ANALYSIS AND DEVELOPMENT OF A SERVICE MASH UP APPLICATION FOR MOBILE USERS

Dr. Atul M. Gonsai¹, Mr Rushi R. Raval²

¹Assistant Professor, Department of Computer Science, Saurashtra University, Rajkot
²Assistant Professor, Dept. of M.Sc. (IT&CA), GK&CK Bosamia College, Jetpur, (Affiliated with Saurashtra University)

ABSTRACT

Mash ups enables user to develop a new application based on a configuration of contents retrieved from external applications or services, and it is gaining drive as a means to develop requirement based Web applications by combining different resources such as web services, data feeds and user interfaces. But, current and future trends are for mobile or smart phone. smart-phone contains more personal data such as Contacts, audios, videos, and GPS based location information, with which user can create, unconventional and logical mash up applications for different areas such as internet marketing, social networking, Entertainment, and e-commerce. We develop a service mash up application for mobile clients which enables users to get information such as location based services, with using Google Map API v2 and android as a platform.

Keywords: Mash up, Web service, Google Map API v2, Yahoo pipes, Mobile App, Android Platform, Web API

I. INTRODUCTION

Current web development, a mash up is a web page or application that combines functionality and/or data from two or more external sources such as RSS feeds, Web services, content bound from third party web sites, or public APIs to create a new service or application. The term mash up implies fast integration, easy, more user friendly nature.

There have been significant improvements on Web-oriented programming technologies, such as JavaScript [1], and Silverlight [2]. With use of these programming technologies, numerous web services provide an open APIs (Application Programming Interfaces) to enable user-generated dynamic web applications or services. For e.g., Google maps provides open APIs, like as event
handling, location based management, object controlling and overlapping functions, to deploy objects on the map. These types of open APIs can enable a user to combine multiple useful sources from multiple web sites on a single web page. These types of web based applications are known as Web mash up application or mash up services.

Currently, the quantity of mobile mash up applications develops rapidly with firm mobile computing development. Present mobile phones sensing supports (e.g., Infrared, and barcode readers), and multi-media (e.g. Audio/Video playback and Camera), and communication part (e.g. EDGE, Wi-Fi, GPRS, Bluetooth, and UMTS) capabilities are model responsive mediators with which users can achieve information for mash up applications.

II. RELATED WORK

There are some actions to gather open API information to deliver possible mash up grouping to the users. For e.g. the mash up matrix [3] delivers the option of connectable mash up services to a user. However, this matrix just provides the 2-dimensional connectivity. If a user wants to combine multiple bases, several traverses are prerequisite to identify the feasibility of mash up services.

Another e.g., of a Mobile Mash up might be when people are sense interested about what exactly is happening on the other side of the world. How does it look like? What kind of pictures people take there? Or, at what time the sun rises? A Mobile mash up that isolates this location on Google Maps, displays sun-setting and sun-rising information from a meteorological service and shows user generated pictures taken from Flickr (provide picture galleries available with chat, social networking, photo ratings, and groups), can help them substantial this curiosity. The Mash up could use mobile's graphical location to get the information of the “antipodes”. When users; location changes, so does their antipodes and thus the information on the Mash up. Giving the nature of this Mash up, we will call it simply “Antipodes”.

III. CHALLENGE AND PROBLEM DEFINITION

Mobile users are referred as a “thin clients” [4] with limited processing power and with limited resources. The limitations are central to mobility issues and not just the inadequacies of current technology [5]. For ex., a service mash up involves parsing and combining different web services results requiring a lot of computation. The challenges are minimising the data processing task on mobile users and extending processing power beyond mobile users. Furthermore, many mobile platforms do not include necessary libraries for SOAP web services.

In general view, if mobile users want any kind of information they have to connect with internet or use application. In both cases mobile / smart-phone have suffered with limited processing power as well as with limited resources/functionality. Some other problems may include privacy. It may be likely that user specific modelling research in these areas will be appropriate to personalised mash ups.

IV. IDEA OF SERVICE MASH UP PLATFORM

Service mash up let mobile users to merge different services. Though, service mash up requires communication with web services and processing power. Reason of the resources limitation (energy consumption, processing power, s/w libraries) of mobile users, it is inept to do service mash up on the mobile users. The middle-ware provides a Personal Mash up Platform which does service mash up for the mobile users. The platform has broad interfaces for defining and consuming web services. However, the services are stored on the middle-ware and
can be connected to form a work-flow (a mash up service) which provides possibility to caching intermediate service results.

V. WEB SERVICE COMPOSITION AND MASH UP

Web services are mainly derived from the service-oriented architecture that is, based on Service-Oriented Computing (SOC). S.O.C. [6] is a computing model that consumes services as essential fundamentals for developing applications. In S.O.C., services are separate, platform in dependent computational objects that can be used in a platform independent method [7]; hence new services can be composited from existing services with low or free cost. There are presently two styles of combining web services, light-weighted web services mash up and the formal web services composition.

