A MASHUP TECHNOLOGY WITH GOOGLEMAPS AND .NET FRAMEWORK

Terumalasetti Sailaja, Jalgama Ramesh, Rajya Lakhsmi

1Dept. of CSE, UCE-JNTUK Vizianagaram, 2HPS-ACS, Honeywell Technology Solutions
Email : sailaja.tirumalasetti@gmail.com, ramesh.jalgama@honeywell.com, rdavuluri@yahoo.com

Abstract—Mashup is a fusion of combining data from one or more resources into a single integrated technology. The paper discusses about the innovative approach of embedding Google Maps in to the website using the ASP.NET technology. The paper gives the details of the mashup architecture which integrates the heterogeneous sources of data. These mashups combine data from one or more sources into a single tool that are used by wide range of people. The Google Maps API is very flexible for developing the mapping capabilities to integrate into the websites or applications. The paper demonstrates an example by displaying the places information on the map based on the selection of the location.

Index Terms—Mashups, Google Maps API, ASP.NET, Microsoft SQL SERVER

I. INTRODUCTION

Recent years have seen an enormous increase in the number of web based applications that use the technique of visualization. The location information that displayed on the map is easy for the user to identify and the interaction of the user is more as it is user friendly. The usage of the Internet is wide and for any information to be known everyone is residing on the Internet. The websites are ready to display the location information on maps for easy understanding for the users as the usage of the maps are very huge.

In the development of web sites, mashup are termed as an application that integrates and uses data, presentation or functionality from two or more sources to create a new application. The term implies easy, fast integration, frequently using Application Programming Interface (API) and data sources to produce enriched results. The main characteristics of a mashup are combination, visualization and aggregation. The usage of the existing data more meaningful way for the professional or personal purpose. For a better human computer interaction WEB 2.0 introduced new possibilities through the Rich Internet Applications such as mashups that provide a user-driven micro integration of web accessible data. Rich Internet applications are web based application that function traditional desktop application. Web browsers are required for the access of the web page and the depending on the application specific requirements based on the technologies need to be installed on the client machine. The basic principle involved in designing mashups is the creation of new connotation by reusing and combining existing connotation from heterogeneous sources.

II. HISTORY

Google Maps is a web mapping service application and technology provided by Google, that powers many map-based services[1]. Google Corporation endorses many free tools over the Internet. One of the many is the Google Maps, which is a free interactive mapping and visualization service providing direct access to zoom-able maps or satellite views of the entire planet. Google Maps and Google Earth have a vivid range of application potential. The availability of these resources provide the developers to use the global maps to develop an application system integrated with the data that are build using the heterogeneous sources. As the demand for the map application increases over the time period more map service providers are available and APIs are also releasing for free to survive as the best map providers in the market. The map service providers are Yahoo! Maps and MSN Windows Live Local.

Google Maps is probably the best-known Ajax application. Google Maps provides some great advances over conventional map websites. The pan and zoom options, the change of location options without being page reloaded attracts the users. The release of API is makes possible for the GIS integration with the services or application developed by the developers for the access of the sources data.

III. MASHUP ARCHITECTURE

The paper presents architecture of mashups based on the data processing pipeline for the development and the execution of the integration logic of mashups.
A. Microsoft SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. Microsoft SQL is a database, the major task is to store and retrieve data as represented by other software applications, for those on the same computer or those on another computer across the network. Microsoft SQL Server supports all types of applications those ranging from small applications like store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet. The primary query languages are T-SQL and ANSI SQL. Microsoft SQL Server is designed to run on all platforms ranging from laptops to large multiprocessor servers.

Microsoft SQL Server is commonly used as the backend system for websites and supports thousands of concurrent users. SQL Server is more robust and scalable than a desktop database management system such as Microsoft Access. Microsoft Access is not the software of choice for a huge application that involves large database. MS Access is not efficient to handle large amounts of data, in the real time business environment.

SQL Server Database Engine is the core service for starting, processing and securing data. The Database Engine supports the controlled access and rapid transaction processing or online analytical processing data. This includes creating tables for storing data, and database objects such as indexes, views and stored procedures for viewing, managing and securing data. Although SQL Server can also be run as a desktop database system, it is most commonly used as a Server Database System.

Server based database systems are designed to run on a central server, so that multiple users can access the data simultaneously. The access of the database is normally through the application.

B. .NET Framework

.NET Framework is a cluster of several technologies. The technologies that included are .NET languages, the Common Language Runtime (CLR), the .NET Framework class library, ASP.NET, and Visual Studio. The .NET languages that include are Visual Basic, C#, J#, and C++. The selected language to develop the application is C#.

ASP.NET is the engine that hosts the web applications that are created with .NET and supports any features from the .NET class library. ASP.NET includes a set of web-specific services like secure authentication and data storage. An ASP.NET web page is a framework that is used to create dynamic web pages. A simple HTML web page is static, its content is determined by the fixed HTML markups which are in the page. Dynamic web pages are easily created using the ASP.NET. ASP.NET is a unified development model that incorporates services necessary to build enterprise-class web applications. ASP.NET web pages are known as web forms that are main building block for application development. The web pages are associated with the markup languages. The JavaScript is the scripting language that is used to build the application discussed in the paper.

