

■ KENNETH E. KENDALL, Feature Editor, School of Business-Camden, Rutgers University

Mashups provide a way of creating a new application by combining two other applications from the Web. They can therefore be called a Web application hybrid. Last month's article served as an introduction to mashups. This month, we'll explain how you can create your own mashups with a minimal amount of code. [Kenneth E. Kendall]

## Mashups: Making Web Application Hybrids from Existing Information on the Web

by Allen Schmidt, Madison Area Technical College; and Kenneth E. Kendall, Rutgers University



**Allen Schmidt**

*teaches JavaScript and Systems Analysis and Design in the Information Technology Department at Madison Area Technical College (MATC) in Madison, Wisconsin. He has written four books on CASE tools and has*

*co-authored an interactive, Web-based simulation that teaches systems analysis and design called HyperCase®.*

**schmidt@chorus.net**



**Kenneth E. Kendall**

*is president-elect and a Fellow of DSI. He is a professor of e-commerce and information technology at Rutgers University, School of Business-Camden. He is also a co-author of HyperCase® which can be*

*found at <http://www.thekendalls.org/hypercas.html> and co-author of Systems Analysis and Design, 6th edition.*

**ken@thekendalls.org**

Mashups are created by taking two of more application programming interfaces (or APIs) and combining them to make one Web site that has an entirely new purpose. Some companies like PayPal will charge a vendor a monthly fee to use their API; other organizations consider their APIs to be open source so that you may freely use their API to create something new. In fact, legal issues are still being debated (Gerber, 2006). We thought you might like to see how a mashup can easily be created with only a few lines of code. Make sure you can use the information and then proceed to write your own mashups.

Google Maps (2007) is a great place to start. A handy reference book on creating new applications with Google Maps was written by Gibson and Earle (2006). A more general tutorial can be found at IBM (2206).

**Creating Mashups**

Mashups may be created in a number of different ways, depending on the provider of the Web service. The developer usually has some options, but since the Web page using the mashup is sending a message to the mashup provider, the format of the message and the sequence of the activities are determined by the provider.

Mashups may be as simple as providing a Google search of your Web site. The code for a Google search mashup would include a Web form to control the search. There are several hidden fields used by Google. The form illustrated below uses a table to control the layout. An image is obtained from the Google site. The action attribute of the form tells the form to use [www.google.com/search](http://www.google.com/search).

```
<form method=get action="http://www.google.com/search">
<input type=hidden name=ie value=UTF-8>
<input type=hidden name=oe value=UTF-8>
<table border="1" cellpadding="4" cellspacing="0" bordercolor="#000066">
<tr>
<td align="center">
<a href="http://www.google.com/">
</a><br>
<font size="-1" face="Verdana, Arial, Helvetica, sans-serif">
```

Search this site using Google</font>

<p>

```
<input type="text" name="q"
size="31" maxlength="255"
value="">
```

<font size=-1>

```
<input type="hidden" name="domains"
value="JavaScript Pages">
```

<br>

```
<input type="radio"
name="sitesearch" value="http://
yourWebSiteAddressHere" checked>
```

<font face="Verdana, Arial, Helvetica, sans-serif"> This Site

```
<input type="radio"
name="sitesearch" value="">
```

Entire Web</font><br>

</font></p>

<font size=-1>

```
<input type="submit"
name="btnG" value="Google
Search">
```

</font> </td></tr>

</table>

</form>

Another simple mashup is including directions on a Web page. This example also uses Google by sending the location to Google Maps. The code is illustrated below. The line

```
http://maps.google.com/
maps?f=q&hl=en&q=3550+
Anderson+St,+Madison,+WI+53704
```

sends the desired address to Google Maps, which opens the map in a new Window. Since a Web address cannot contain any spaces, the plus sign is used as a substitute.

```
<a href="http://
maps.google.com/
maps?f=q&hl=en&q=3550+
Anderson+St,+Madison,+WI+53704"
target="popup"
onclick="new_window()">
```

```
</a></p>
```

```
<p><font color="#000099" size="-1" face="Verdana, Arial, Helvetica, sans-serif">Map and Driving Directions</font>
```

The first two examples use Google's Web services, and are fairly easy to code. A last example of a more

formal mashup uses Google Maps to put points on the map. The code is fairly lengthy, and rather than include it, mention will be made of some of the key points. The code shows the DSI annual meeting in Phoenix, Arizona as well as the Sky Harbor International Airport using Google maps and is illustrated in Figure 1.

