashups have been registered every single day for the last two years.<sup>[3]</sup>

## Types of mashup

There are many types of mashup, such as business mashups, consumer mashups, and data mashups.<sup>[4]</sup> The most common type of mashup is the consumer mashup, aimed at the general public.

- *Business (or enterprise) mashups* define applications that combine their own resources, application and data, with other external Web services.<sup>[2]</sup> They 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 (or customer proxy, typically a product manager) for defining and implementing the business requirements. Enterprise mashups are secure, visually rich Web applications that expose actionable information from diverse internal and external information sources.
- *Consumer mashups* combines data from multiple public sources in the browser and organizes it through a simple browser user interface.<sup>[5]</sup> (e.g.: *Wikipediavision combines Google Map and a Wikipedia API*)
- *Data mashups*, opposite to the consumer mashups, combine similar types of media and information from multiple sources into a single representation. The combination of all these resources create a new and distinct Web service that was not originally provided by either source.

## By API type

Mashups can also be categorized by the basic API type they use but any of these can be combined with each other or embedded into other applications.

## Data types

- Indexed data (documents, weblogs, images, videos, shopping articles, jobs ...) used by metasearch engines
- Cartographic and geographic data: geolocation software, geovisualization
- Feeds, podcasts: news aggregators

## Functions

- Data converters: language translators, speech processing, URL shorteners...
- Communication: email, instant messaging, notification...
- Visual data rendering: information visualization, diagrams
- Security related: electronic payment systems, ID identification...
- Editors

## Mashup enabler

In technology, a **mashup enabler** is a tool for transforming incompatible IT resources into a form that allows them to be easily combined, in order to create a mashup. Mashup enablers allow powerful techniques and tools (such as mashup platforms) for combining data and services to be applied to new kinds of resources. An example of a mashup enabler is a tool for creating an RSS feed from a spreadsheet (which cannot easily be used to create a mashup). Many mashup editors include mashup enablers, for example, Presto Mashup Connectors (<http://www.jackbe.com/products/connectors.php>), Convertigo Web Integrator (<http://www.convertigo.com/en/overview/features/web.html>) or Caspio Bridge.

Mashup enablers have also been described as "the service and tool providers, that make mashups possible".

## History

Early mashups were developed manually by enthusiastic programmers. However, as mashups became more popular, companies began creating platforms for building mashups, which allow designers to visually construct mashups by connecting together mashup components.

Mashup editors have greatly simplified the creation of mashups, significantly increasing the productivity of mashup developers and even opening mashup development to end-users and non-IT experts. Standard components and connectors enable designers to combine mashup resources in all sorts of complex ways with ease. Mashup platforms, however, have done little to broaden the scope of resources accessible by mashups and have not freed mashups from their reliance on well-structured data and open libraries (RSS feeds and public APIs).

Mashup enablers evolved to address this problem, providing the ability to convert other kinds of data and services into mashable resources.

## Web resources

Of course, not all valuable data is located within organizations. In fact, the most valuable information for business intelligence and decision support is often external to the organisation. With the emergence of rich internet applications and online Web portals, a wide range of business-critical processes (such as ordering) are becoming available online. Unfortunately, very few of these data sources syndicate content in RSS format and very few of these services provide publicly accessible APIs. Mashup editors therefore solve this problem by providing enablers or connectors.

## Data integration challenges

There are a number of challenges to address when integrating data from different sources. The challenges can be classified into four groups: text/data mismatch, object identifiers and schema mismatch, abstraction level mismatch, data accuracy.<sup>[6]</sup>

## **Text–data mismatch**

A large portion of data is described in text. Human language is often ambiguous - the same company might be referred to in several variations (e.g. IBM, International Business Machines, and Big Blue). The ambiguity makes cross-linking with structured data difficult. In addition, data expressed in human language is difficult to process via software programs. One of the functions of a data integration system is to overcome the mismatch between documents and data.<sup>[6]</sup>

## **Object identity and separate schemata**

Structured data are available in a plethora of formats. Lifting the data to a common data format is thus the first step. But even if all data is available in a common format, in practice sources differ in how they state what is essentially the same fact. The differences exist both on the level of individual objects and the schema level. As an example for a mismatch on the object level, consider the following: the SEC uses a so-called Central Index Key (CIK) to identify people (CEOs, CFOs), companies, and financial instruments while other sources, such as DBpedia (a structured data version of Wikipedia), use URIs to identify entities. In addition, each source typically uses its own schema and idiosyncrasies for stating what is essentially the same fact. Thus, Methods have to be in place for reconciling different representations of objects and schemata.