C. Web Browser

A Web Browser commonly referred to as a browser is a web application that is used to retrieve, presents and traverse information resources on the World Wide Web. An information resource is identifies by a Uniform Resource Identifier (URI) and it consists of a web page, image, video or any other piece of content. Hyperlinks presents in resources enable users to navigate through the browsers to related resources. A web browser is defined as an application software or program designed to allow users to access, retrieve and view documents and other resources on the Internet. The major intention of the browsers is to use the World Wide Web, but they can also be used to access information provided by web servers in private networks or files in file systems. The major web browsers are Google Chrome, Firefox, Internet Explorer, Opera and Safari.

D. User

The users are the end users who access the website for the information. The users who have the intention of viewing the information on map. In this paper the users can able to access the places information based on the location that is selected.
IV. GOOGLE MAPS API

Google Maps is a web-based application that is designed to view the maps on the web pages. The features of the Google Maps include the search by location option, options for the direction in a multimodal transportation system and the addition data also is added to display on the map. Google maps is a web mapping service, free for commercial use that contains detailed global base maps such as streets, satellite imaginary and terrain. API is an interface to a software program that allows users to interact and customize the maps. The Mercator projection is the projection that is used by the Google Maps, the displaying the locations around the poles is very easy by using this projection.

Google Maps API (Application Programming Interface) allows wrapping up the Google Maps in to web pages and customizes the Google maps [5]. The developers can easily customize the Google Maps using the Google Maps API. The API provides the developer with rich set of applications. The JavaScript is used to embed the maps in to the applications. The current version of the Google Maps API is the Google Maps JavaScript API v3. The version 2 is deprecated. The version 2 provides all the features of version 3 but the version 2 does not support to design the applications for the smart phones. The version 3 is introduced that supports for the smart phones. The version 3 is lighter and faster and is specially oriented toward the mobile devices through it performs well and works fine on the personal computers. The maps API web services provides the many APIs like the Direction API, Distance Matrix API, Geocoding API, Elevation API, Time Zone API. The API is rich the libraries such as AdSense library, drawing library, Geometry library, Panoramio library, Places library, Weather library.

V. EMBEDDING OF GOOGLE MAPS USING THE MASHUP TECHNOLOGY

The application discussing in this paper contains a SQL Server database. The SQL server database contains two tables. The first table gives the details of the locations for an instance consider the states of India and another column contains the location Id. The second table gives the details of the location information, in this example it provides the important places in particular location that is the places in the state. The selection of the state is based on the user choice from the list available in the drop down list on the first page. The second table contains the columns such as Id, Places, location Id, Coordinates, Information. The mapping on the Google Maps is based on the coordinate’s information, useful when GPS devices are used to locate the location information.

The design of the application includes the two web pages. The first page gives the details of the locations that are available. The user can select the location from the available location on the drop down list.
Figure 2 give the list of the states in India. User can select any of the state. The list is available in the drop down list for the easy access of the user. The lists of states are retrieved from the database.

The second page contains the location information displayed on the maps based on the selection of the location by the user. As discussed earlier Google Maps API is used to embed in the website. The initial criteria for the Google Maps API are the API key provided by the Google. The Google Key is obtained by accessing the Google account. The client side scripting of ASP.NET JavaScript code must be injected to asynchronously load the Maps. A web service request of the form “http://maps.googleapis.com/maps/api/js?v=3.exp.&sens or=true” The sensor options give the details of the usage of the GPS devices. If the sensor is true then the GPS device is activated if it is false then the GPS is not activated. The GPS option is mostly useful for the mobile device users where the GPS is the functionality in the mobile devices.

The conversion of human readable address to geographic coordinates is known as Geocoding. The Geocoding process is used to place the markers on the map. The Geocoding API also is useful for the conversion of the address. Reverse Geocoding is the process of converting geo-coordinates to the human readable format. The Geocoding API provides the services directly just by accessing HTTP request within in the source page. The required parameters for the Geocoding are the address or latlng and the sensor. The optional parameters are the bounds, language, region and components. The address parameter is the addresses that need to be geocode. The latlng parameter describes the textual latitude/longitude value that wish to obtain the closest human readable address. The components parameter is an optional parameter that is used to filter for which need to geocode. The components filters, separated by a pipe (|). Each component filter consists of a component value pairs and will fully restrict the results from the geocoder. The sensor indicates whether or not the geocoding request comes from a device with a location sensor. The value of the sensor may be either true or false. The bounds parameter gives the description of the bounding box of the view port within which to bias geocode results more prominently. This parameter just influence but not fully restrict, results from the geocoder. The parameter language gives the details into which to return the results. If the language is not supplied the geocoder attempts to use the native language of the domain from which the request is sent whenever possible. The region parameter specifies a character value. The geocoding responses are returned in the format indicates by the output flag within the URLs request’s path. The JSON and XML are the two formats that are used for the geocoding responses. The google.maps.Geocoder is a class to define the geocoding options.