### Using Google Maps to Create Mashups

The first step in using Google Maps is to register with Google and obtain a key for the map. The key is unique to a particular Web domain and cannot be used on any other domain.

The next step is to create a map object in the body of the Web page with the desired height and width. This is followed by adding some standard controls to the map, such as a zoom and pan control and a view to control whether a map, satellite or both should display. This is rather easy and the developer has some control over which controls to use. The following code adds these controls using JavaScript:

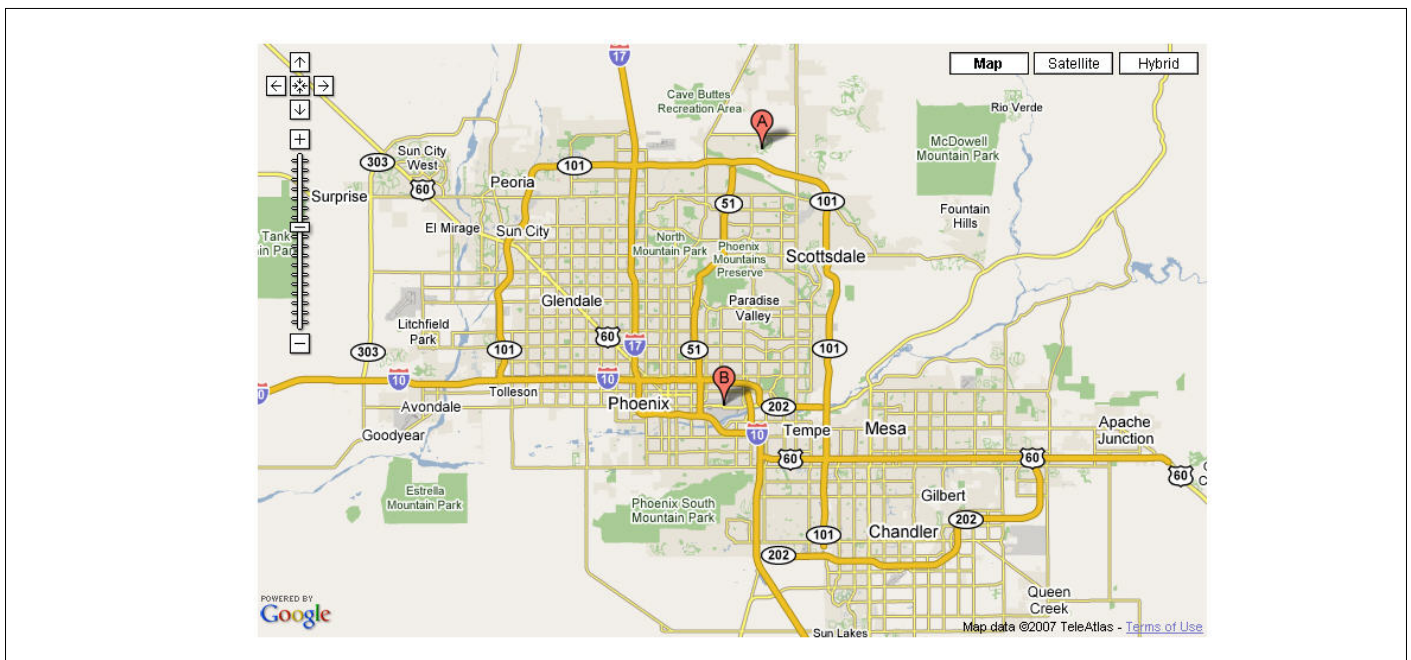


Figure 1: This map shows the site of the 38<sup>th</sup> DSI Annual Meeting as well as the location of the airport. (Source: Google Maps)

