

MASHUP

MASHUP

In technology, a mashup is a web application that combines data from more than one source into a single integrated tool; an example is the use of cartographic data from Google Maps to add location information to real-estate data, thereby creating a new and distinct web service that was not originally provided by either source.

Mashup originally referred to the practice in pop music (notably hip-hop) of producing a new song by mixing two or more existing pieces.

Overview

Content used in mashups is typically sourced from a third party via a public interface or API (web services), although some in the community believe that cases where private interfaces are used should not count as mashups [citation needed]. Other methods of sourcing content for mashups include Web feeds (e.g. RSS or Atom), and screen scraping. Many people are experimenting with mashups using Amazon, eBay, Flickr, Google, Microsoft, Yahoo and YouTube APIs, which has led to the creation of the mashup editor.

The architecture of Mashup web applications is always composed of three parts:

- The content provider: it is the source of the data. Data is made available using an API and different Web-protocols such as RSS, REST, and Web Service
- The Mashup site: is the web application which provides the new service using different data sources that are not owned by it.
- The client web browser: is the user interface of the Mashup. In a web-application, the content can be mashed by the client web browsers using client side web language for example JavaScript, kolwole.

Quotes

A lot of talk about Web 2.0, web mashups, Ajax etc., which in my mind are all facets of the same phenomenon: that information and presentation are being separated in ways that allow for novel forms of reuse. —Sho Kuwamoto

The mash-up part of this equation, is the offspring of an environment where application developers see it in their own selfish interest to facilitate the creation of integrated, yet highly derivative application hybrids by third parties, something they do by providing rich public APIs to their user base. —Mark Sigal

(Popfly is) a (mashup) tool for end users, not necessarily code heads. —Steve Ballmer

Mashup was a good word for perhaps 2003-2007, but it implies limitations and barriers that simply no longer exist. —Dave Troy

Types of mashups

Mashups currently come in three general flavors: consumer mashups, data mashups, and business mashups.

The best known type is the consumer mashup, best exemplified by the many Google Maps applications. Consumer mashups combine data elements from multiple sources, hiding this behind a simple unified graphical interface.

Other common types are "data mashups" and "enterprise mashups". A data mashup mixes data of similar types from different sources (see Yahoo Pipes), as for example combining the data from multiple RSS feeds into a single feed with a graphical front end. An enterprise mashup (see JackBe), usually integrates data from internal and external sources - for example, it could create a market share report by combining an external list of all houses sold in the last week with internal data about which houses one agency sold.

MASHUP

A business mashup is a combination of all the above, focusing on both data aggregation and presentation, and additionally adding collaborative functionality, making the end result suitable for use as a business application.

Mashups within mashups are known as 'monster mashups'.

Examples

Mashups can be divided into four main categories, mapping, video & photo, search & shopping and news.

Some examples of each are:

Mapping

Chicago Crime

The Chicago Police Department has a mashup which integrates the department's database of reported crimes with Google Maps in order to help stop crime in areas and warn citizens of areas where the crime rate is high.

Video and Photo

Flickr

Flickr is an image storage site that allows users to organize their collection of images and share them. Through the use of its Application Programming Interface (API) the content can be used by other sites to create a mashup. Flickrvision is an example of a mashup made using Flickr's API.

Search and Shopping

Travature

Travature is a travel portal which has integrated airfare meta search engines, wiki travel guides, hotel reviews. It also allows the user to share photos and discuss experiences with other travellers.

News

Digg

Mashup of various news websites controlled almost entirely by the users of the website.

Telecom mashups

A "telecom mashup" is a telecommunications service where service elements come from more than one source and are combined into an integrated experience. For example, one could get the base service from company A, a ringback tone from company B, a voicemail service from company C, etc. Microsoft's Connected Services Sandbox is an example of a "service development environment" (also referred to as a "service development platform" or a "service development framework") that enables developers to more easily create telecom mashups.