Overlays are the objects on the map that are tied to the latitude or longitude coordinates. The overlays move on when drag or zoom the map. Overlays reflect the objects that added to the map to designate points, lines, areas or collection of objects. The maps API have several types of overlays such as markers, polylines, and areas of arbitrary shapes, map layers, and infowindow. The overlays also can be customized. Overlays are often added to the map upon their construction. The setMap () is the option for directly using the overlays. To remove an overlay from the map, the setMap () method is called by passing null.

Locations on the map are displayed using the markers. By default a standard icon is used for the marker, the marker can be customized using the custom icon within the marker constructor or by calling the setIcon () on the marker. The google.maps.Marker is a class to define the marker options. Marker options object specifies the initial properties of the marker. The fields that are important and commonly set for the construction of the marker are position, map, and icon if the marker is customized. The position specifies the latlng by identifying the initial location. The map is an optional parameter that specifies the Map object on which to place the marker. Markers are designed to be interactive. By default, they receive ‘click’ events, for an instance, and are often used within the event listeners to bring up info windows on clicking the marker. The markers property draggable can set to true to make the marker user editable on the map. The markers can also be animated for the exhibition of the dynamic movement in a variety of different circumstances. Marker animation property is used to specify the animation of the marker. The defining of icon involves setting a number of properties that define the visual behavior of the marker. Icon can simply indicate an image to use instead of the default Google Maps pushpin icon by setting the markers icon property to the URL of an image. The Google Maps API will size the icon automatically or the MarkerImage object is used to define the image such as the size of the icon, the origin of the icon and the anchor where the icon’s hotspot must be located.

Lines on the markers are displayed using the polylines. The Polyline class defines a linear overlay of connected line segments on the map. A polyline object consists of an array of LatLng locations and creates a series of line segments that connects those locations in an ordered sequence. The polyline constructor takes a set of polyline options specifying the LatLng coordinates of the line and set of styles to adjust the polylines visual behavior. Polylines are drawn as sequence of straight segments on the map. The custom colors, weights and opacities can be specified for the stroke of the line within a polyline options object that used when constructing the line or change those properties after construction. The polyline supports the stroke styles such as strokeColor, strokeOpacity and strokeWeight. The strokeColor specifies a hexadecimal HTML color of the format “#FFFFFF”. The polyline class
does not support named colors. The strokeOpacity specifies a numerical fractional value between 0.0 and 1.0 (default) of the line’s color. The strokeWeight specifies the weight of the line’s stroke in pixels. The polylines editable option defines whether the shape is editable by users on the map. A polyline specifies a series of coordinates as an array of LatLng objects. To retrieve the coordinates by calling the getPath() that return the array of the type MVC Array. The arrays can be manipulated and inspected using the options as getAT(), insertAT(), removeAT().

JavaScript responds to the interaction by generating events and expects a program to listen for the interesting events. The event model for the Google Maps API version 3 is similar to that used in the version 2 of the API. There are two types of events the user events and the MVC state changes. The user events are propagates from the DOM to the Google Maps API. These events are separate and distinct from standard DOM events. MVC state changes notification reflect changes in maps API objects and are named using property changed convention. Each maps API object exports a number of named events. Programs interested in certain events will JavaScript event listeners for those events and execute code when those events are received by registering addListener() event handlers on the google.maps.event namespace.

The infowindow display the content in floating window above the map. The infowindow has a content area and a tapered stem, where the tip of the stem is at the specified location on the map. The infowindows can be displayed by clicking the markers on the maps. The InfoWindow is a constructor to specify the options of the marker. The InfoWindow options are the content, pixeloffset, maxwidth, and position. The InfoWindow.open() is used for the displaying of the InfoWindow. The InfoWindow.close() is used for the closing the infowindow.

The mapOptions are used to customize the map. The options include the zoom, center, mapTypeId.

---

Figure 3: Displaying the places information on the map, the satellite view.
Figure 3 and Figure 4 gives the details of the places on the maps based on the location selected by the user. The figure 3 displays map using the satellite view and the map view. The markers are customized. The places information is retrieved from the database and based on the selected location the places are retrieved and displayed on the map. The polylines are used just for connecting the different places. The infowindow gives the information that is retrieved from the database. On clicking the marker the infowindow will be displayed.

VI. CONCLUSION

The paper discussed the mashup architecture for the heterogeneous sources. The paper apostle a mashup approach for the embedding the Google maps in the application. The architecture describes the integration of the different applications, the wrapping up of the .net framework, the database and the browser for the interaction to the user. The main contribution is the compendium of the emerging technologies in the form of mashups to meet the needs of the user. The paper presents an example of the embedding the Google Maps.

VII. ACKNOWLEDGMENT

The authors would like to thank B.Vijayalakshmi, P.Manonmani, and Matthew Donald as their papers helped us to contribute and present the paper.

REFERENCES


