Mashup Concept and Challenges

Ms. Ruchita Joshi and Mr. Rujul Trivedi

[1] Venus International College of Technology, Gandhinagar, Gujarat 382420, India
ruchita.joshi18388@gmail.com

[2] RMIT University, Melbourne, Victoria, 3001, Australia
rujul04@gmail.com

Abstract

In today's fast growing technical world, mashup created its own space in users' mind. The basic idea behind mashup is to collect web data from multiple sources and using mashup application and client application, collaborating them in such a way that satisfy the user needs. With lots of entertainment and faster world of mashup, we have to face certain challenges to make it success. Some of those challenges do have solutions and some of them are still on research.

Keywords: *architecture; challenges; functionality; mashup types; outcome.*

1. Introduction

We can define Mashup in several different ways. In general, "Mashups" are web applications that combine information or data from two or more sources into one interface. For example, plotting the top 25 best companies to work for onto a map is considered a mashup. Another example could be cross referencing CNN news feeds with Wikipedia [1]

By combining existing web resources with the concept of reusability, we can create our own mashup to satisfy user's need. "Programmableweb.com" can help you to see the list of existing mashups and available APIs for the same. The approx. numbers are 10120 APIs and 7225 Mashups. With the help of these tools and APIs, any non-programmer individual can register his own mashup simply by drags and drops. We can create mashup applications with the help of technologies named, RDF (Resource Description Framework), OWL (Web Ontology Language) and RDFS (RDF Schema). RDF is a developers' language and not of end-users' because of its complex syntax.

2. TYPES OF MASHUP

Depending on the sort of assets being combined or integrated, there are three categories of mashup:

A. Presentation mashup:

It does not consider the web sources' data and application functionality. It focuses the way of fetching the layout and information from different resources. It concentrates on visualization effects than the data to be carried with the model. Logic is not a part of this phase. [1]

B. Data mashup:

Different merging techniques such as string similarity, structure similarity, sub-tree matching method, nested node list method, linked list logic and many more; can be used to merge data from several web sources into one combined content page. It concentrates on the actual data to be carried over the web and not on the layouts. Data Mashup is the way of representing data as per the user's need. It only fetches that information which is requested by the end user. [1]

C. Functionality mashup:

It combines both the underlying data and application functionality to a new service. It is accessible via APIs. There are number of different APIs available for novice users that guides them to access mashup data. User interface is most important part of functionality mashup. Data are displayed based on user's need and interest area in such a way that catches more user eyes. [1]

3. ARCHITECTURE

There are many defined architectures for mashups do exist but by considering the basics, we can have below shown architecture.

A. Data

Data are considered as a core element of mashup for aggregation and presentation to the user. Data can be stored in any database i.e. local or internet based. Data may come from the web services that are XML/ JSON serialized. Type of database vary depends on the retrieval requirement. [6]

B. RSS Feeds

Really Simple Syndication (RSS) is XML documents. It is understood and well documented. As with the increasing number of social sites, there are number of extensions of it like licensing information, add new attachments to the existing feeds, location data and more. [6]

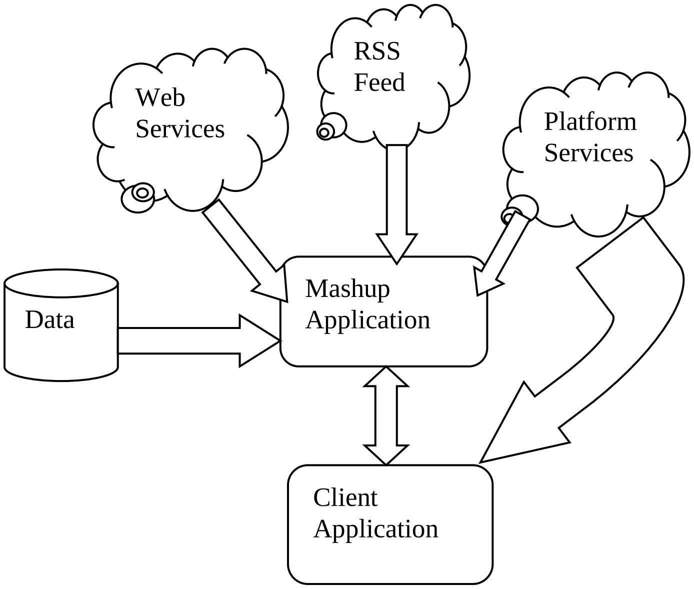

Fig. 1 Mashup Architecture. [6]

C. Web and Platform Services

Web services are useful in terms of providing addition data or to transform the data carried out over the web. It can be a REST-based or WSDL-based service. Along with this, platform services are basically used to provide functionality beyond the request/ response model of traditional web services. [6]

D. Mashup Applications

Various tools and APIs are available for its creation. It can be written using different web technologies such as ASP.Net or PHP but it also provides a connection between the server and the client in terms of some Rich Internet Applications. It does not require having installation on client side. [6]

E. Client Applications

It manages the task of how to deliver and represent the mashup to the user. Web browser can be considered as the most common client application which receives data from the server over HTTP. It can also be used for rich visualization and mashup processing on client side. [6]

4. CHALLENGES

A new type of applications has taken advantage of the new data and services available by mixing them, in order to generate new applications fast and efficiently [3]. As we know, even non-programming person can create his own mashup application just by drag and drop operations on a canvas. But mashup has to face several challenges related to security, versioning, reusability and more.