## **Abstraction levels**

Data sources provide data at incompatible levels of abstraction or classify their data according to taxonomies pertinent to a certain sector. Since data is being published at different levels of abstraction (e.g. person, company, country, or sector), data aggregated for the individual viewpoint may not match data e.g. from statistical offices. Also, there are differences in geographic aggregation (e.g. region data from one source and country-level data from another). A related issue is the use of local currencies (USD vs. EUR) which have to be reconciled in order to make data from disparate sources comparable and amenable for analysis.

## **Data quality**

Data quality is a general challenge when automatically integrating data from autonomous sources. In an open environment the data aggregator has little to no influence on the data publisher. Data is often erroneous, and combining data often aggravates the problem. Especially when performing reasoning (automatically inferring new data from existing data), erroneous data has potentially devastating impact on the overall quality of the resulting dataset. Hence, a challenge is how data publishers can coordinate in order to fix problems in the data or blacklist sites which do not provide reliable data. Methods and techniques are needed to; check integrity, accuracy, highlight, identify and sanity check, corroborating evidence; assess the probability that a given statement is true, equate weight differences between market sectors or companies; act as clearing houses for raising and settling disputes between competing (and possibly conflicting) data providers and interact with messy erroneous Web data of potentially dubious provenance and quality. In summary, errors in signage, amounts, labeling, and classification can seriously impede the utility of systems operating over such data.

# Mashups versus portals

Mashups and portals are both content aggregation technologies. Portals are an older technology designed as an extension to traditional dynamic Web applications, in which the process of converting data content into marked-up Web pages is split into two phases: generation of markup "fragments" and aggregation of the fragments into pages. Each markup fragment is generated by a "portlet", and the portal combines them into a single Web page. Portlets may be hosted locally on the portal server or remotely on a separate server.

Portal technology defines a complete event model covering reads and updates. A request for an aggregate page on a portal is translated into individual read operations on all the portlets that form the page ("render" operations on local, JSR 168 portlets or "getMarkup" operations on remote, WSRP portlets). If a submit button is pressed on any portlet on a portal page, it is translated into an update operation on that portlet alone (`processAction` on a local portlet or `performBlockingInteraction` on a remote, WSRP portlet). The update is then immediately followed by a read on *all* portlets on the page.

Portal technology is about server-side, presentation-tier aggregation. It cannot be used to drive more robust forms of application integration such as two-phase commit.

Mashups differ from portals in the following respects:

	<b>Portal</b>	<b>Mashup</b>
<b>Classification</b>	Older technology, extension to traditional Web server model using well-defined approach	Using newer, loosely defined "Web 2.0" techniques
<b>Philosophy/approach</b>	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
<b>Content dependencies</b>	Aggregates presentation-oriented markup fragments (HTML, WML, VoiceXML, etc.)	Can operate on pure XML content and also on presentation-oriented content (e.g., HTML)
<b>Location dependencies</b>	Traditionally, content aggregation takes place on the server	Content aggregation can take place either on the server or on the client
<b>Aggregation style</b>	"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
<b>Event model</b>	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
<b>Relevant standards</b>	Portlet behavior 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 such as EMMML are emerging.

The portal model has been around longer and has had greater investment and product research. Portal technology is therefore more standardized and mature. Over time, increasing maturity and standardization of mashup technology will likely make it more popular than portal technology because it is more closely associated with Web 2.0 and lately Service-oriented Architectures (SOA).<sup>[7]</sup> New versions of portal products are expected to eventually add mashup support while still supporting legacy portlet applications. Mashup technologies, in contrast, are not expected to provide support for portal standards.

## **Business mashups**

Mashup uses are expanding in the business environment. Business mashups are useful for integrating business and data services, as business mashups technologies provide the ability to develop new integrated services quickly, to combine internal services with external or personalized information, and to make these services tangible to the business user through user-friendly Web browser interfaces.<sup>[8]</sup>

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 (SaaS) offering.

Many of the providers of business mashups technologies have added SOA features.

## **Architectural aspects of mashups**

The architecture of a mashup is divided into three layers:

- Presentation / user interaction: this is the user interface of mashups. The technologies used are HTML/XHTML, CSS, Javascript, Asynchronous Javascript and Xml (Ajax).
- Web Services: the products functionality can be accessed using the API services. The technologies used are XMLHttpRequest, XML-RPC, JSON-RPC, SOAP, REST.
- Data: handling the data like sending, storing and receiving. The technologies used are XML, JSON, KML.