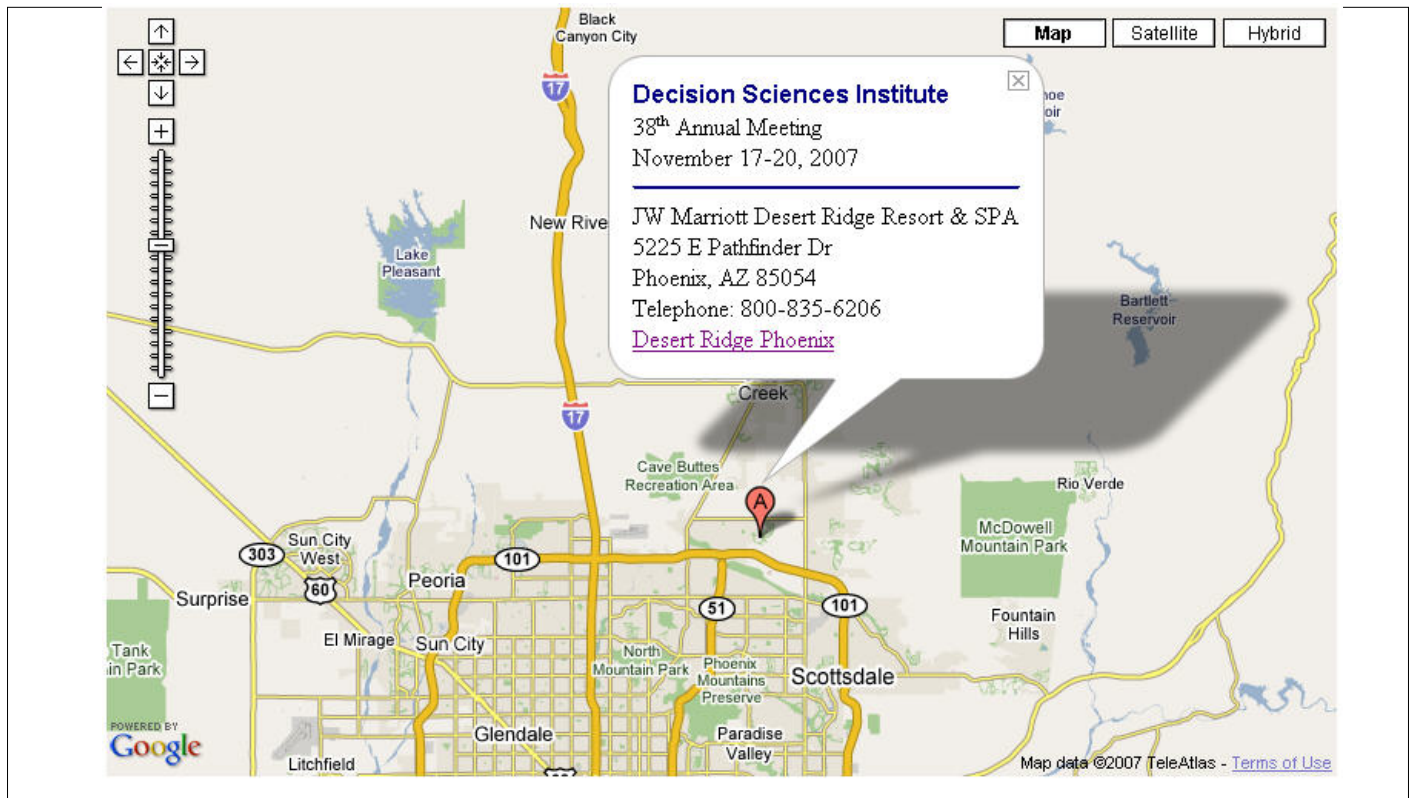


Figure 2: Typically a balloon can show details about a point on a map. (Source: Google Maps)

```
map.addControl(new
GLargeMapControl()); //add a
large control for pan/zoom
```

```
map.addControl(new
GMapTypeControl()); //add a
control for map or satellite
```

The next step is to center the map, in this case over New Jersey, and select the default zoom level. Zoom values range from 0 (showing the whole earth) to 17 which is very detailed. In this example the zoom is set to 10, which shows the Phoenix city:

```
map.setCenter(new GLatLng(33.5, -
112), 10);
```

The next steps are to set the map pushpin icon image, shadow, size and so on. This is done by using Google methods, such as:

```
baseIcon.iconSize = new GSize(20,
34);
```

and

```
baseIcon.shadow = "http://
www.google.com/mapfiles/
shadow50.png";
```

The final steps include defining the markers that appear on the map. This is a bit complicated and consists of specifying the latitude and longitude of each marker, as well as the text that should display (in HTML format) when the marker is clicked. Since the latitude and longitude is not often known, Google provides a geocoder function. The street, city, and state are sent to the geocoder, which returns the latitude and longitude, if known. This process is repeated for each point. One disadvantage of using the geocoder function is that it will not work for addresses that it cannot resolve. The developer may have to provide an alternative way to look up latitude and longitude and write code for these exceptions.