A. Cataloguing

User, who has created a mashup, can make it available for others. But there should be a way to maintain a list of inserted mashups on the web, called directory. That should catalogue mashups in a consistent way. If there are number of mashups available with the same name and concept then it simply wastes the storage on the web. Also the user will get confuse about having multiple occurrences for the same. [5]

B. Making data web enabled

As we know, there are some types of mashup. In some of the mashups, data is capsulated with the layout and such data cannot be reused. Thus, some tools are required to decouple these data and layout for reusability. Reusability is the main and basic aim of mashup where users can innocently copy the existing data available on the web and give a touch to his own mashup. [5]

C. Security & Identity

These aspects are needed when our mashup contains some confidential data. Even in a company, if some documents must be kept confidential, only privileged users are allowed to access or read the document. Same kind of security and identity is needed in the case of mashups. [5]

D. Sharing & Reusing

As discussed earlier, even non- programmers can be a mashup creator. For this, there must be efficient search functionality & light-weight formats for an easy to use purpose. Once created, used by any and many. Person can create own mashup for some good conceptual requirement and then can register it with his name. He can share it with friends, family members or with any by clicking the sharable option available at the time of registration. Once made sharable, any uses can use those existing information for fulfilling their purpose and even can change it further. Owner cannot secure only some part of the mashup. It can either be sharable or not at all. [5]

E. Minimizing the live response time

Because of variety of factors, end users may face execution delay. This factor totally depends on how well the tool gets compiled and executed. When user fires a query to retrieve information from web, response time matters. Tool should allow user to interact with the interface while executing the previous request. Also make sure about the internet connection errors if any. As soon as the data is loaded, show the user interface on immediate basis. [4]

F. Concurrent access

How can we support concurrent access of multiple users to a same view of a mashup application? Concurrent in this case means that all users work together on a same view of the mashup. For instance, we can imagine a mashup in which the employee co-browses his options for the leave request together with his manager and they jointly decide when it is fine for both.

G. Role-based access

How can we support cooperative access of multiple users to a mashup application via different views on the application? The goal here is to allow multiple users to cooperate, but by confronting each user with mashup information that is relevant for his/her role and activities only. As an example, the employee and the manager may agree on the leave request in such a way that the employee selects his configuration alone, and the manager then has another view just to accept or reject the proposal.

H. Input for Mashup creator environment

This challenge is generally faced by a designer. Applications support a broad variety of data types which produces diverse types of mashups. Data as input for the mashup comes with certain challenges. Those are, Data available on the web can be in unpredictable format, some of the data are required to be protected (that requires authentication to access them and data availability on the web. [3]

I. Output from mashup

Output can be in a format that can be consumed by other applications. It may possible that the produced output may contain some data which might prefer to have security. As user merges information from various sources, in enterprise mashup, confidential data might be used by unauthorized access. Such thing is not good for the organization that supports classification. [3]

References

[1] Nan Zang and Mary Beth Rosson, “in a mashup? And why? Studying the perceptions of web-active end users”, IEEE 2008
[2] Valerie B¨onstr¨om, Annika Hinze, Heinz Schweppe, “Storing RDF as a Graph”, 0-7695-2058-8/03, 2003 IEEE.
[3] Lampros Goussis, Ioannis E. Foukarakis, Dimitios N. Kallergis, “Aspects and challenges of mashup creator design”, 14th Panhellenic Conference on Informatics, 2010
[4] Saeed Aghaee, Cesare Pautasso, “Live Mashup Tools:Challenges and Opportunities”, -1-4673-6265-8/13/$31.00 c 201, IEEE.
[5] Agnes Koschmider, Victoria Torres, Vicente Pelechano, “Elucidating the Mashup Hype: Definition, Challenges”, Germany
[6] msdn.microsoft.com/en-us/library/bb906060

Ms. Ruchita Joshi has completed M.E. in I.T. from Gujarat Technological University in 2014. She had her Bachelor degree from Dharamsinh Desai Univesity, Nadiad in 2011. Currently, she is working as an assistant professor with Venus International College of Technology since last 4 years. She has published a book on Web Data Management in Oct, 2015.

Mr. Rujul Trivedi has completed his M.S. in I.T. from RMIT University, Melbourne, Australia in 2014. He had his Bachelor degree from Dharamsinh Desai University, Nadiad in 2011. Currently, he is working as a backend developer with an IT company in Melbourne since last 1 and a half years. He has published a book on Web Data Management in Oct, 2015.