Architecturally, there are two styles of mashups: Web-based and server-based. Whereas 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.<sup>[9]</sup>

Mashups appear to be a variation of a façade pattern.<sup>[10]</sup> That is: a software engineering design pattern that provides a simplified interface to a larger body of code (in this case the code to aggregate the 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 them available as discrete Web services. Web services provide open, standardized protocols to provide a unified means of accessing information from a diverse set of platforms (operating systems, programming languages, applications). These Web services can be reused to provide completely new services and applications within and across organizations, providing business flexibility.

## See also

- Open Mashup Alliance
- Web scraping

## Notes

1. ^ "Enterprise Mashups: The New Face of Your SOA" (<http://soa.sys-con.com/node/719917>). <http://soa.sys-con.com/>: SOA WORLD MAGAZINE. Retrieved 2010-03-03. "The term mashup isn't subject to formal definition by any standards-setting body."
2. ^ *a b* Holmes, Josh. "Enterprise Mashups" (<http://msdn.microsoft.com/en-us/architecture/bb906060.aspx>). *MSDN Architecture Journal*. MSDN Architecture Center.
3. ^ "Enterprise Mashups: The New Face of Your SOA" (<http://soa.sys-con.com/node/719917>). <http://soa.sys-con.com/>: SOA WORLD MAGAZINE. Retrieved 2010-03-03. "One popular mashup site, *Programmable Web*, reports that three new mashups have been registered every single day for the last two years."
4. ^ Sunilkumar Peenikal (2009). "Mashups and the enterprise" ([http://www.mphasis.com/pdfs/Mashups\\_and\\_the\\_Enterprise.pdf](http://www.mphasis.com/pdfs/Mashups_and_the_Enterprise.pdf)). Mphasis - HP.
5. ^ "Enterprise Mashups: The New Face of Your SOA" (<http://soa.sys-con.com/node/719917>). <http://soa.sys-con.com/>: SOA WORLD MAGAZINE. Retrieved 2010-03-03. "A consumer mashup is an application that combines data from multiple public sources in the browser and organizes it through a simple browser user interface."
6. ^ *a b* E. Curry, A. Harth, and S. O'Riain, "Challenges Ahead for Converging Financial Data," (<http://sw.deriv.org/2009/09/financial-data/>) in Proceedings of the XBRL/W3C Workshop on Improving Access to Financial Data on the Web, 2009.
7. ^ Digna, Larry (2007). "Gartner: The future of portals is mashups, SOA, more aggregation" (<http://blogs.zdnet.com/BTL/?p=4912>). ZDNET.
8. ^ Holt, Adams (2009). "Executive IT Architect, Mashup business scenarios and patterns" (<http://www.ibm.com/developerworks/lotus/library/mashups-patterns-pt1/>). IBM DeveloperWorks.
9. ^ Bolim, Michael (2005). "End-User Programming for the Web, MIT MS thesis, 2.91 MB PDF" ([http://bolinfest.com/Michael\\_Bolin\\_Thesis\\_Chickenfoot.pdf](http://bolinfest.com/Michael_Bolin_Thesis_Chickenfoot.pdf)). pp. 22–23.
10. ^ Design Patterns: Elements of Resuable Object-Oriented Software (ISBN 0-201-63361-2) by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides

## References

- Ahmet Soylu, Felix Mödritscher, Fridolin Wild, Patrick De Causmaecker, Piet Desmet. 2012 . "Mashups by Orchestration and Widget-based Personal Environments: Key Challenges, Solution Strategies, and an Application." ([http://www.ahmetsoylu.com/wp-content/uploads/2011/05/Program\\_Soylu\\_et\\_al\\_2012.pdf](http://www.ahmetsoylu.com/wp-content/uploads/2011/05/Program_Soylu_et_al_2012.pdf)) Program: Electronic Library and Information Systems 46 (4): 383–428.

## External links

- Why Mashups = (REST + 'Traditional SOA') \* Web 2.0 (<http://blog.sherifmansour.com/?p=187>)
- Mashups Part I: Bringing SOA to the People (<http://www.soamag.com/I18/0508-1.asp>)
- Mashups Part II: Why SOA Architects Should Care (<http://www.soamag.com/I21/0808-1.asp>)
- A Mashup with Google Maps and Youtube (<http://my-bilingual.com/maps>)

Retrieved from "http://en.wikipedia.org/w/index.php?title=Mashup\_(web\_application\_hybrid)&oldid=557697624"

Categories: Mashup (web application hybrid) | Software architecture | Web 2.0 | Web 2.0 neologisms

| Web development | World Wide Web

---

- This page was last modified on 31 May 2013 at 15:32.
  - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy.
- Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.