There are numerous approaches to web services composition such as Web Component [8], Semantic Web (OWL-S) [9] and BPEL (J) [10]. According to the review of Liu et al. [11], all of them familiarize robust outlays (developer skill and supporting infra-structure). Though, this research focuses on a light-weighted approach to service conformations, web services mash ups [12] which “typically serve an unambiguous situational necessity (short-live) and are composed of the latest, easy-to-use web technologies such as, RSS, Restful web service, and Atom feeds).”

One subclass of web services mash up is Enterprise Mash up (EM) [13] which is a, standard that “end-users are authorized to adapt their individual business to their individual and varied requests”. Figure: - 1, shows the two styles of E.M., wiring and piping. Hoyer and Fischer [14] also characterized mash up tools in the marketplace based on their functionality and target group.

- Widget is, a graphical edges which provides simple user interaction intellectualizing from the underlying resources. Wiring inter connects visually input and output parameters of widgets, which requires no programming skill at all.
- Resource is, the actual content, data, or application which rendering interface as Web services, web open API, and other. Piping adds resources to processing chain/graph by directing output of one resource to input of next resource.

Piping frequently contains Domain Specific Language (DSL) [16]. Maximilien, Ranabahu [17] intended an online platform for service mash up based on DSL. Users can create and share mash up services using web browsers interface. The core of the platform is a DSL Engine which can generate a Ruby on Rail application from the DSL code defined users. The DSL supports three essential functions in defining service mash up, data mediation, process/protocol mediation, and user interface customization.
The idea of mobile mash ups has gained popularity recently. Xu, Song [18] developed a mash up platform for mobile devices based on aspect-oriented programming technology. The key feature of the platform is the mash up management framework which monitors and controls mash up execution. Both execution status and performance is monitored and compared to expecting quality of service defined by Service Level Agreement (SLA). The adaptive engine then optimizes or resolves the quality of service problems, for e.g., replacing composed services.

Web service mash up shares certain advantages over the formal web service composition method, since a web service mash up requires less programming skills and overhead. Mash ups can be achieved by either wiring at the interface/widget level or piping at the resource/service level. The proposed mash up platform supports “piping” mash ups. The piping and QoS management is done on the middleware, but the interface for defining a mash up service is on mobile client.

VI. DEVELOPMENT OF A SERVICE MASH UP

On the mobile user side, the middleware has a user interface which lets users define a mash up services. The middleware has a service storage which stores user defined service data and an execution engine which performs web services and pipes input and output of web services. So as to support a service mash up, the middleware must first support consuming existing web services. Specific web service calls are pre-defined by users using the mobile client and stored in the service-storage for upcoming accomplishment. The following gives a user state of how to consume a web services from the mobile user through the middleware.

We are open-handed some user scenario, such as User1 is a mash up service developer. He wants to know all the forthcoming events in his city using his mobile phone. He knows that Yahoo Upcoming (Restful web service) offers such service and reads its online API document which describes how the service is used (e.g., providing co-ordinates as a parameters). Over the user interface on the mobile user, he then defines a mash up service (task) which contains a service action with all the required parameter and desired results. Finally, he executes the mash up service and gets the result displayed on the mobile user.

Another user scenario is, if User2 wants to know where he is going or at what location he wants to reach then he can able to view live map location which is based on Google API features [19] [20] and also add some mark symbol to remember that points or location with use of mash up application (task). At the result point-of-view user can execute app on their android based smartphone and gets the map after that user can add marking point with use of touch screen and the get suitable result for his discriminatory location with use of different Google map types such as Normal and Hybrid. For development of this mobile app, we are using android 2.2 platforms with eclipse tool using Helios service release 2 versions. We have also used Google map API to connect live map location with use of smartphone device.

To create this mobile mash up app we have follow below steps:

- Create My App project in to the eclipse IDE
- Download Google Play Service
- Import Google Play Service library
- Upgrade My App to show map
- Define <uses-permission> into My App activity manifest file
- Add UI and fragment controls in main.xml file
- Go to the Google API Console and grab the Google API key
- Configure Android Key for API Project
- Generate MD5 and SHA1 key for android app with use of “keytool” command from command prompt
- Take SHA1 key in Google API console and generate new android application key
- Now take new generated android application key and put into My App manifest file
- Create map fragment activity, Google Map object and also generate onMapClick event as well as setOnMapClickListener for getting latitude and longitude location for users
- Add interactive different touch feature like Add multiple marker, Zoom-In, Zoom-Out, Map Tagging (mash up the app)
- User can also able to view their location in normal Google map or in hybrid Google map

Figure 2 shows the process with a sequence of screenshots on the mobile client.

With use of Normal Google Map
With use of Hybrid Google Map

**Figure 2:** Live Map on Android based Smart-phone with use of Google Map API

**VII. CONCLUSIONS**

In this paper, we were doing analysis for mash up applications as services on web platform as well as on mobile/smart-phones platform; moreover we propose idea of service platform for mash up application for android mobile users, and we implement the service mash up app for smart-phone users on the android based mobile devices to show the feasibility of our proposed service mash up platform app.

**REFERENCES**