Once the map displays, each marker may be clicked to provide information about the location. A balloon appears that can contain any information that the developer wishes to include. The name, address, telephone number and a link to a related Web site

are usually included, illustrated in Figure 2. Since the information shown in the balloon is coded using HTML, it may also include a photo or a video clip.

Google Maps allows a sidebar containing information in text format. Clicking one of the sidebar links causes the map to scroll to the selected point and display the balloon with marker information.

### Mashups and Ajax

One of the problems that occurs when working with a mashup such as Google Maps is that there may be too many markers on the map or that the markers may be spread over a large geographical area. The solution is to store the marker information in an XML document, or in a database with server side programming to create an XML document. The XML document should contain the marker information, such as the information needed for the geocoder function to work, or the latitude and

```

<?xml version="1.0" ?>
- <mapObjects>
- <mapObject>
  <name>JW Marriott Desert Ridge Resort & SPA</name>
  <street>5225 E Pathfinder Dr</street>
  <city>Phoenix</city>
  <state>AZ</state>
  <zip>85054</zip>
  <url>http://jwdesertridgeresort.com///</url>
  <linkName>Desert Ridge Phoenix</linkName>
</mapObject>
- <mapObject>
  <name>Sky Harbor International Airport</name>
  <street>3400 E Sky Harbor Blvd</street>
  <city>Phoenix</city>
  <state>AZ</state>
  <zip>85034</zip>
  <url>http://www.skyharborairport.com</url>
  <linkName>Sky Harbor International Airport</linkName>
</mapObject>
</mapObjects>

```

Figure 3: This XML document shows information needed for the geocoder function to work.

longitude, as well as the information that should display in the marker balloon. An example is shown in Figure 3. The XML tag names are not fixed, and may be created or changed by the developer. For example, the telephone number or an image file name may be included as well.

Since the Web page must use a later browser and have JavaScript enabled to use Google Maps, it will also have Ajax capabilities. Ajax uses JavaScript code to obtain the XML document, which is stored as an array of elements for each marker (called mapObject in Figure 3). The JavaScript code loops through each mapObject creating the marker point and the corresponding balloon information, as well as any sidebar information.

The advantage of using an XML document is that it may be used to select the locations that match some query, such as a country, state, province, or city location. This may be done on the server, which would create the XML document, or it may be done in the browser using JavaScript code. The user

interface may have an entry text box, a drop-down list, or radio buttons to enter the conditions used to select the desired map markers. An example of this is the Bikram Yoga Finder at <http://www.bikramfinder.com/>. Enter a ZIP code or select a city to display a map for that particular location. Notice the use of a convenient sidebar to quickly locate markers. The Google Map scrolls to the sidebar location. Another advantage of using XML is that it is easy to update the XML document and the next time the map is used, it will contain the additional or removed points, as well as any change in the balloon text.

A final advantage overcomes the limitation that each domain may have only one Google Map. Clicking a radio button could use Ajax to load an entirely different XML document, including the point used to center the map and the zoom level. The developer could use radio buttons if there were a few different maps to display, or a drop-down list. For example, when teaching a history course, different points could display for different time periods.

There are many Web resources for using Google maps and other mashups. Many of them provide examples and detailed explanations of the code. One of the best sites is <http://www.programmableweb.com/>, which has a comprehensive list of mashups along with links and details for each mashup and a dashboard showing some statistics about the growth of mashups.

The use of mashups will only increase over time, making it easier to create Web sites that have increased functionality and do many things that would be time consuming, expensive, or in the case of mapping, almost impossible for the developer to accomplish.

## References

- Bikram Yaga Finder. (2007). <http://www.bikramfinder.com/>
- Gerber, R. S. (2006). Mixing it up on the Web: Legal issues arising from Internet "mashups." *Intellectual Property & Technology Law Journal*, 18(8), August, 11-14.
- Gibson, R., & Erle, S. (2006). *Google maps hacks*. Sebastopol, CA: O'Reilly.
- Google. (2007). <http://desktop.google.com/plugins/>
- Google Maps. (2007). <http://maps.google.com/>
- IBM. (2006). *The ultimate mashup — Web services and the semantic Web*, <http://www-128.ibm.com/developerworks/edu/x-dw-x-ultimashup1.html>
- Programmableweb. (2007). <http://www.programmableweb.com/> ■

**Kenneth E. Kendall**  
 School of Business-Camden  
 Rutgers University  
 Camden, NJ 08102  
 (856) 225-6586  
 fax: (856) 424-6157  
 ken@thekendalls.org  
<http://www.thekendalls.org>