Nested, embedded, sequential or recursive telecom mashups, where a large number of services are used, are considered risky or "over-the-top" (OTT) by network operators. Because of their complexity, such mashed up services can be difficult to control and charge, and therefore may cause operators to lose revenues. In addition, operators are subject to

MASHUP

government fines or customer litigation from lax or overzealous conformance to Lawfully Authorized Electronic Surveillance (LAES) in the management of OTT mashups.

The following example of an over-the-top mashup is found in the online version of Pipeline magazine: "an address book lookup gives a name which is used to find a person's location which brings up a map of reasonable trip routing, resulting in a call, once connected, linking in web co-browsing, with a location, type search to find for a restaurant along the trip route, clicking the restaurant's advertisement, resulting in a reservation, and a message to social-network shared friends to met [sic] them with directions inserted from that person's location."

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 of these markup fragments 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 another 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, that is translated into a single update operation on that portlet alone ("processAction" on a local, JSR 168 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 the one using Two_phase_commit.

Mashups differ from portals in the following respects:

Portal Mashup

Classification Older technology, extension to traditional web server model using well defined approach Using newer "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 contents in another way Content dependencies. Aggregates presentation-oriented markup fragments (HTML, WML, 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 standard is XML Data Interchange with REST semantics. RSS and Atom are commonly used. More specific mashup standards are expected to emerge.

The portal model has been around longer and has seen greater investment and product research. The technology is therefore more standardised and mature. In 2-3 years, increasing maturity and standardisation of mashup technology may propel it beyond portal technology in popularity. Alternatively, the two technologies may converge, with newer versions of current portal products resembling mashup servers using mashup techniques while still supporting legacy portlet applications.

MASHUP

Mashups in a business environment

Mashups can have a big role in the business environment. This section will discuss a few of the practical uses that could be made of this technology; with specific reference to, the end-user desire for personalized information services, addressing information overload issues, and the need for technology integration.

Mashups create new services for consumers which allow them to create a lot of new possibilities. It is becoming more straightforward for end-users to create mashup sites regardless of their technical skill level by using a site's Application Programming Interface (API). This provides a simple way to implement mashups and new consumers can create a mashup easily. With all APIs available, enterprises can easily build applications that are cheaper, more reusable and more maintainable. The web 2.0 is all about the personalization of information. In that context, mashups can grow faster since every mashup can offer new features from existing websites, in fact Emory M. Craig (2007) states that on Programmableweb.com around three new mashup sites are added every day. It's easy to imagine how you can create a personalized information service when you have the ability to add features with mashups. As they are a part of the web 2.0, mashups increase the level of personalization of the Internet. Many people are convinced in the power of mashups and view mashups as the future of the web.

Information overload is a problem that many companies face, whether it's within the company itself or when presenting information to users. Mashups can help overcome this problem by allowing one company to display detailed concise information derived from a wide variety of sources. For example, prior to mashup sites, users that wanted to find images of a particular type (e.g. animals) would first have to search for websites and then trawl through them trying to find suitable images. There are obviously a vast number of web sites that could potentially have suitable images on them; therefore the user would be presented with an overload of information. To help solve this problem, a mashup site could be created that pulls in the images from the other sites and categorizes them. This would enable the user to easily search in one location for what they want, and even allow them to see all possible images within a particular category in a few clicks.

Mashups are a form of technology integration as they can be adapted to join together many technologies and use many types of implementation languages. Some mash-ups can be as simple as mixing together some JavaScript code with XML to create a new and innovative web-based service. Other larger mashups which are the main focus of their respective website will take advantage of services such as Google Maps and a database which uses street addresses to link the two together and project information onto the map.

Architectural aspects of mashups

For all appearances it would seem that a mashup is an example of a Facade pattern (see Design Patterns: Elements of Reusable Object-Oriented Software (ISBN 0-201-63361-2) by the GoF, Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides). I.e. it 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 API's).

Mashup editors

There are already several mashup platforms that help user create mashups. Examples include (alphabetically):

- a) Google Mashup Editor
- b) Microsoft Pop fly
- c) Yahoo pipes